Das FreeBSD-Handbuch

The FreeBSD German Documentation Project

Das FreeBSD-Handbuch
von The FreeBSD German Documentation Project

Veröffentlicht Februar 1999

Willkommen bei FreeBSD! Dieses Handbuch beschreibt die Installation und den täglichen Umgang mit *FreeBSD 8.4-RELEASE* und *FreeBSD 9.1-RELEASE*. Das Handbuch ist *jederzeit unter Bearbeitung* und das Ergebnis der Arbeit vieler Einzelpersonen. Dies kann dazu führen, dass bestimmte Bereiche nicht mehr aktuell sind und auf den neuesten Stand gebracht werden müssen. Bei Unklarheiten empfiehlt es sich daher stets, die englische Originalversion (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/index.html) des Handbuchs zu lesen.

Wenn Sie bei der Übersetzung des Handbuchs mithelfen möchten, senden Sie bitte eine E-Mail an die Mailingliste 'FreeBSD German Documentation Project' <de-bsd-translators@de.FreeBSD.org>.

Die aktuelle Version des Handbuchs ist immer auf dem FreeBSD-Webserver (http://www.FreeBSD.org/) verfügbar und kann in verschiedenen Formaten und in komprimierter Form vom FreeBSD-FTP-Server (ftp://ftp.FreeBSD.org/pub/FreeBSD/doc) oder einem der vielen Spiegel herunter geladen werden (ältere Versionen finden Sie hingegen unter http://docs.FreeBSD.org/doc/). Vielleicht möchten Sie das Handbuch aber auch durchsuchen (http://www.FreeBSD.org/search/index.html).

FreeBSD ist ein eingetragenes Warenzeichen der FreeBSD Foundation.
3Com und HomeConnect sind eingetragene Warenzeichen der 3Com Corporation.
3ware und Escalade sind eingetragene Warenzeichen von 3ware Inc.
ARM ist ein eingetragenes Warenzeichen von ARM Limited.
Adaptec ist ein eingetragenes Warenzeichen von Adaptec, Inc.
Adobe, Acrobat, Acrobat Reader und PostScript sind entweder eingetragene Warenzeichen oder Warenzeichen von Adobe Systems Incorporated in den Vereinigten Staaten und/oder in anderen Ländern.
Apple, FireWire, Mac, Macintosh, Mac OS, Quicktime und TrueType sind eingetragene Warenzeichen von Apple Computer, Inc., in den Vereinigten Staaten und anderen Ländern.
Corel und WordPerfect sind Warenzeichen oder eingetragene Warenzeichen der Corel Corporation und/oder ihren Gesellschaften in den Vereinigten Staaten und/oder anderen Ländern.
Sound Blaster ist ein Warenzeichen von Creative Technology Ltd. in den Vereinigten Staaten und/oder in anderen Ländern.
CVSup ist ein eingetragenes Warenzeichen von John D. Polstra.
Heidelberg, Helvetica, Palatino und Times Roman sind Marken der Heidelberger Druckmaschinen AG in Deutschland und anderen Ländern.
IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390 und ThinkPad sind Warenzeichen der International Business Machines Corporation in den Vereinigten Staaten, anderen Ländern oder beiden.
IEEE, POSIX und 802 sind eingetragene Warenzeichen vom Institute of Electrical and Electronics Engineers, Inc. in den Vereinigten Staaten.
Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium und Xeon sind Warenzeichen oder eingetragene Warenzeichen der Intel Corporation oder ihrer Gesellschaften in den Vereinigten Staaten und in anderen Ländern.
Intuit und Quicken sind eingetragene Warenzeichen und/oder Dienstleistungsmarken von Intuit Inc. oder einer ihrer Geselllschaften in den Vereinigten Staaten und in anderen Ländern.
Linux ist ein eingetragenes Warenzeichen von Linus Torvalds.
LSI Logic, AcceleRAID, eXtremeRAID, MegaRAID und Mylex sind Warenzeichen oder eingetragene Warenzeichen der LSI Logic Corp.
M-Systems und DiskOnChip sind Warenzeichen oder eingetragene Warenzeichen von M-Systems Flash Disk Pioneers, Ltd.
Macromedia, Flash und Shockwave sind Warenzeichen oder eingetragene Warenzeichen von Macromedia, Inc. in den Vereinigten Staaten und/oder in anderen Ländern.
Microsoft, MS-DOS, Outlook, Windows, Windows Media und Windows NT sind entweder eingetragene Warenzeichen oder Warenzeichen der Microsoft Corporation in den Vereinigten Staaten und/oder in anderen Ländern.
Netscape und Netscape Navigator sind eingetragene Warenzeichen der Netscape Communications Corporation in den Vereinigten Staaten und in anderen Ländern.
GateD und NextHop sind eingetragene Warenzeichen und Warenzeichen von NextHop in den Vereinigten Staaten und in anderen Ländern.
Motif, OSF/1 und UNIX sind eingetragene Warenzeichen und IT DialTone und The Open Group sind Warenzeichen der The Open Group in den Vereinigten Staaten und in anderen Ländern.
Oracle ist ein eingetragenes Warenzeichen der Oracle Corporation.
PowerQuest und PartitionMagic sind eingetragene Warenzeichender PowerQuest Corporation in den Vereinigten Staaten und/oder anderen Ländern.
RealNetworks, RealPlayer und RealAudio sind eingetragene Warenzeichen von RealNetworks, Inc.
Red Hat, RPM, sind Warenzeichen oder eingetragene Warenzeichen von Red Hat, Inc. in den Vereinigten Staaten und in anderen Ländern.
SAP, R/3 und mySAP sind Warenzeichen oder eingetragene Warenzeichen der SAP AG in Deutschland und in anderen Ländern der Welt.
Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JSP, JVM, Netra, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS und Ultra sind Warenzeichen oder eingetragene Warenzeichen von Sun Microsystems, Inc. in den Vereinigten Staaten und in anderen Ländern.
Symantec und Ghost sind eingetragene Warenzeichen der Symantec Corporation in den Vereinigten Staaten und in anderen Ländern.
MATLAB ist ein eingetragenes Warenzeichen von The MathWorks, Inc.
SpeedTouch ist ein Warenzeichen von Thomson
U.S. Robotics und Sportster sind eingetragene Warenzeichen der U.S. Robotics Corporation.
VMware ist ein Warenzeichen von VMware, Inc
Waterloo Maple und Maple sind Warenzeichen oder eingetragene Warenzeichen von Waterloo Maple Inc.

Inhaltsverzeichnis

Tabellenverzeichnis

Vorwort

Über dieses Buch

Der erste Teil dieses Buchs führt FreeBSD-Einsteiger durch den Installationsprozess und stellt leicht verständlich Konzepte und Konventionen vor, die UNIX® zu Grunde liegen. Sie müssen nur neugierig sein und bereitwillig neue Konzepte aufnehmen, wenn diese vorgestellt werden, um diesen Teil durchzuarbeiten.

Wenn Sie den ersten Teil bewältigt haben, bietet der umfangreichere zweite Teil eine verständliche Darstellung vieler Themen, die für FreeBSD-Administratoren relevant sind. Wenn Kapitel auf anderen Kapiteln aufbauen, wird das in der Übersicht am Anfang eines Kapitels erläutert.

Weitere Informationsquellen entnehmen Sie bitte Anhang B.

Änderungen gegenüber der dritten Auflage

Die aktuelle Auflage des Handbuchs ist das Ergebnis der engagierten Arbeit Hunderter Mitarbeiter des FreeBSD Documentation Projects in den vergangenen 10 Jahren. Die wichtigsten Änderungen dieser Auflage gegenüber der dritten Auflage von 2004 sind:

- Kapitel 26, DTrace, ein neues Kapitel, informiert Sie über die mächtigen Funktionen zur Leistungsmessung, die dieses Werkzeug bietet.
- Kapitel 21, File Systems Support, ebenfalls ein neues Kapitel, enthält Informationen über die Unterstützung nicht-nativer Dateisysteme (beispielsweise ZFS von Sun™) durch FreeBSD.
- Kapitel 18, Security Event Auditing, wurde neu angelegt, um über die neuen Auditing-Fähigkeiten von FreeBSD zu informieren.
- Kapitel 23, Virtualisierung, wurde hinzugefügt und enthält Informationen zur Installation von FreeBSD in verschiedenen Virtualisierungs-Programmen.
- Kapitel 3, FreeBSD 9.*x* (und neuer) installieren, wurde hinzugefügt, um die Installation von FreeBSD mit dem neuen Installationswerkzeug, **bsdinstall**, zu dokumentieren.

Änderungen gegenüber der zweiten Auflage (2004)

Die dritte Auflage des Handbuchs war das Ergebnis der über zwei Jahre dauernden engagierten Arbeit des FreeBSD Documentation Projects. Die gedruckte Ausgabe war derart umfangreich, dass es notwendig wurde, sie in zwei Bände aufzuteilen. Die wichtigsten Änderungen dieser Auflage waren:

- Kapitel 12, Konfiguration und Tuning, enthält neue Abschnitte über ACPI, Energie- und Ressourcenverwaltung und das Werkzeug `cron`.
- Kapitel 15, Sicherheit, erläutert nun Virtual Private Networks (VPNs), Zugriffskontrolllisten (ACLs) und Sicherheitshinweise.
- Kapitel 17, Mandatory Access Control (MAC), ist ein neues Kapitel, das vorgeschriebene Zugriffskontrollen vorstellt und erklärt, wie FreeBSD-Systeme mit MACs abgesichert werden können.

- Kapitel 22, Vinum, ist ebenfalls ein neues Kapitel in dieser Auflage. Dieses Kapitel beschreibt den Logical-Volume-Manager Vinum, der geräteunabhängige logische Platten und RAID-0, RAID-1 sowie RAID-5 auf Software-Ebene bereitstellt.
- Zum Kapitel Kapitel 28, PPP und SLIP, wurde ein Abschnitt über Fehlersuche hinzugefügt.
- Kapitel 29, Elektronische Post (E-Mail), wurde um Abschnitte über andere Transport-Agenten (MTAs), SMTP-Authentifizierung, UUCP, **fetchmail**, **procmail** und weitere Themen erweitert.
- Kapitel 30, Netzwerkserver, ist ein weiteres neues Kapitel dieser Auflage. Das Kapitel beschreibt, wie der **Apache** HTTP-Server, **ftpd** und ein **Samba**-Server für Microsoft® Windows®-Clients eingerichtet werden. Einige Abschnitte aus dem Kapitel 32, Weiterführende Netzwerkthemen, befinden sich nun, wegen des thematischen Zusammenhangs, in diesem Kapitel.
- Das Kapitel 32, Weiterführende Netzwerkthemen, beschreibt nun den Einsatz von Bluetooth®-Geräten unter FreeBSD und das Einrichten von drahtlosen Netzwerken sowie ATM-Netzwerken.
- Neu hinzugefügt wurde ein Glossar, das die im Buch verwendeten technischen Ausdrücke definiert.
- Das Erscheinungsbild der Tabellen und Abbildungen im Buch wurde verbessert.

Änderungen gegenüber der ersten Auflage (2001)

Die zweite Auflage ist das Ergebnis der engagierten Arbeit der Mitglieder des FreeBSD Documentation Projects über zwei Jahre. Die wichtigsten Änderungen gegenüber der ersten Auflage sind:

- Ein Index wurde erstellt.
- Alle ASCII-Darstellungen wurden durch Grafiken ersetzt.
- Jedes Kapitel wird durch eine Übersicht eingeleitet, die den Inhalt des Kapitels zusammenfasst und die Voraussetzungen für ein erfolgreiches Durcharbeiten des Kapitels darstellt.
- Der Inhalt wurde in die logischen Abschnitte “Erste Schritte”, “Systemadministration” und “Anhänge” unterteilt.
- Kapitel 2 (“FreeBSD installieren”) wurde komplett neu geschrieben und mit Abbildungen versehen, die Einsteigern das Verständnis des Texts erleichtern.
- Kapitel 4 (“Grundlagen des UNIX Betriebssystems”) wurde um den Abschnitt “Dämonen, Signale und Stoppen von Prozessen” erweitert.
- Das Kapitel 5 (“Installieren von Anwendungen”) behandelt nun auch Pakete.
- Kapitel 6 (“Das X Window System”) wurde neu geschrieben. Der Schwerpunkt liegt auf modernen Benutzeroberflächen unter XFree86™ 4.X wie **KDE** und **GNOME**.
- Das Kapitel 13 (“FreeBSDs Bootvorgang”) wurde erweitert.
- Kapitel 19 (“Speichermedien”) ist aus den beiden Kapiteln “Laufwerke” und “Sicherungen” entstanden. Die in den beiden Kapiteln diskutierten Themen sind so leichter zu verstehen. Hinzugekommen ist ein Abschnitt über Software- und Hardware-RAID.
- Das Kapitel 27 (“Serielle Datenübertragung”) wurde umorganisiert und auf FreeBSD 4.X/5.X angepasst.
- Das Kapitel 28 (“PPP und SLIP”) wurde aktualisiert.
- Kapitel 32 (“Weiterführende Netzwerkthemen”) wurde um viele neue Abschnitte erweitert.
- Kapitel 29 (“Electronic Mail”) wurde um einen Abschnitt über die Konfiguration von **sendmail** erweitert.

- Kapitel 11 (“Linux® Compatibility”) behandelt zusätzlich die Installation von **Oracle®** und **Mathematica®**.
- Neu hinzugekommen sind:
  - Konfiguration und Tuning (Kapitel 12) und
  - Multimedia (Kapitel 8).

Gliederung

Dieses Buch ist in fünf Abschnitte unterteilt. Der erste Abschnitt, *Erste Schritte*, behandelt die Installation und die Grundlagen von FreeBSD. Dieser Abschnitt sollte in der vorgegebenen Reihenfolge durchgearbeitet werden, schon Bekanntes darf aber übersprungen werden. Der zweite Abschnitt, *Oft benutzte Funktionen*, behandelt häufig benutzte Funktionen von FreeBSD. Dieser Abschnitt sowie alle nachfolgenden Abschnitte können in beliebiger Reihenfolge gelesen werden. Jeder Abschnitt beginnt mit einer kurzen Übersicht, die das Thema des Abschnitts und das nötige Vorwissen erläutert. Die Übersichten helfen dem Leser, interessante Kapitel zu finden und erleichtern das Stöbern im Handbuch. Der dritte Abschnitt, *Systemadministration*, behandelt die Administration eines FreeBSD-Systems. Der vierte Abschnitt, *Netzwerke*, bespricht Netzwerke und Netzwerkdienste. Der fünfte Abschnitt enthält Anhänge und Verweise auf weitere Informationen.

Kapitel 1, Einführung

Dieses Kapitel macht Einsteiger mit FreeBSD vertraut. Es behandelt die Geschichte, die Ziele und das Entwicklungsmodell des FreeBSD-Projekts.

Kapitel 2, FreeBSD 8.x (und älter) installieren

Beschreibt den Ablauf der Installation von FreeBSD 8.*x* und früher mittels **sysinstall**. Spezielle Installationsmethoden, wie die Installation mit einer seriellen Konsole, werden ebenfalls behandelt.

Kapitel 3, FreeBSD 9.x (und neuer) installieren

Beschreibt den Ablauf der Installation von FreeBSD 9.*x* und neuere mittels **bsdinstall**.

Kapitel 4, Grundlagen des UNIX Betriebssystems

Erläutert die elementaren Kommandos und Funktionen von FreeBSD. Wenn Sie schon mit Linux oder einem anderen UNIX System vertraut sind, können Sie dieses Kapitel überspringen.

Kapitel 5, Installieren von Anwendungen

Zeigt wie mit der innovativen Ports-Sammlung oder mit Paketen Software von Fremdherstellern installiert wird.

Kapitel 6, Das X Window System

Beschreibt allgemein das X Window System und geht speziell auf X11 unter FreeBSD ein. Weiterhin werden grafische Benutzeroberflächen wie **KDE** und **GNOME** behandelt.

Kapitel 7, Desktop-Anwendungen

Enthält eine Aufstellung verbreiteter Anwendungen wie Browser, Büroanwendungen und Office-Pakete und beschreibt wie diese Anwendungen installiert werden.

Kapitel 8, Multimedia

Erklärt, wie Sie auf Ihrem System Musik und Videos abspielen können. Beispielhaft werden auch Anwendungen aus dem Multimedia-Bereich beleuchtet.

Kapitel 9, Konfiguration des FreeBSD-Kernels

Erklärt, warum Sie einen angepassten Kernel erzeugen sollten und gibt ausführliche Anweisungen wie Sie einen angepassten Kernel konfigurieren, bauen und installieren.

Kapitel 10, Drucken

Beschreibt, wie Sie Drucker unter FreeBSD verwalten. Diskutiert werden Deckblätter, das Einrichten eines Druckers und ein Abrechnungssystem für ausgedruckte Seiten.

Kapitel 11, Linux-Binärkompatibilität

Beschreibt die binäre Kompatibilität zu Linux. Weiterhin werden ausführliche Installationsanleitungen für **Oracle**, **SAP® R/3®** und **Mathematica** gegeben.

Kapitel 12, Konfiguration und Tuning

Beschreibt die Einstellungen, die ein Systemadministrator vornehmen kann, um die Leistungsfähigkeit eines FreeBSD Systems zu verbessern. In diesem Kapitel werden auch verschiedene Konfigurationsdateien besprochen.

Kapitel 13, FreeBSDs Bootvorgang

Erklärt den Bootprozess von FreeBSD und beschreibt die Optionen, mit denen sich der Bootprozess beeinflussen lässt.

Kapitel 14, Benutzer und grundlegende Account-Verwaltung

Beschreibt, wie Benutzer-Accounts angelegt, verändert und verwaltet werden. Weiterhin wird beschrieben, wie dem Benutzer zur Verfügung stehende Ressourcen beschränkt werden können.

Kapitel 15, Sicherheit

Beschreibt die Werkzeuge mit denen Sie Ihr FreeBSD-System absichern. Unter Anderem werden Kerberos, IPsec und OpenSSH besprochen.

Kapitel 16, Jails

Dieses Kapitel beschreibt das Jails-Framework sowie die Vorteile von Jails gegenüber der traditionellen chroot-Unterstützung von FreeBSD.

Kapitel 17, Mandatory Access Control

Erklärt vorgeschriebene Zugriffskontrollen (MACs) und wie mit ihrer Hilfe FreeBSD-Systeme gesichert werden.

Kapitel 18, Security Event Auditing

Beschreibt, was FreeBSD Event Auditing ist, wie Sie diese Funktion installieren und konfigurieren und die damit erzeugten Audit-Trails überwachen und auswerten können.

Kapitel 19, Speichermedien

Erläutert den Umgang mit Speichermedien und Dateisystemen. Behandelt werden Plattenlaufwerke, RAID-Systeme, optische Medien, Bandlaufwerke, RAM-Laufwerke und verteilte Dateisysteme.

Kapitel 20, GEOM

Beschreibt das GEOM-Framework von FreeBSD sowie die Konfiguration der verschiedenen unterstützten RAID-Level.

Kapitel 21, File Systems Support

Beschreibt die Unterstützung nicht-nativer Dateisysteme (beispielsweise des Z-Dateisystems (zfs) von Sun) durch FreeBSD.

Kapitel 22, Vinum

Beschreibt den Vinum Volume Manager, der virtuelle Laufwerke, RAID-0, RAID-1 und RAID-5 auf Software-Ebene bereitstellt.

Kapitel 23, Virtualisierung

Dieses Kapitel beschreibt verschiedene Virtualisierungslösungen und wie diese mit FreeBSD zusammenarbeiten.

Kapitel 24, Lokalisierung

Zeigt wie Sie FreeBSD mit anderen Sprachen als Englisch einsetzen. Es wird sowohl die Lokalisierung auf der System-Ebene wie auch auf der Anwendungs-Ebene betrachtet.

Kapitel 25, FreeBSD aktualisieren

Erklärt die Unterschiede zwischen FreeBSD-STABLE, FreeBSD-CURRENT und FreeBSD-Releases. Das Kapitel enthält Kriterien anhand derer Sie entscheiden können, ob es sich lohnt, ein Entwickler-System zu installieren und aktuell zu halten. Außerdem wird beschrieben, wie Sie Ihr System durch das Einspielen neuer Sicherheits-Patches absichern.

Kapitel 26, DTrace

Beschreibt, wie das von Sun entwickelte DTrace-Werkzeug unter FreeBSD konfiguriert und eingesetzt werden kann. Dynamisches Tracing kann Ihnen beim Aufspüren von Leistungsproblemen helfen, indem Sie Echtzeit-Systemanalysen durchführen.

Kapitel 27, Serielle Datenübertragung

Erläutert, wie Sie Terminals und Modems an Ihr FreeBSD-System anschließen und sich so ein- und auswählen können.

Kapitel 28, PPP und SLIP

Erklärt wie Sie mit PPP, SLIP oder PPP über Ethernet ein FreeBSD-System mit einem entfernten System verbinden.

Kapitel 29, Elektronische Post (E-Mail)

Erläutert die verschiedenen Bestandteile eines E-Mail Servers und zeigt einfache Konfigurationen für **sendmail**, dem meist genutzten E-Mail-Server.

Kapitel 30, Netzwerkserver

Bietet ausführliche Informationen und Beispielkonfigurationen, die es Ihnen ermöglichen, Ihren FreeBSD-Rechner als *Network File System Server*, *Domain Name Server*, *Network Information Server*, oder als Zeitsynchronisationsserver einzurichten.

Kapitel 31, Firewalls

Erklärt die Philosophie hinter softwarebasierten Firewalls und bietet ausführliche Informationen zur Konfiguration der verschiedenen, für FreeBSD verfügbaren Firewalls.

Kapitel 32, Weiterführende Netzwerkthemen

Behandelt viele Netzwerkthemen, beispielsweise das Verfügbarmachen einer Internetverbindung für andere Rechner eines LANs, Routing, drahtlose Netzwerke, Bluetooth, IPv6, ATM und andere mehr.

Anhang A, Bezugsquellen für FreeBSD

Enthält eine Aufstellung der Quellen von denen Sie FreeBSD beziehen können: CD-ROM, DVD sowie Internet-Sites.

Anhang B, Bibliografie

Dieses Buch behandelt viele Themen und kann nicht alle Fragen erschöpfend beantworten. Die Bibliografie enthält weiterführende Bücher, die im Text zitiert werden.

Anhang C, Ressourcen im Internet

Enthält eine Aufstellung der Foren, die FreeBSD Benutzern für Fragen und Diskussionen zur Verfügung stehen.

Anhang D, PGP Schlüssel

Enthält PGP-Fingerabdrücke von etlichen FreeBSD Entwicklern.

Konventionen in diesem Buch

Damit der Text einheitlich erscheint und leicht zu lesen ist, werden im ganzen Buch die nachstehenden Konventionen beachtet:

Typographie

Kursiv

Für Dateinamen, URLs, betonte Teile eines Satzes und das erste Vorkommen eines Fachbegriffs wird ein *kursiver* Zeichensatz benutzt.

`Fixschrift`

Fehlermeldungen, Kommandos, Umgebungsvariablen, Namen von Ports, Hostnamen, Benutzernamen, Gruppennamen, Gerätenamen, Variablen und Code-Ausschnitte werden in einer `Fixschrift` dargestellt.

Fett

Fett kennzeichnet Anwendungen, Kommandozeilen und Tastensymbole.

Benutzereingaben

Tasten werden **fett** dargestellt, um sie von dem umgebenden Text abzuheben. Tasten, die gleichzeitig gedrückt werden müssen, werden durch ein + zwischen den einzelnen Tasten dargestellt:

Ctrl+**Alt**+**Del**

Im gezeigten Beispiel soll der Benutzer die Tasten **Ctrl**, **Alt** und **Del** gleichzeitig drücken.

Tasten, die nacheinander gedrückt werden müssen, sind durch Kommas getrennt:

Ctrl+**X**, **Ctrl**+**S**

Das letzte Beispiel bedeutet, dass die Tasten **Ctrl** und **X** gleichzeitig betätigt werden und danach die Tasten **Ctrl** und **S** gleichzeitig gedrückt werden müssen.

Beispiele

Beispiele, die durch `E:\>` eingeleitet werden, zeigen ein MS-DOS® Kommando. Wenn nichts Anderes angezeigt wird, können diese Kommandos unter neuen Versionen von Microsoft Windows auch in einem DOS-Fenster ausgeführt werden.

```
E:\> tools\fdimage floppies\kern.flp A:
```

Beispiele, die mit # beginnen, müssen unter FreeBSD mit Superuser-Rechten ausgeführt werden. Dazu melden Sie sich entweder als `root` an oder Sie wechseln von Ihrem normalen Account mit su(1) zu dem Benutzer `root`.

```
# dd if=kern.flp of=/dev/fd0
```

Beispiele, die mit % anfangen, werden unter einem normalen Benutzer-Account ausgeführt. Sofern nichts Anderes angezeigt wird, verwenden die Beispiele die Syntax der C-Shell.

```
% top
```

Danksagung

Dieses Buch ist aus Beiträgen von vielen Leuten aus allen Teilen der Welt entstanden. Alle eingegangen Beiträge, zum Beispiel Korrekturen oder vollständige Kapitel, waren wertvoll.

Einige Firmen haben dieses Buch dadurch unterstützt, dass Sie Autoren in Vollzeit beschäftigt und die Veröffentlichung des Buchs finanziert haben. Besonders BSDi (das später von Wind River Systems (http://www.windriver.com) übernommen wurde) beschäftigte Mitglieder des FreeBSD Documentation Projects, um dieses Buch zu erstellen. Dadurch wurde die erste (englische) gedruckte Auflage im März 2000 möglich (ISBN 1-57176-241-8). Wind River Systems bezahlte dann weitere Autoren, die die zum Drucken nötige Infrastruktur verbesserten und zusätzliche Kapitel beisteuerten. Das Ergebnis dieser Arbeit ist die zweite (englische) Auflage vom November 2001 (ISBN 1-57176-303-1). Zwischen 2003 und 2004 bezahlte FreeBSD Mall, Inc (http://www.freebsdmall.com) mehrere Mitarbeiter für die Vorbereitung der gedruckten dritten Auflage.

I. Erste Schritte

Dieser Teil des FreeBSD-Handbuchs richtet sich an Benutzer und Administratoren für die FreeBSD neu ist. Diese Kapitel

- geben Ihnen eine Einführung in FreeBSD,
- geleiten Sie durch den Installationsprozess,
- erklären Ihnen die Grundlagen von UNIX Systemen,
- zeigen Ihnen, wie Sie die Fülle der erhältlichen Anwendungen Dritter installieren und
- führen Sie in X, der Benutzeroberfläche von UNIX Systemen ein. Es wird gezeigt, wie Sie den Desktop konfigurieren, um effektiver arbeiten zu können.

Wir haben uns bemüht, Referenzen auf weiter vorne liegende Textteile auf ein Minimum zu beschränken, so dass Sie diesen Teil des Handbuchs ohne viel Blättern durcharbeiten können.

Kapitel 1. Einführung

Neu zusammengestellt, umstrukturiert und um Abschnitte erweitert durch Jim Mock. Übersetzt von Sascha Edelburg.

1.1. Übersicht

Herzlichen Dank für Ihr Interesse an FreeBSD! Das folgende Kapitel behandelt verschiedene Aspekte des FreeBSD Projects wie dessen geschichtliche Entwicklung, dessen Ziele oder dessen Entwicklungsmodell.

Nach dem Durcharbeiten des Kapitels wissen Sie über folgende Punkte Bescheid:

- Wo FreeBSD im Vergleich zu anderen Betriebssystemen steht
- Die Geschichte des FreeBSD Projects
- Die Ziele des FreeBSD Projects
- Die Grundlagen des FreeBSD-Open-Source-Entwicklungsmodells
- Und natürlich wo der Name “FreeBSD” herrührt

1.2. Willkommen bei FreeBSD!

FreeBSD ist ein auf 4.4BSD-Lite basierendes Betriebssystem für Intel (x86 und Itanium®), AMD64 und Sun UltraSPARC® Rechner. An Portierungen zu anderen Architekturen wird derzeit gearbeitet. Mehr zu Geschichte von FreeBSD können Sie im kurzen geschichtlichen Abriss zu FreeBSD oder im Abschnitt Das aktuelle FreeBSD-Release nachlesen. Falls Sie das FreeBSD Project unterstützen wollen (mit Quellcode, Hardware- oder Geldspenden), sollten Sie den Artikel FreeBSD unterstützen (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/articles/contributing/index.html) lesen.

1.2.1. Was kann FreeBSD?

FreeBSD hat zahlreiche bemerkenswerte Eigenschaften. Um nur einige zu nennen:

- *Präemptives Multitasking* mit dynamischer Prioritätsanpassung zum reibungslosen und ausgeglichenen Teilen der Systemressourcen zwischen Anwendungen und Anwendern, selbst unter schwerster Last.
- Der *Mehrbenutzerbetrieb* von FreeBSD erlaubt es, viele Anwender gleichzeitig am System mit verschiedenen Aufgaben arbeiten zu lassen. Beispielsweise Geräte wie Drucker oder Bandlaufwerke, die sich nur schwerlich unter allen Anwendern des Systems oder im Netzwerk teilen lassen, können durch Setzen von Verwendungsbeschränkungen auf Benutzer oder Benutzergruppen wichtige Systemressourcen vor Überbeanspruchung schützen.
-

Hervorragende *TCP/IP-Netzwerkfähigkeit* mit Unterstützung von Industriestandards wie SCTP, DHCP, NFS, NIS, PPP, SLIP, IPsec und IPv6. Das heißt, Ihr FreeBSD-System kann in einfachster Weise mit anderen Systemen interagieren. Zudem kann es als Server-System im Unternehmen wichtige Aufgaben übernehmen, beispielsweise als NFS- oder E-Mail-Server oder es kann Ihren Betrieb durch HTTP- und FTP-Server beziehungsweise durch Routing und Firewalling Internet-fähig machen.

- Der *Speicherschutz* stellt sicher, dass Anwendungen (oder Anwender) sich nicht gegenseitig stören. Stürzt eine Anwendung ab, hat das keine Auswirkung auf andere Prozesse.
- FreeBSD ist ein *32-Bit*-Betriebssystem (*64-Bit* auf Itanium, AMD64, und UltraSPARC) und wurde als solches von Grund auf neu entworfen.
- Das *X-Window-System* (X11R7) als Industriestandard bietet eine grafische Benutzeroberfläche (GUI). Minimale Voraussetzung zur Verwendung ist lediglich eine Grafikkarte und ein Bildschirm, die beide den VGA-Modus unterstützen.
- *Binärkompatibilität* mit vielen unter verschiedenen Betriebssystemen erstellten Programmen wie Linux, SCO, SVR4, BSDI und NetBSD.
- Tausende von *sofort lauffähigen* Anwendungen sind aus den *Ports*- und *Packages*-Sammlungen für FreeBSD verfügbar. Warum mühselig im Netz Software suchen, wenn sie bereits hier vorhanden ist?
- Tausende zusätzliche *leicht zu portierende* Anwendungen sind über das Internet zu beziehen. FreeBSD ist Quellcode-kompatibel mit den meisten kommerziellen UNIX Systemen. Daher bedürfen Anwendungen häufig nur geringer oder gar keiner Anpassung, um auf einem FreeBSD-System zu kompilieren.
- Seitenweise anforderbarer *Virtueller Speicher* und der “merged VM/buffer cache”-Entwurf bedient effektiv den großen Speicherhunger mancher Anwendungen bei gleichzeitigem Aufrechterhalten der Bedienbarkeit des Systems für weitere Benutzer.
- *SMP*-Unterstützung für Mehrprozessorsysteme
- Ein voller Satz von *C*, *C++* und *Fortran*- Entwicklungswerkzeugen. Viele zusätzliche Programmiersprachen für Wissenschaft und Entwicklung sind aus der Ports- und Packages-Sammlung zu haben.
- *Quellcode* für das gesamte System bedeutet größtmögliche Kontrolle über Ihre Umgebung. Warum sollte man sich durch proprietäre Lösungen knebeln und sich auf Gedeih und Verderb der Gnade eines Herstellers ausliefern, wenn man doch ein wahrhaft offenes System haben kann?
- Umfangreiche *Online-Dokumentation*.

FreeBSD basiert auf dem 4.4BSD-Lite-Release der Computer Systems Research Group (CSRG) der Universität von Kalifornien in Berkeley und führt die namhafte Tradition der Entwicklung von BSD-Systemen fort. Zusätzlich zu der herausragenden Arbeit der CSRG hat das FreeBSD Project tausende weitere Arbeitsstunden investiert, um das System zu verfeinern und maximale Leistung und Zuverlässigkeit bei Alltagslast zu bieten. Während viele

kommerzielle Riesen Probleme haben PC-Betriebssysteme mit derartigen Funktionen, Leistungpotential und Zuverlässigkeit anzubieten, kann FreeBSD damit schon *jetzt* aufwarten!

Die Anwendungsmöglichkeiten von FreeBSD werden nur durch Ihre Vorstellungskraft begrenzt. Von Software-Entwicklung bis zu Produktionsautomatisierung, von Lagerverwaltung über Abweichungskorrektur bei Satelliten; Falls etwas mit kommerziellen UNIX Produkten machbar ist, dann ist es höchstwahrscheinlich auch mit FreeBSD möglich. FreeBSD profitiert stark von tausenden hochwertigen Anwendungen aus wissenschaftlichen Instituten und Universitäten in aller Welt. Häufig sind diese für wenig Geld oder sogar kostenlos zu bekommen. Kommerzielle Anwendungen sind ebenso verfügbar und es werden täglich mehr.

Durch den freien Zugang zum Quellcode von FreeBSD ist es in unvergleichbarer Weise möglich, das System für spezielle Anwendungen oder Projekte anzupassen. Dies ist mit den meisten kommerziellen Betriebssystemen einfach nicht möglich. Beispiele für Anwendungen, die unter FreeBSD laufen, sind:

- *Internet-Dienste*: Die robuste TCP/IP-Implementierung in FreeBSD macht es zu einer idealen Plattform für verschiedenste Internet-Dienste, wie zum Beispiel:
  - FTP-Server
  - HTTP-Server (Standard-Web-Server oder mit SSL-Verschlüsselung)
  - IPv4- und IPv6-Routing
  - Firewalls und NAT-Gateways (“IP-Masquerading”)
  - E-Mail-Server
  - Usenet-News und Foren (BBS)

  Zum Betreiben von FreeBSD reicht schon ein günstiger 386-PC. Wenn es das Wachstum Ihres Unternehmens verlangt, kann FreeBSD aber auch auf einem hochgerüsteten 4-Wege-System mit Xeon-Prozessoren und RAID-Plattenspeicher Verwendung finden.
- *Bildung*: Sind Sie Informatikstudent oder Student eines verwandten Studiengangs? Die praktischen Einblicke in FreeBSD sind die beste Möglichkeit etwas über Betriebssysteme, Rechnerarchitektur und Netzwerke zu lernen. Einige frei erhältliche CAD-, mathematische und grafische Anwendungen sind sehr nützlich, gerade für diejenigen, die FreeBSD nicht zum Selbstzweck, sondern als *Arbeitsmittel* einsetzen.
- *Wissenschaft*: Mit dem frei verfügbaren Quellcode für das gesamte System bildet FreeBSD ein exzellentes Studienobjekt in der Disziplin der Betriebssysteme, wie auch in anderen Zweigen der Informatik. Es ist beispielsweise denkbar, das räumlich getrennte Gruppen gemeinsam an einer Idee oder Entwicklung arbeiten. Das Konzept der freien Verfügbarkeit und -nutzung von FreeBSD ermöglicht so einen Gebrauch, auch ohne sich groß Gedanken über Lizenzbedingungen oder -beschränkungen machen zu müssen.
- *Netzwerkfähigkeit*: Brauchen Sie einen neuen Router? Oder einen Name-Server (DNS)? Eine Firewall zum Schutze Ihres Intranets vor Fremdzugriff? FreeBSD macht aus dem in der Ecke verstaubenden 386- oder 486-PC im Handumdrehen einen leistungsfähigen Router mit anspruchsvollen Packet-Filter-Fähigkeiten.

- *X-Window-Workstation*: FreeBSD ist eine gute Wahl für kostengünstige X-Terminals mit dem frei verfügbaren X11-Server. Im Gegensatz zu einem X-Terminal erlaubt es FreeBSD, viele Anwendungen lokal laufen zu lassen, was die Last eines zentralen Servers erleichtern kann. FreeBSD kann selbst "plattenlos" starten, was einzelne Workstations noch günstiger macht und die Wartung erleichtert.
- *Software-Entwicklung*: Das Standard-System von FreeBSD wird mit einem kompletten Satz an Entwicklungswerkzeugen bereitgestellt, unter anderem mit dem bekannten GNU C/C++-Kompiler und -Debugger.

FreeBSD ist sowohl in Form von Quellcode als auch in Binärform auf CD-ROM, DVD und über anonymous FTP erhältlich. Näheres zum Bezug von FreeBSD enthält Anhang A.

1.2.2. Wer benutzt FreeBSD?

FreeBSD dient als Plattform für Geräte und Produkte einiger der weltgrößten IT-Firmen, darunter:

- Apple (http://www.apple.com/)
- Cisco (http://www.cisco.com/)
- Juniper (http://www.juniper.net/)
- NetApp (http://www.netapp.com/)

Außerdem laufen einige der größten Internet-Auftritte unter FreeBSD, beispielsweise:

- Yahoo! (http://www.yahoo.com/)
- Yandex (http://www.yandex.ru/)
- Apache (http://www.apache.org/)
- Rambler (http://www.rambler.ru/)
- Sina (http://www.sina.com/)
- Pair Networks (http://www.pair.com/)
- Sony Japan (http://www.sony.co.jp/)

- Netcraft (http://www.netcraft.com/)
- NetEase (http://www.163.com/)
- Weathernews (http://www.wni.com/)
- TELEHOUSE America (http://www.telehouse.com/)
- Experts Exchange (http://www.experts-exchange.com/)

und viele andere.

1.3. Das FreeBSD Project

Der folgende Abschnitt bietet einige Hintergrundinformationen zum FreeBSD Project, einschließlich einem kurzen geschichtlichen Abriss, den Projektzielen und dem Entwicklungsmodell.

1.3.1. Kurzer geschichtlicher Abriss zu FreeBSD

Beigesteuert von Jordan Hubbard.

Das FreeBSD Project erblickte das Licht der Welt Anfang 1993 teils als Auswuchs des “Unofficial 386BSD Patchkit” unter der Regie der letzten drei Koordinatoren des Patchkits: Nate Williams, Rod Grimes und mir.

Unser eigentliches Ziel war es, einen zwischenzeitlichen Abzug von 386BSD zu erstellen, um ein paar Probleme zu beseitigen, die das Patchkit-Verfahren nicht lösen konnte. Einige von Ihnen werden sich in dem Zusammenhang noch an die frühen Arbeitstitel “386BSD 0.5” oder “386BSD Interim” erinnern.

386BSD war das Betriebssystem von Bill Jolitz. Dieses litt bis zu diesem Zeitpunkt heftig unter fast einjähriger Vernachlässigung. Als das Patchkit mit jedem Tag anschwoll und unhandlicher wurde, waren wir einhellig der Meinung, es müsse etwas geschehen. Wir entschieden uns Bill Jolitz zu helfen, indem wir den übergangsweise “bereinigten” Abzug zur Verfügung stellten. Diese Pläne wurden unschön durchkreuzt als Bill Jolitz plötzlich seine Zustimmung zu diesem Projekt zurückzog, ohne einen Hinweis darauf, was stattdessen geschehen sollte.

Es hat nicht lange gedauert zu entscheiden, dass das Ziel es wert war, weiterverfolgt zu werden, selbst ohne Bills Unterstützung. Also haben wir den von David Greenman geprägten Namen “FreeBSD” angenommen. Unsere anfänglichen Ziele setzten wir nach Rücksprache mit den damaligen Benutzern des Systems fest. Und als deutlich wurde, das Projekt würde möglicherweise Realität, nahm ich Kontakt mit Walnut Creek CDROM auf, mit einem Auge darauf, den Vertriebsweg für die vielen Missbegünstigten zu verbessern, die keinen einfachen Zugang zum Internet hatten. Walnut Creek CDROM unterstützte nicht nur die Idee des CD-ROM-Vertriebs, sondern stellte sogar dem Projekt einen Arbeitsrechner und eine schnelle Internetverbindung zur Verfügung. Ohne den beispiellosen Glauben von Walnut Creek CDROM in ein zu der Zeit absolut unbekanntes Projekt, gäbe es FreeBSD in der heutigen Form wohl nicht.

Die erste auf CD-ROM (und netzweit) verfügbare Veröffentlichung war FreeBSD 1.0 im Dezember 1993. Diese basierte auf dem Band der 4.3BSD-Lite (“Net/2”) der Universität von Kalifornien in Berkeley. Viele Teile stammten aus 386BSD und von der Free Software Foundation. Gemessen am ersten Angebot, war das ein ziemlicher Erfolg und wir ließen dem das extrem erfolgreiche FreeBSD 1.1 im Mai 1994 folgen.

Zu dieser Zeit formierten sich unerwartete Gewitterwolken am Horizont, als Novell und die Universität von Kalifornien in Berkeley (UCB) ihren langen Rechtsstreit über den rechtlichen Status des Berkeley Net/2-Bandes mit einem Vergleich beilegten. Eine Bedingung dieser Einigung war es, dass die UCB große Teile des Net/2-Quellcodes als “belastet” zugestehen musste, und dass diese Besitz von Novell sind, welches den Code selbst einige Zeit vorher von AT&T bezogen hatte. Im Gegenzug bekam die UCB den “Segen” von Novell, dass sich das 4.4BSD-Lite-Release bei seiner endgültigen Veröffentlichung als unbelastet bezeichnen darf. Alle Net/2-Benutzer sollten auf das neue Release wechseln. Das betraf auch FreeBSD. Dem Projekt wurde eine Frist bis Ende Juli 1994 eingeräumt, das auf Net/2-basierende Produkt nicht mehr zu vertreiben. Unter den Bedingungen dieser Übereinkunft war es dem Projekt noch erlaubt ein letztes Release vor diesem festgesetzten Zeitpunkt herauszugeben. Das war FreeBSD 1.1.5.1.

FreeBSD machte sich dann an die beschwerliche Aufgabe, sich Stück für Stück, aus einem neuen und ziemlich unvollständigen Satz von 4.4BSD-Lite-Teilen, wieder aufzubauen. Die “Lite”-Veröffentlichungen waren deswegen leicht, weil Berkeleys CSRG große Code-Teile, die für ein start- und lauffähiges System gebraucht wurden, aufgrund diverser rechtlicher Anforderungen entfernen musste und weil die 4.4-Portierung für Intel-Rechner extrem unvollständig war. Das Projekt hat bis November 1994 gebraucht diesen Übergang zu vollziehen, was dann zu dem im Netz veröffentlichten FreeBSD 2.0 und zur CD-ROM-Version (im späten Dezember) führte. Obwohl FreeBSD gerade die ersten Hürden genommen hatte, war dieses Release ein maßgeblicher Erfolg. Diesem folgte im Juni 1995 das robustere und einfacher zu installierende FreeBSD 2.0.5.

Im August 1996 veröffentlichten wir FreeBSD 2.1.5. Es schien unter ISPs und der Wirtschaft beliebt genug zu sein, ein weiteres Release aus dem 2.1-STABLE-Zweig zu rechtfertigen. Das war FreeBSD 2.1.7.1. Es wurde im Februar 1997 veröffentlicht und bildete das Ende des Hauptentwicklungszweiges 2.1-STABLE. Derzeit unterliegt dieser Zweig dem Wartungsmodus, das heißt, es werden nur noch Sicherheitsverbesserungen und die Beseitigung von kritischen Fehlern vorgenommen (RELENG_2_1_0).

FreeBSD 2.2 entsprang dem Hauptentwicklungszweig (“-CURRENT”) im November 1996 als RELENG_2_2-Zweig und das erste komplette Release (2.2.1) wurde im April 1997 herausgegeben. Weitere Veröffentlichungen des 2.2-Zweiges gab es im Sommer und Herbst 1997. Das letzte Release des 2.2-Zweiges bildete die Version 2.2.8, die im November 1998 erschien. Das erste offizielle 3.0-Release erschien im Oktober 1998 und läutete das Endes des 2.2-Zweiges ein.

Am 20. Januar 1999 teilte sich der Quellbaum in die Zweige 4.0-CURRENT und 3.X-STABLE. Auf dem 3.X-STABLE-Zweig wurden folgende Releases erstellt: 3.1 am 15. Februar 1999, 3.2 am 15. Mai 1999, 3.3 am 16. September 1999, 3.4 am 20. Dezember 1999 und 3.5 am 24. Juni 2000. Letzterem folgte ein paar Tage später das Release 3.5.1, welches einige akute Sicherheitslöcher von Kerberos stopfte und die letzte Veröffentlichung des 3.X-Zweiges darstellte.

Eine weitere Aufspaltung, aus dem der 4.X-STABLE-Zweig hervorging, erfolgte am 13. März 2000. Bisher gab es mehrere Veröffentlichungen aus diesem Zweig: 4.0-RELEASE erschien im März 2000. Das letzte Release, 4.11-RELEASE, erschien im Januar 2005.

Das lang erwartete 5.0-RELEASE wurde am 19. Januar 2003 veröffentlicht. Nach nahezu drei Jahren Entwicklungszeit brachte dieses Release die Unterstützung für Mehrprozessor-Systeme sowie für Multithreading. Mit diesem Release lief FreeBSD erstmalig auf den Plattformen UltraSPARC und `ia64`. Im Juni 2003 folgte 5.1-RELEASE. Das letzte 5.X-Release aus dem CURRENT-Zweig war 5.2.1-RELEASE, das im Februar 2004 veröffentlicht wurde.

Der Zweig RELENG_5 wurde im August 2004 erzeugt. Als erstes Release dieses Zweiges wurde 5.3-RELEASE veröffentlicht, bei dem es sich gleichzeitig auch um das erste 5-STABLE-Release handelte. Das aktuelle 5.5-RELEASE (dem keine RELENG_5-Versionen mehr folgen werden) erschien im Mai 2006.

Der Zweig RELENG_6 wurde im Juli 2005 erzeugt. 6.0-RELEASE, das erste Release des 6.X-Zweiges, wurde im November 2005 veröffentlicht. Das aktuelle 6.4-RELEASE (erschienen im November 2008) ist das letzte Release aus RELENG_6-Zweig. RELENG_6 ist der letzte Zweig, der die Alpha-Architektur noch unterstützt.

Der Zweig RELENG_7 wurde im Oktober 2007 erzeugt. 7.0-RELEASE, das erste Release des 7.X-Zweiges, wurde im Februar 2008 veröffentlicht. Das aktuelle 8.4-RELEASE (dem keine weiteren RELENG_7-Versionen folgen werden) erschien im Februar 2011.

Im August 2009 wurde der RELENG_8-Zweig angelegt. 8.0-RELEASE, das erste Release des 8.X-Zweiges, erschien im November 2009. Das aktuelle 9.1-RELEASE (dem weitere RELENG_8-Versionen folgen werden) wurde im Juli 2010 veröffentlicht.

Zurzeit werden Projekte mit langem Entwicklungshorizont im Zweig 9.X-CURRENT verfolgt, Schnappschüsse von 9.X auf CD-ROM (und natürlich im Netz) werden bei fortlaufender Entwicklung auf dem Snapshot-Server (ftp://current.FreeBSD.org/pub/FreeBSD/snapshots/) zur Verfügung gestellt.

1.3.2. Ziele des FreeBSD Projects

Beigesteuert von Jordan Hubbard.

Das FreeBSD Project stellt Software her, die ohne Einschränkungen für beliebige Zwecke eingesetzt werden kann. Viele von uns haben beträchtlich in Quellcode und Projekt investiert und hätten sicher nichts dagegen, hin und wieder ein wenig finanziellen Ausgleich dafür zu bekommen. Aber in keinem Fall bestehen wir darauf. Wir glauben unsere erste und wichtigste “Mission” ist es, Software für jeden Interessierten und zu jedem Zweck zur Verfügung zu stellen, damit die Software größtmögliche Verbreitung erlangt und größtmöglichen Nutzen stiftet. Das ist, glaube ich, eines der grundlegenden Ziele freier Software, welche wir mit größter Begeisterung unterstützen.

Der Code in unserem Quellbaum, der unter die General Public License (GPL) oder die Library General Public License (LGPL) fällt, stellt geringfügig mehr Bedingungen. Das aber vielmehr im Sinne von eingefordertem Zugriff, als das übliche Gegenteil der Beschränkungen. Aufgrund zusätzlicher Abhängigkeiten, die sich durch die Verwendung von GPL-Software bei kommerziellem Gebrauch ergeben, bevorzugen wir daher Software unter dem transparenteren BSD-Copyright, wo immer es angebracht ist.

1.3.3. Das Entwicklungsmodell von FreeBSD

Beigesteuert von Satoshi Asami.

Die Entwicklung von FreeBSD ist ein offener und vielseitiger Prozess. FreeBSD besteht aus Beisteuerungen von Hunderten Leuten rund um die Welt, wie Sie aus der Liste der Beitragenden (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html) ersehen können. Die vielen Entwickler können aufgrund der Entwicklungs-Infrastruktur von FreeBSD über das Internet zusammenarbeiten. Wir suchen ständig nach neuen Entwicklern, Ideen und jenen, die sich in das Projekt tiefer einbringen wollen. Nehmen Sie einfach auf der Mailingliste FreeBSD technical discussions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers) Kontakt mit uns auf. Die Mailingliste FreeBSD announcements (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce) steht für wichtige Ankündigungen, die alle FreeBSD-Benutzer betreffen, zur Verfügung.

Unabhängig davon ob Sie alleine oder mit anderen eng zusammen arbeiten, enthält die folgende Aufstellung nützliche Informationen über das FreeBSD Project und dessen Entwicklungsabläufe.

CVS- und SVN-Repositories

Der Hauptquellbaum von FreeBSD wurde über viele Jahre ausschließlich mit CVS (http://ximbiot.com/cvs/wiki/) gepflegt, einem frei erhältlichen Versionskontrollsystem, welches mit FreeBSD geliefert wird. Im Juni 2008 begann das FreeBSD Project mit dem Umstieg auf SVN (http://subversion.tigris.org) (Subversion). Dieser Schritt wurde notwendig, weil **CVS** aufgrund des rapide wachsenden Quellcodebaumes und dem Umfang der bereits gespeichterten Revisisionsinformationen an seine Grenzen zu stoßen begann. Während das Hauptrepository nun **SVN** verwendet, hat sich auf der Client-Seite nichts geändert. Werkzeuge wie **CVSup** und **csup**, die auf der alten **CVS**-Infrastruktur aufbauen, funktionieren weiterhin, weil alle Änderungen, die im **SVN**-Repository erfolgen, in das **CVS**-Repository portiert werden. Im Moment wird nur src-Quellcodebaum über **SVN** verwaltet. Die Dokumentation, die Webseiten sowie die Ports befinden sich weiterhin in einem **CVS**-Repository. Das Haupt-CVS-Repository (http://www.FreeBSD.org/cgi/cvsweb.cgi) läuft auf einer Maschine in Santa Clara, Kalifornien, USA. Von dort wird es auf zahlreiche Server in aller Welt gespiegelt. Der **SVN**-Quellbaum, der die Zweige -CURRENT und -STABLE enthält, kann so einfach auf Ihr eigenes System gespiegelt werden. Näheres dazu können Sie im Handbuch unter Synchronisation der Quellen in Erfahrung bringen.

Die Committer-Liste

Die *Committer* sind Personen mit *Schreibzugriff* auf den CVS-Quellbaum (der Begriff “Committer” stammt vom cvs(1)-Befehl `commit`, der zum Einspeisen von Änderungen ins Repository gebraucht wird). Der beste Weg, Vorschläge zur Prüfung durch die Mitglieder der Committer-Liste einzureichen, bietet der Befehl send-pr(1). Sollte es unerwartete Probleme mit diesem Verfahren geben, besteht immer noch die Möglichkeit eine E-Mail an die Liste “FreeBSD committers” zu schicken.

Das FreeeBSD-Core-Team

Würde man das FreeBSD Project mit einem Unternehmen vergleichen, so wäre das *FreeBSD-Core-Team* das Gegenstück zum Vorstand. Die Hauptaufgabe des Core-Teams ist es, das Projekt als Ganzes in gesunder Verfassung zu halten und die weitere Entwicklung in die richtige Bahn zu lenken. Das Anwerben leidenschaftlicher und verantwortungsbewusster Entwickler ist eine Aufgabe des Core-Team, genauso wie die Rekrutierung neuer Mitglieder für das Core-Team, im Falle, dass Altmitglieder aus dem Projekt aussteigen. Das derzeitige Core-Team wurde im Juli 2010 aus einem Kreis kandidierender Committer gewählt. Wahlen werden alle zwei Jahre abgehalten.

Einige Core-Team-Mitglieder haben auch spezielle Verantwortungsbereiche. Das bedeutet, sie haben sich darauf festgelegt, sicherzustellen, dass ein größerer Teil des Systems so funktioniert wie ausgewiesen. Eine vollständige Liste an FreeBSD beteiligter Entwickler und ihrer Verantwortungsbereiche kann in der Liste der Beitragenden (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html) eingesehen werden.

Anmerkung: Die Mehrzahl der Mitglieder des Core-Teams sind Freiwillige in Bezug auf die FreeBSD-Entwicklung und profitieren nicht finanziell vom Projekt. Daher sollte “Verpflichtung” nicht als

> “garantierter Support” fehlinterpretiert werden. Der oben angeführte Vergleich mit einem Vorstand hinkt und es wäre angebrachter zu erwähnen, dass diese Leute – wider besseres Wissen – ihr eigenes Leben für FreeBSD aufgegeben haben!

Weitere Beitragende

Die größte Entwicklergruppe sind nicht zuletzt die Anwender selbst, die Rückmeldungen und Fehlerbehebungen in einem anhaltend hohen Maße an uns senden. Der bevorzugte Weg an dem weniger zentralisierten Bereich der FreeBSD-Entwicklung teilzuhaben, ist die Möglichkeit sich bei der Liste FreeBSD technical discussions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers) anzumelden. Weitere Informationen über die verschiedenen FreeBSD-Mailinglisten erhalten Sie in Anhang C.

Die Liste der zu FreeBSD Beitragenden (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html) ist eine lange und wachsende. Also warum nicht selbst dort stehen, indem Sie gleich persönlich etwas zu FreeBSD beitragen?

Quellcode ist nicht der einzige Weg, etwas zum Projekt beizusteuern. Eine genauere Übersicht über offene Aufgaben finden Sie auf der FreeBSD-Web-Site (http://www.FreeBSD.org/index.html).

Zusammengefasst bildet unser Entwicklungsmodell einen losen Verbund konzentrischer Kreise. Das zentralisierte Modell ist auf die Bedürfnisse der *Anwender* zugeschnitten, mit der einfachen Möglichkeit eine zentrale Code-Basis zu verfolgen und möglichen neuen Beitragenden nicht das Leben zu erschweren! Unser Ziel ist es, ein stabiles Betriebssystem mit einer großen Zahl passender Programme zu bieten, die der Anwender leicht installieren und anwenden kann. Und dieses Modell funktioniert für diese Aufgabe ziemlich gut.

Das Einzige was wir von möglichen neuen Mitgliedern fordern, ist die gleiche Hingabe, mit der die jetzigen Mitglieder am dauerhaften Erfolg arbeiten!

1.3.4. Das aktuelle FreeBSD-Release

FreeBSD ist ein (mit vollem Quellcode und ein frei erhältliches) auf 4.4BSD-Lite-basierendes Release für Intel i386™, i486™, Pentium®, Pentium Pro, Celeron®, Pentium II, Pentium III, Pentium 4 (oder ein dazu kompatibler Prozessor), Xeon™, und Sun UltraSPARC Systeme. Es stützt sich zum größten Teil auf Software der Computer Systems Research Group (CSRG) der Universität von Kalifornien in Berkeley mit einigen Verbesserungen aus NetBSD, OpenBSD, 386BSD und der Free Software Foundation.

Seit unserem FreeBSD 2.0 von Ende 1994 haben sich Leistung, Funktionsvielfalt und Stabilität dramatisch verbessert. Die größte Änderung erfuhr das virtuelle Speichermanagement durch eine Kopplung von virtuellem Speicher und dem Buffer-Cache, das nicht nur die Leistung steigert, sondern auch den Hauptspeicherverbrauch reduziert und ein 5 MB-System zu einem nutzbaren Minimal-System verhilft. Weitere Verbesserungen sind volle NIS-Client- und Server-Unterstützung, T/TCP, Dial-On-Demand-PPP, integriertes DHCP, ein verbessertes SCSI-Subsystem, ISDN-Support, Unterstützung für ATM-, FDDI-, Fast- und Gigabit-Ethernet-Karten (1000 Mbit), verbesserter Support der neusten Adaptec-Controller und tausende Fehlerkorrekturen.

Zusätzlich zur Standard-Distribution bietet FreeBSD eine Sammlung von portierter Software mit tausenden begehrten Programmen. Zum Verfassungszeitpunkt waren über 24,000 Anwendungen in der Ports-Sammlung! Das Spektrum der Ports-Sammlung reicht von HTTP-Servern über Spiele, Programmiersprachen, Editoren und so

ziemlich allem dazwischen. Die gesamte Ports-Sammlung benötigt 500 MB an Speicherplatz, wobei jeder Port anhand eines “Deltas” zu den Quellen angegeben wird. Das macht es für uns erheblich leichter, Ports zu aktualisieren und es verringert den Plattenbedarf im Vergleich zur älteren 1.0-Port-Sammlung. Um ein Port zu übersetzen, müssen Sie einfach ins Verzeichnis des Programms wechseln und ein `make install` absetzen. Den Rest erledigt das System. Die originalen Quellen jedes zu installierenden Port werden dynamisch von CD-ROM oder einem FTP-Server bezogen. Es reicht also für genügend Plattenplatz zu sorgen, um die gewünschten Ports zu erstellen. Allen, die Ports nicht selbst kompilieren wollen: Es gibt zu fast jedem Port ein vorkompiliertes Paket, das einfach mit dem Befehl (`pkg_add`) installiert wird. Pakete und Ports werden in Kapitel 5 beschrieben.

Eine Reihe von weiteren Dokumenten, die sich als hilfreich bei der Installation oder dem Arbeiten mit FreeBSD erweisen könnten, liegen auf neueren FreeBSD-Systemen im Verzeichnis `/usr/share/doc`. Die lokal installierten Anleitungen lassen sich mit jedem HTML-fähigen Browser unter folgenden Adressen betrachten:

Das FreeBSD-Handbuch

```
/usr/share/doc/handbook/index.html
```

Die FreeBSD-FAQ

```
/usr/share/doc/faq/index.html
```

Es besteht auch die Möglichkeit, sich die jeweils aktuellste Version der Referenzdokumente auf der FreeBSD-Homepage (http://www.FreeBSD.org/de/index.html) anzusehen.

Kapitel 2. FreeBSD 8.*x* (und älter) installieren

Überarbeitet und teilweise neu geschrieben von Jim Mock. Der Gang durch sysinstall und alle Bildschirmabzüge von Randy Pratt. Übersetzt von Martin Heinen und Johann Kois.

2.1. Übersicht

FreeBSD wird mit dem textorientierten und einfach zu benutzendem Installationsprogramm installiert. Beginnend mit FreeBSD 9.0-RELEASE handelt es sich dabei um das Programm **bsdinstall**. Ältere FreeBSD-Versionen verwenden hingegen nach wie vor **sysinstall** für die Installation. Dieses Kapitel beschreibt die Installation von FreeBSD über **sysinstall**. Der Einsatz von **bsdinstall** wird hingegen in Kapitel 3 besprochen.

Dieses Kapitel behandelt folgende Punkte:

- Das Erzeugen von FreeBSD-Startdisketten.
- Wie FreeBSD Platten anspricht und aufteilt.
- Wie **sysinstall** ausgeführt wird.
- Die Menüs von **sysinstall** und die erforderlichen Eingaben in den Menüs.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Hardware-Notes der FreeBSD-Release, die Sie installieren wollen, lesen und sicherstellen, dass Ihre Hardware unterstützt wird.

> **Anmerkung:** Diese Installationsanleitung gilt für Rechner mit i386-Architektur (PC-kompatible Rechner). Abweichende Anweisungen für andere Plattformen werden, falls notwendig, gegeben. Obwohl diese Anleitung so aktuell wie möglich ist, kann das Installationsverfahren von dem hier gezeigten geringfügig abweichen. Legen Sie bitte daher diese Anleitung nicht wortwörtlich aus, sondern lassen Sie sich von diesem Kapitel durch den Installationsprozess leiten.

2.2. Hardware-Anforderungen

2.2.1. Minimalkonfiguration

Die zur Installation von FreeBSD erforderliche Minimalkonfiguration hängt von der zu installierenden FreeBSD-Version sowie von der Hardware-Architektur ab.

Informationen zur jeweiligen Minimalkonfiguration finden Sie in den folgenden Abschnitten dieses Kapitels. Je nachdem, wie Sie FreeBSD installieren, benötigen Sie eventuell auch ein Diskettenlaufwerk, ein unterstütztes CD-ROM-Laufwerk, oder auch eine Netzwerkkarte. Abschnitt 2.3.7 des Handbuchs enthält weitere Informationen zu den verschiedenen Installationsarten.

2.2.1.1. Die FreeBSD/i386- und FreeBSD/pc98-Architekturen

Sowohl FreeBSD/i386 als auch FreeBSD/pc98 benötigen jeweils mindestens einen 486-Prozessor sowie mindestens 24 MB RAM. Außerdem benötigen Sie für eine Minimalinstallation mindestens 150 MB freien Platz auf Ihrer Festplatte.

> **Anmerkung:** In den meisten derartigen Konfigurationen ist es besser, für mehr RAM und mehr Plattenplatz zu sorgen, statt einen schnelleren Prozessor einzubauen.

2.2.1.2. Die FreeBSD/amd64-Architektur

Es gibt zwei Klassen von Prozessoren, auf denen Sie FreeBSD/amd64 ausführen können. Die erste Klasse bilden die AMD64-Prozessoren (zu denen AMD Athlon 64-, AMD Athlon 64-FX-, oder AMD Opteron-Prozessoren gehören).

Die zweite Klasse von Prozessoren, auf denen Sie diese FreeBSD/amd64 einsetzen können, ist die Intel® EM64T-Architektur. Prozessoren dieser Klasse sind beispielsweise Intel Core™ 2 Duo-, Quad-, und Extreme-Prozessoren sowie die Intel Xeon-Prozessorreihen 3000, 5000, und 7000.

Wenn Sie einen auf dem Chipsatz nVidia nForce3 Pro-150 basierenden Rechner haben, *müssen* Sie im BIOS das IO-APIC deaktivieren. Erlaubt ihr BIOS dies nicht, müssen Sie stattdessen ACPI deaktivieren. Der Grund dafür sind Fehler im Pro-150-Chipsatz, die bis jetzt nicht behoben werden konnten.

2.2.1.3. Die FreeBSD/sparc64-Architektur

Um FreeBSD/sparc64 zu installieren, benötigen Sie eine unterstützte Plattform (lesen Sie dazu auch Abschnitt 2.2.2 des Handbuchs).

Sie benötigen außerdem eine seperate Festplatte, wenn Sie FreeBSD/sparc64 installieren wollen, da es derzeit leider noch nicht möglich ist, die Platte mit einem weiteren Betriebssystem zu teilen.

2.2.2. Unterstützte Hardware

Die Hardware-Notes, die mit jedem FreeBSD-Release ausgeliefert werden, enthalten eine Liste lauffähiger Hardware. Die Hardware-Notes befinden sich üblicherweise in der Datei `HARDWARE.TXT` im Wurzelverzeichnis der Distribution (CD-ROM oder FTP). Sie können die Hardware-Notes außerdem im Dokumentationsmenü von **sysinstall** oder auf der Webseite Release Information (http://www.FreeBSD.org/de/releases/index.html) lesen.

2.3. Vor der Installation

2.3.1. Erstellen Sie eine Geräteliste

Bevor Sie FreeBSD installieren, erfassen Sie die Komponenten Ihres Rechners. Die FreeBSD-Installation wird die Komponenten (Festplatten, Netzwerkkarten, CD-ROM-Laufwerke) zusammen mit der Modellbezeichnung und des Herstellers anzeigen. FreeBSD wird auch versuchen, die richtige Konfiguration der Geräte zu ermitteln. Dazu

gehören die benutzten Interrupts (IRQ) und IO-Ports. Wegen der Unwägbarkeiten von PC-Hardware kann die Konfiguration der Geräte allerdings fehlschlagen. In diesem Fall müssen Sie die von FreeBSD ermittelte Konfiguration korrigieren.

Wenn Sie schon ein anderes Betriebssystem, wie Windows oder Linux installiert haben, können Sie die Hardware-Konfiguration mit den Mitteln dieses Betriebssystems bestimmen. Wenn Sie nicht sicher sind, welche Einstellungen eine Erweiterungskarte besitzt, sehen Sie auf der Karte selbst nach. Manchmal sind die Einstellungen dort aufgedruckt. Gebräuchliche IRQs sind `3`, `5` und `7`. Die Adressen von IO-Ports werden normalerweise hexadezimal, zum Beispiel `0x330`, angegeben.

Halten Sie die Gerätekonfiguration vor der Installation in einer Tabelle wie der nachstehenden fest:

Tabelle 2-1. Gerätekonfiguration

Gerät	IRQ	IO-Ports	Anmerkung
erste Festplatte	-	-	40 GB, Seagate, erster IDE-Master
CD-ROM	-	-	erster IDE-Slave
zweite Festplatte	-	-	20 GB, IBM, zweiter IDE-Master
erster IDE-Controller	14	0x1f0	
Netzwerkkarte	-	-	Intel 10/100
Modem	-	-	3Com® 56K Faxmodem, an COM1
…			

Nachdem Sie wissen, über welche Hardware Ihr Rechner verfügt, müssen Sie diese Informationen mit den Hardwareanforderungen der zu installierenden FreeBSD-Version abgleichen.

2.3.2. Sichern Sie Ihre Daten

Wenn der Rechner, auf dem Sie FreeBSD installieren wollen, wichtige Daten enthält, sichern Sie bitte diese Daten. Prüfen Sie auch, dass Sie die Daten aus der Sicherung wiederherstellen können, bevor Sie FreeBSD installieren. Die FreeBSD-Installation fragt zwar nach, bevor Sie Daten auf Ihre Festplatte schreibt, Ihre Daten sind allerdings unwiderruflich verloren, wenn der Installationsvorgang einmal angelaufen ist.

2.3.3. Den Installationsort von FreeBSD festlegen

Wenn Sie die gesamte Festplatte für FreeBSD verwenden wollen, müssen Sie sich an dieser Stelle keine weiteren Gedanken machen – lesen Sie bitte im nächsten Abschnitt weiter.

Wenn Sie allerdings FreeBSD neben anderen Betriebssystemen betreiben wollen, müssen Sie wissen, wie Daten auf einer Festplatte abgelegt werden und welche Auswirkungen dies hat.

2.3.3.1. Platteneinteilung von FreeBSD/i386-Systemen

Eine PC-Festplatte wird in einzelne Bereiche unterteilt, die *Partitionen* heißen. FreeBSD verwendet intern ebenfalls Partitionen. Um Verwechslungen und Unklarheiten zu vermeiden, werden diese Plattenbereiche unter FreeBSD als Slices bezeichnet. So verwendet beispielsweise das Werkzeug `fdisk` den Begriff Slices, um sich auf PC-Partitionen zu beziehen. Auf einer PC-Festplatte können maximal vier Partitionen, die *primäre Partitionen* genannt werden,

angelegt werden. Eine *erweiterte Partition* hebt diese Beschränkung auf. Eine Festplatte kann nur eine erweiterte Partition enthalten, die wiederum weitere so genannte *logische Partitionen* enthalten kann.

Jede Partition besitzt eine *Partitions-ID* – eine Zahl, die den Typ der Partition festlegt. FreeBSD-Partitionen tragen die Partitions-ID `165`.

Üblicherweise kennzeichnen Betriebssysteme Partitionen in einer besonderen Art und Weise. Beispielsweise werden jeder primären und logischen Partition unter MS-DOS und dem verwandten Windows Laufwerksbuchstaben beginnend mit `C:` zugewiesen.

FreeBSD muss auf einer primären Partition installiert werden. In dieser Partition hält FreeBSD alle Daten einschließlich der Dateien, die Sie anlegen. Verfügt das System über mehrere Festplatten, können Sie auf allen oder einigen Platten eine FreeBSD-Partition einrichten. Zur Installation von FreeBSD benötigen Sie eine freie Partition: Dies kann eine extra für die Installation eingerichtete Partition sein oder eine existierende Partition, die nicht mehr benötigte Daten enthält.

Wenn auf allen Platten bereits sämtliche Partitionen benutzt werden, müssen Sie eine der Partitionen für FreeBSD frei machen. Benutzen Sie dazu die Werkzeuge des eingesetzten Betriebssystems (`fdisk` unter MS-DOS oder Windows).

Verfügt das System über eine freie Partition, benutzen Sie diese Partition. Es kann allerdings sein, dass Sie eine oder mehrere der vorhandenen Partitionen vorher verkleinern müssen.

Eine minimale FreeBSD-Installation benötigt nur 100 MB Plattenplatz. Diese Installation ist allerdings *sehr* begrenzt und lässt wenig Platz für Ihre eigenen Dateien. Realistischer sind 250 MB für FreeBSD ohne graphische Benutzeroberfläche und 350 MB für FreeBSD mit einer graphischen Benutzeroberfläche. Sie benötigen weiteren Platz für die Installation zusätzlicher Software.

Um die Partitionen zu verkleinern, können Sie beispielsweise das kommerzielle **PartitionMagic®** oder das freie **GParted** benutzen. Sowohl **GParted** als auch **PartitionMagic** können auch NTFS-Partitionen verändern. **GParted** ist auf vielen Linux-Live-CDs, beispielsweise der SystemRescueCD (http://www.sysresccd.org/), verfügbar.

Bei der Veränderung von Microsoft Vista-Partionen kommt es manchmal zu Problemen. In einem solchen Fall ist es von Vorteil, wenn Sie eine Vista-Installations-CD zur Verfügung haben. Wie bei jeder Änderung an Ihrer Festplatte sollten Sie auch hier zuerst ein aktuelles Backup anlegen.

> **Warnung:** Der falsche Gebrauch dieser Werkzeuge kann Daten auf der Festplatte löschen. Vor dem Einsatz dieser Werkzeuge stellen Sie bitte sicher, dass Sie frische, funktionierende Datensicherungen besitzen.

Beispiel 2-1. Eine bestehende Partition verwenden

Nehmen wir an, Sie haben einen Rechner mit einer 4 GB Festplatte, auf der schon eine Version von Windows installiert ist. Weiterhin haben Sie die Platte in zwei Laufwerke `C:` und `D:` unterteilt, die jeweils 2 GB groß sind. Auf `C:` wird 1 GB benutzt und 0,5 GB von Laufwerk `D:` werden benutzt.

Sie haben also eine Festplatte mit zwei Partitionen und könnten alle Daten von Laufwerk `D:` auf das Laufwerk `C:` kopieren. Damit wäre die zweite Partition für FreeBSD frei.

Beispiel 2-2. Eine bestehende Partition verkleinern

Nehmen wir an, Sie haben einen Rechner mit einer 4 GB Festplatte auf der schon eine Version von Windows installiert ist. Während der Installation von Windows haben sie eine große Partition `C:` angelegt, die 4 GB groß ist. Von den 4 GB werden 1,5 GB benutzt und Sie wollen 2 GB für FreeBSD verwenden.

Sie haben zwei Möglichkeiten, FreeBSD zu installieren:

1. Sichern Sie die Daten der Windows-Partition und installieren Sie Windows erneut auf einer 2 GB großen Partition.
2. Verkleinern Sie die Windows-Partition mit einem der oben aufgeführten Werkzeuge.

2.3.4. Netzwerkparameter ermitteln

Wird während der Installation ein Netzwerk benötigt (weil Sie über FTP oder von einem NFS-Server installieren wollen), müssen Sie die Konfiguration des Netzwerks kennen. Während der Installation werden Netzwerkparameter abgefragt, damit sich FreeBSD mit dem Netzwerk verbinden und die Installation abschließen kann.

2.3.4.1. Verbindung über Ethernet oder ein Kabel/DSL-Modem

Wenn Sie sich mit einem Ethernet verbinden oder eine Internet-Verbindung mit einem Ethernet-Adapter über Kabel oder DSL herstellen, benötigen Sie die nachstehenden Daten:

1. IP-Adresse
2. IP-Adresse des Default-Gateways
3. Hostname
4. IP-Adressen der DNS-Server
5. Subnetzmaske

Wenn Sie die Daten nicht besitzen, fragen Sie bitte Ihren Systemadministrator oder Ihren Service-Provider. Können die Daten über *DHCP* bezogen werden, merken Sie sich diese Tatsache.

2.3.4.2. Verbindung über ein Modem

Auch wenn Sie sich mit einem normalen Modem bei einem ISP einwählen, können Sie FreeBSD aus dem Internet installieren. Die Installation über ein Modem dauert nur sehr lange.

Sie benötigen die nachstehenden Daten:

1. Die Telefonnummer des ISPs.
2. Die COM-Schnittstelle, an der das Modem angeschlossen ist.
3. Den Benutzernamen und das Passwort für Ihr Konto.

2.3.5. Lesen Sie die FreeBSD-Errata

Auch wenn das FreeBSD-Project bemüht ist, ein Release so stabil wie möglich herzustellen, treten ab und an Fehler auf. In seltenen Fällen betrifft ein Fehler die Installations-Prozedur. Die Fehler und deren Behebungen werden in den FreeBSD-Errata (http://www.FreeBSD.org/releases/9.1R/errata.html) festgehalten. Lesen Sie bitte die Errata, bevor Sie FreeBSD installieren, damit Sie nicht in frisch entdeckte Probleme laufen.

Dokumentation zu jedem Release, inklusive der Errata zu jedem Release, finden Sie im Release-Bereich (http://www.FreeBSD.org/de/releases/index.html) des FreeBSD Webauftritts (http://www.FreeBSD.org/index.html).

2.3.6. Die Installationsdateien beschaffen

FreeBSD kann von Dateien aus irgendeiner der nachstehenden Quellen installiert werden:

Lokale Medien

- von einer CD-ROM oder einer DVD
- von einem USB-Stick
- von einer MS-DOS-Partition auf demselben Rechner
- von einem SCSI- oder QIC-Bandlaufwerk
- von Disketten

Netzwerk

- von einem FTP-Server, wenn erforderlich auch durch eine Firewall oder durch einen HTTP-Proxy
- von einem NFS-Server
- über eine feste serielle oder eine feste parallele Verbindung

Wenn Sie eine FreeBSD-CD oder FreeBSD-DVD gekauft haben, besitzen Sie schon alles, was Sie zur Installation benötigen. Lesen Sie bitte im nächsten Abschnitt (Abschnitt 2.3.7) weiter.

Wenn Sie sich die FreeBSD-Installationsdateien noch nicht besorgt haben, lesen Sie bitte zuerst den Abschnitt 2.13. Dort werden die notwendigen Vorbereitungen für eine Installation von den eben genannten Medien beschrieben. Wenn Sie den Abschnitt durchgearbeitet haben, lesen Sie bitte in Abschnitt 2.3.7 weiter.

2.3.7. Das Startmedium vorbereiten

Um FreeBSD zu installieren, müssen Sie Ihren Rechner mit einem speziellen Startmedium hochfahren, das die Installationsroutine startet. Sie können das Installationsprogramm nicht unter einem anderen Betriebssystem ausführen. Ein Rechner startet normalerweise das auf der Festplatte installierte Betriebssystem, er kann aber auch von Disketten gestartet werden. Aktuelle Rechner können in der Regel auch von einer CD-ROM oder von einem USB-Stick starten.

> **Tipp:** Wenn Sie eine FreeBSD CD-ROM oder DVD besitzen (gekauft oder selbst erstellt) und Ihr Rechner von CD-ROM oder DVD starten kann (üblicherweise können Sie das mit der BIOS-Option `Boot Order` einstellen), können Sie diesen Abschnitt überspringen. Eine FreeBSD CD-ROM oder DVD lässt sich direkt starten; Sie können damit FreeBSD ohne weitere Vorbereitungen installieren.

Um einen bootbaren USB-Stick zu erstellen, gehen Sie wie folgt vor:

1. Das Speicherabbild für den USB-Stick herunterladen

   Das Speicherabbild finden Sie auf dem FreeBSD-FTP-Server `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/`*`arch`*`/ISO-IMAGES/`*`version`*`/FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-` (oder einem Spiegelserver) im Verzeichnis `ISO-IMAGES/`. Ersetzen Sie *`arch`* und *`version`* durch die von Ihnen verwendete Architektur und die FreeBSD-Version, die Sie einsetzen wollen. Für FreeBSD/i386 9.1-RELEASE finden Sie das Speicherabbild für den USB-Stick beispielsweise unter folgenden Link: ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/9.1/FreeBSD-9.1-RELEASE-i386-memstick.img.

   Das benötigte Speicherabbild hat den Dateityp `.img`. Das Verzeichnis `ISO-IMAGES/` enthält verschiedene Speicherabbilder. Sie müssen also (basierend auf der zu installierenden FreeBSD-Version und/oder Ihrer Hardware) das für Sie passende Speicherabbild herunterladen.

   > **Wichtig:** *Sichern Sie Ihre Daten*, bevor Sie fortfahren, da im nächsten Schritt alle auf dem USB-Stick befindlichen Daten *gelöscht* werden.

2. Den USB-Stick vorbereiten

   Den USB-Stick unter FreeBSD vorbereiten

   > **Warnung:** Das Beispiel im nächsten Schritt verwendet `/dev/da0` als die Gerätedatei, über die Sie den USB-Stick ansprechen. Achten Sie besonders darauf, dass Sie die richtige Gerätedatei verwenden, da Sie ansonsten unbeabsichtigt Daten löschen könnten.

   1. Das Image mit dd(1) auf den Stick schreiben

      Bei der `.img`-Datei handelt es sich *nicht* um eine normale Datei, die Sie einfach auf den Stick kopieren können. Vielmehr handelt es sich dabei um ein Image des kompletten Dateisystems, das Sie mit dd(1) direkt auf den USB-Stick schreiben müssen:

      ```
      # dd if=FreeBSD-9.1-RELEASE-i386-memstick.img of=/dev/da0 bs=64k
      ```

      Wird dabei die Fehlermeldung `Operation not permitted` angezeigt, stellen Sie bitte sicher, dass das Zielgerät nicht verwendet, manuell eingehängt oder von einem Systemprogramm automatisch eingehängt wurde. Dann versuchen Sie es erneut.

   Den USB-Stick unter Windows® vorbereiten

   > **Warnung:** Stellen Sie unbedingt sicher, dass Sie im folgenden Schritt den korrekten Laufwerkbuchstaben für Ihren USB-Stick angeben, da Sie ansonsten unbeabsichtigt Daten löschen könnten.

   1. **Image Writer für Windows** herunterladen

Image Writer für Windows ist ein frei verfügbares Programm, mit dem Sie ein Image auf einen USB-Stick schreiben können. Laden Sie das Programm von https://launchpad.net/win32-image-writer/ herunter und entpacken Sie es in einen Ordner auf Ihrer Festplatte.

2. Das Image mit Image Writer auf den Stick schreiben

   Klicken Sie doppelt auf das Symbol **Win32DiskImager**, um das Programm zu starten. Vergewissern Sie sich, dass es sich bei dem unter `Device` angezeigten Laufwerk um Ihren USB-Stick handelt. Danach klicken Sie auf das Ordnersymbol und wählen die zuvor heruntergeladene Image-Datei aus. Klicken Sie auf Save, um die Image-Datei zu laden. Nachdem Sie alle Eingaben nochmals geprüft haben, müssen Sie noch sicherstellen, dass kein anderes Programm auf den USB-Stick zugreift. Danach klicken Sie auf den Button Write, um das Image auf den USB-Stick zu schreiben.

Um Startdisketten zu erzeugen, benutzen Sie die nachstehende Anleitung:

1. Abbilder der Startdisketten besorgen

   Wichtig: Beachten Sie, dass ab FreeBSD 8.*X* Startdisketten nicht mehr unterstützt werden. Lesen Sie bitte weiter oben in diesem Kapitel, wie Sie FreeBSD von einer CD-ROM, einer DVD oder einem USB-Stick installieren können.

   Die Abbilder der Startdisketten befinden sich auf dem Installationsmedium im Verzeichnis `floppies/`; sie können auch aus dem Internet heruntergeladen werden: `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/`*`arch`*`/`*`version`*`-RELEASE/floppies/`. Ersetzen Sie *`arch`* und *`version`* durch die passende Architektur und die passende Version. Beispielsweise stehen die Startdisketten von FreeBSD/i386 8.4-RELEASE unter ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/8.4-RELEASE/floppies/.

   Die Abbilder besitzen die Dateinamenerweiterung `.flp`. Im Verzeichnis `floppies/` befinden sich verschiedene Abbilder; welches Sie benutzen, hängt von der zu installierenden FreeBSD-Version und in einigen Fällen vom Zielrechner ab. In den meisten Fällen werden Sie vier Disketten benötigen: `boot.flp`, `kern1.flp`, `kern2.flp` sowie `kern3.flp`. Lesen Sie bitte die Datei `README.TXT` im Verzeichnis `floppies/`, sie enthält aktuelle Informationen zu den Abbildern.

   Wichtig: Wenn Sie die Abbilder aus dem Internet herunterladen, benutzen Sie bitte den *Binärmodus* des FTP-Programms. Einige Web-Browser verwenden den *Textmodus* (oder *ASCII*-Modus), was dazu führt, dass sich die erstellten Disketten nicht starten lassen.

2. Die Disketten vorbereiten

   Pro Abbild benötigen Sie eine Diskette. Es ist wichtig, dass die verwendeten Disketten fehlerfrei sind. Sie können dies sicherstellen, indem Sie die Disketten selbst formatieren, verlassen Sie sich bitte nicht auf vorformatierte Disketten. Das Formatierprogramm von Windows zeigt fehlerhafte Blöcke nicht an, es markiert die Blöcke einfach als fehlerhaft und ignoriert sie dann. Benutzen Sie neue Disketten, wenn Sie diese Installationsart verwenden.

Wichtig: Wenn Sie FreeBSD installieren und das Installationsprogramm abstürzt, einfriert oder sich merkwürdig verhält, sind oft fehlerbehaftete Disketten die Ursache. Schreiben Sie die Abbilder auf neue Disketten und versuchen Sie, noch mal zu installieren.

3. Die Abbilder auf Disketten schreiben

   Die `.flp`-Dateien sind *keine* normalen Dateien, die Sie auf eine Diskette kopieren. Sie können die Abbilder *nicht* von einem Laufwerk auf ein anderes Laufwerk kopieren. Die Abbilder werden mit einem speziellen Werkzeug direkt auf die Diskette geschrieben.

   Wenn Sie die Startdisketten unter MS-DOS oder Windows erstellen, können Sie das mitgelieferte Werkzeug `fdimage` verwenden.

   Wenn Sie die Abbilder auf der CD-ROM verwenden und das CD-ROM-Laufwerk den Laufwerksbuchstaben `E:` besitzt, führen Sie den nachstehenden Befehl aus:

   ```
   E:\> tools\fdimage floppies\boot.flp A:
   ```

   Führen Sie das Kommando für jede `.flp`-Datei aus. Wechseln Sie bitte jedes Mal die Diskette und beschriften Sie die Diskette mit dem Namen der kopierten Datei. Falls Sie die Abbilder an anderer Stelle liegen haben, passen Sie bitte die Kommandozeile an. Wenn Sie keine CD-ROM besitzen, können Sie `fdimage` aus dem Verzeichnis `tools` (ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/) des FreeBSD-FTP-Servers herunterladen.

   Wenn Sie Startdisketten auf einem UNIX System (zum Beispiel einem anderen FreeBSD System) erstellen, schreiben Sie die Abbilder mit dem Befehl dd(1) direkt auf die Disketten. Auf einem FreeBSD-System lautet die Kommandozeile:

   ```
   # dd if=boot.flp of=/dev/fd0
   ```

   Unter FreeBSD spricht `/dev/fd0` das erste Diskettenlaufwerk an (das Laufwerk `A:`), `/dev/fd1` spricht das Laufwerk `B:` an. Andere UNIX Varianten verwenden unter Umständen andere Gerätenamen, die in der Dokumentation des jeweiligen Systems beschrieben sind.

Nun ist alles für die FreeBSD-Installation vorbereitet.

2.4. Die Installation starten

Wichtig: Die Installationsprozedur lässt die Daten auf Ihren Laufwerken solange unverändert bis die nachstehende Meldung erscheint:

```
Last Chance: Are you SURE you want continue the installation?

If you’re running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

We can take no responsibility for lost disk contents!
```

Vor dieser Meldung kann die Installationsprozedur jederzeit abgebrochen werden, ohne die Daten auf der Festplatte zu verändern. Wenn Sie meinen, etwas falsch konfiguriert zu haben, können Sie vor diesem Zeitpunkt einfach den Rechner ausschalten.

2.4.1. Der Systemstart

2.4.1.1. Systemstart von i386™-Systemen

1. Schalten Sie zunächst Ihren Rechner aus.
2. Schalten Sie den Rechner ein. Während des Starts sollte angezeigt werden, wie Sie das Systemeinstellungsmenü (oder BIOS) erreichen. Meist drücken Sie dazu die Tasten **F2**, **F10**, **Del** oder **Alt**+**S**. Benutzen Sie die angezeigte Tastenkombination. Viele Rechner zeigen beim Systemstart eine Grafik an. Typischerweise können Sie die Grafik mit der Taste **Esc** entfernen und so die angezeigten Meldungen lesen.
3. Suchen Sie Option, die einstellt von welchem Gerät der Rechner startet. Normalerweise wird die Option `Boot Order` genannt und zeigt eine Geräteliste, beispielsweise `Floppy`, `CD-ROM`, `First Hard Disk` an.

   Wenn Sie von einer CD-ROM starten, stellen Sie sicher, das das CD-ROM-Laufwerk ausgewählt ist. Starten Sie hingegen von einem USB-Stick oder von einer Startdiskette, wählen Sie ebenfalls den entsprechenden Eintrag aus. Wenn Sie nicht sicher sind, lesen Sie bitte im Handbuch des Rechners oder im Handbuch der Systemplatine nach.

   Stellen Sie das gewünschte Startmedium ein und sichern Sie die Einstellungen. Der Rechner sollte dann neu starten.
4. Wenn Sie (wie in Abschnitt 2.3.7 beschrieben) ein bootbaren USB-Stick vorbereitet haben, stöpseln Sie diesen bitte ein, bevor Sie Ihren Rechner einschalten.

   Wenn Sie den Rechner von einer CD-ROM starten, legen Sie die CD-ROM so früh wie möglich in das Laufwerk ein.

   > **Anmerkung:** Bis einschließlich FreeBSD 7.3 kann FreeBSD auch von einer Startdiskette aus installiert werden. Eine Anleitung hierzu finden Sie in Abschnitt 2.3.7). Legen Sie die erste Diskette (diese enthält das Abbild `boot.flp`) in das Diskettenlaufwerk ein und starten Sie den Rechner.

   Wenn Ihr Rechner wieder normal startet und das existierende Betriebssystem lädt, kann das folgende Ursachen haben:

   1. Das Startmedium (Diskette, CD-ROM) ist nicht schnell genug eingelegt worden. Belassen Sie das Startmedium im Laufwerk und starten Sie Ihren Rechner neu.
   2. Die BIOS-Einstellungen sind falsch vorgenommen worden. Wiederholen Sie diesen Schritt, bis Sie die richtige Einstellung gefunden haben.
   3. Das verwendete BIOS kann nicht von dem gewünschten Medium starten.
5. FreeBSD startet jetzt. Wenn Sie von einer CD-ROM starten, sehen Sie die folgenden Meldungen (Versionsangaben entfernt):

   ```
   Booting from CD-Rom...
   645MB medium detected
   CD Loader 1.2

   Building the boot loader arguments
   Looking up /BOOT/LOADER... Found
   Relocating the loader and the BTX
   Starting the BTX loader
   ```

```
BTX loader 1.00 BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS 636kB/261056kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 syms=[0x4+0x6cac0+0x4+0x88e9d]
\
```

Wenn Sie mit Startdisketten hochfahren, sehen Sie folgende Meldungen (Versionsangaben entfernt):

```
Booting from Floppy...
Uncompressing ... done

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive C: is disk1
BIOS 639kB/261120kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/kernel text=0x277391 data=0x3268c+0x332a8 |

Insert disk labelled "Kernel floppy 1" and press any key...
```

Folgen Sie der Anweisung und entfernen Sie die `boot.flp`-Diskette, anschließend legen Sie die `kern1.flp`-Diskette ein und drücken **Enter**. Starten Sie das System mit der ersten Diskette und legen Sie, wenn Sie dazu aufgefordert werden, die anderen Disketten ein.

6. Unabhängig davon, ob Sie von Disketten oder von CD-ROM gestartet haben, erscheint danach das FreeBSD Bootloader-Menü:

Abbildung 2-1. FreeBSD Boot Loader Menu

Warten Sie entweder zehn Sekunden oder drücken Sie **Enter**.

2.4.1.2. Systemstart bei SPARC64®-Systemen

Die meisten SPARC64®-Systeme sind so konfiguriert, dass sie automatisch von der Festplatte starten. Um FreeBSD auf einem solchen System zu installieren, müssen Sie das System aber über das Netzwerk oder von einer CD-ROM starten. Daher müssen Sie den Bootprozess unterbrechen und das System über das PROM (OpenFirmware) starten.

Dazu starten Sie Ihr System neu und warten, bis die Startmeldung erscheint. Der genaue Wortlaut hängt vom eingesetzten Modell ab, die Nachricht sollte aber ähnlich der folgenden aussehen:

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial #51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

Um den Startvorgang zu unterbrechen, drücken Sie nun die Tastenkombination **L1**+**A** oder **Stop**+**A**. Verwenden Sie eine serielle Verbindung, senden Sie das Signal `BREAK` über die serielle Konsole (etwa durch die Eingabe von ~# in den Programmen tip(1) oder cu(1)). In beiden Fällen landen Sie anschließend am PROM-Prompt:

```
ok     ❶
ok {0} ❷
```

❶ Der auf Einprozessorsystemen verwendete Prompt.

❷ Der Prompt auf Mehrprozessorsystemen. Die Zahl steht dabei für die Anzahl der vorhandenen Prozessoren.

Nun legen Sie Ihre CD-ROM in das Laufwerk ein und geben am PROM-Prompt `boot cdrom` ein. Danach startet Ihr System von der eingelegten CD-ROM.

2.4.2. Die Geräteerkennung prüfen

Die letzten paar Hundert Zeilen der Bildschirmausgabe werden gesichert und können geprüft werden.

Um sich den Bildschirmpuffer anzusehen, drücken Sie die Taste **Scroll-Lock**. Im Puffer können Sie mit den Pfeiltasten oder den Tasten **PageUp** und **PageDown** blättern. Um zur normalen Bildschirmausgabe zurückzukehren, drücken Sie nochmals die Taste **Scroll-Lock**.

Prüfen Sie mit diesem Verfahren nun die Ausgaben der Geräteerkennung. Sie werden einen Text ähnlich wie in Abbildung 2-2 sehen. Die genauen Ausgaben sind abhängig von den in Ihrem System installierten Geräten.

Abbildung 2-2. Ausgabe der Geräteerkennung

```
avail memory = 253050880 (247120K bytes)
Preloaded elf kernel "kernel" at 0xc0817000.
Preloaded mfs_root "/mfsroot" at 0xc0817084.
md0: Preloaded image </mfsroot> 4423680 bytes at 0xc03ddcd4

md1: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1:<VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <iSA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0 <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci
0
usb0: <VIA 83572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr1
uhub0: 2 ports with 2 removable, self powered
pci0: <unknown card> (vendor=0x1106, dev=0x3040) at 7.3
dc0: <ADMtek AN985 10/100BaseTX> port 0xe800-0xe8ff mem 0xdb000000-0xeb0003ff ir
q 11 at device 8.0 on pci0
dc0: Ethernet address: 00:04:5a:74:6b:b5
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xec00-0xec1f irq 9 at device 10.
0 on pci0
ed0 address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
orm0: <Option ROM> at iomem 0xc0000-0xc7fff on isa0
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5” drive> on fdc0 drive 0
```

```
atkbdc0: <Keyboard controller (i8042)> at port 0x60,0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/@ mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
pppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
plip0: <PLIP network interface> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master UDMA33
acd0: CD-RW <LITE-ON LTR-1210B> at ata1-slave PIO4
Mounting root from ufs:/dev/md0c
/stand/sysinstall running as init on vty0
```

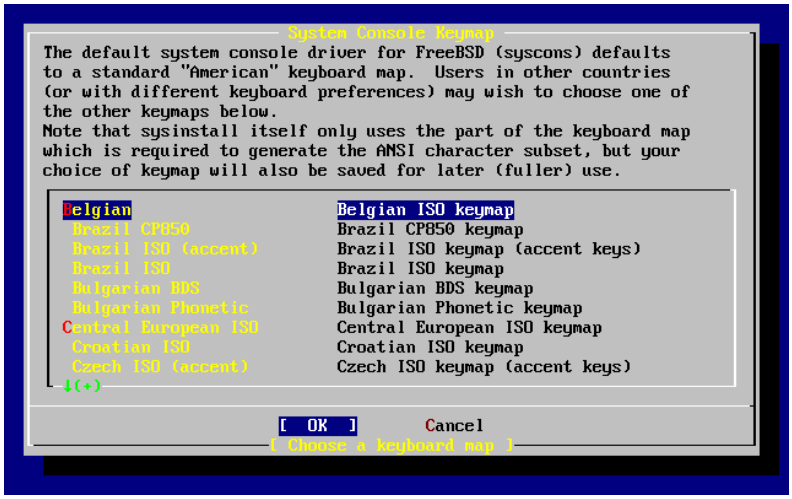
Prüfen Sie die Ausgabe der Geräteerkennung sorgfältig und stellen Sie sicher, dass FreeBSD alle erwarteten Geräte gefunden hat. Wenn ein Gerät nicht gefunden wurde, wird es nicht angezeigt. Ist dies bei Ihnen der Fall, müssen Sie einen angepassten Kernel erstellen, da das betroffene Gerät (beispielsweise eine Soundkarte) in diesem Fall vom `GENERIC`-Kernel nicht unterstützt wird.

Sie gelangen im nächsten Schritt in ein Menü, in dem Sie über die Cursortasten das Land, in dem Sie sich befinden, auswählen können (Abbildung 2-3). Durch die Bestätigung mit der **Enter**-Taste wird automatisch das von Ihnen gewählte Land sowie die dazu passende Tastaturbelegung gewählt.

Abbildung 2-3. Ihr Land auswählen

Haben Sie als Land United States gewählt, wird automatisch die amerikanische Standardtastatur verwendet. Haben Sie hingegen ein anderes Land angegeben, erscheint das folgende Menü, in dem Sie Ihre Tastaturbelegung auswählen können (bestätigen Sie Ihre Auswahl mit der **Enter**-Taste).

Abbildung 2-4. Die Tastaturbelegung auswählen



Nachdem Sie das Land ausgewählt haben, erscheint das Hauptmenü von **sysinstall**.

2.5. Das Werkzeug sysinstall

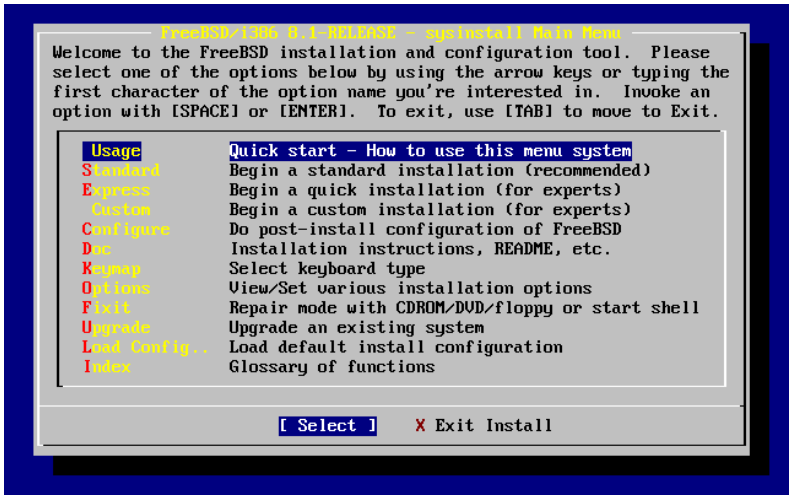
Zum Installieren von FreeBSD stellt das FreeBSD-Project das Werkzeug **sysinstall** zur Verfügung. Das Werkzeug arbeitet textorientiert und bietet eine Reihe von Menüs und Bildschirmen, um den Installationsprozess zu konfigurieren und zu steuern.

Die Menüs von **sysinstall** werden mit Tasten wie den Pfeiltasten, **Enter**, **Tab** oder **Space** bedient. Eine ausführliche Beschreibung der Tastenbelegung ist in der Gebrauchsanweisung von **sysinstall** enthalten.

Die Gebrauchsanweisung können Sie lesen, indem Sie den Menüpunkt Usage auswählen. Stellen Sie sicher, dass die Schaltfläche [Select], wie in Abbildung 2-5 gezeigt, aktiviert ist und drücken Sie die Taste **Enter**.

Es erscheinen Anweisungen wie das Menüsystem zu bedienen ist. Wenn Sie diese gelesen haben, drücken Sie **Enter**, um in das Hauptmenü zurückzukehren.

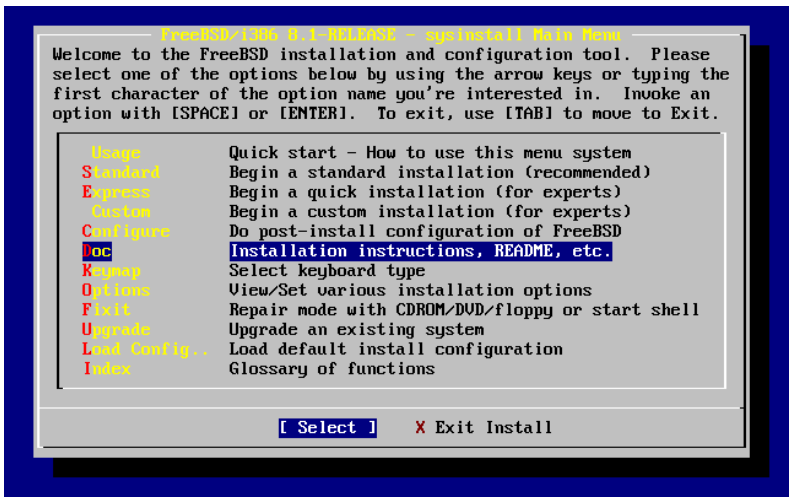
Abbildung 2-5. Die Gebrauchsanweisung von sysinstall auswählen



2.5.1. Die Dokumentation abrufen

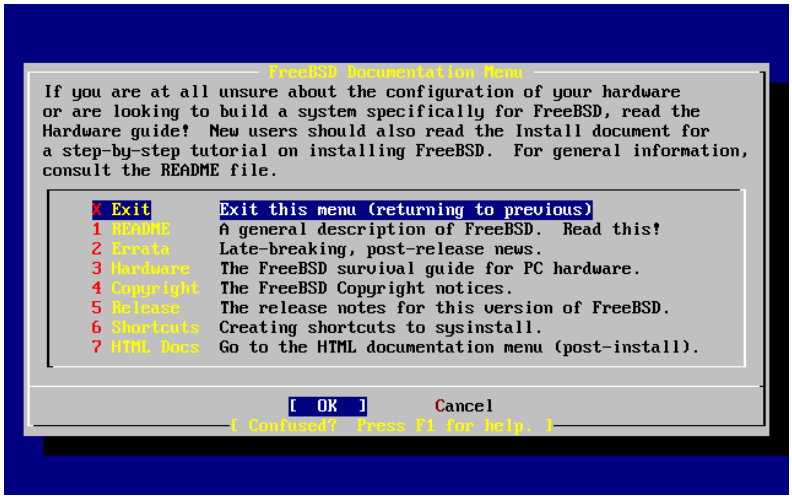
Aus dem Hauptmenü wählen Sie mit den Pfeiltasten Doc aus und drücken **Enter**.

Abbildung 2-6. Die Dokumentation abrufen



Es wird das Dokumentationsmenü angezeigt.

Abbildung 2-7. Das Dokumentationsmenü von sysinstall



Lesen Sie bitte unbedingt die mitgelieferte Dokumentation.

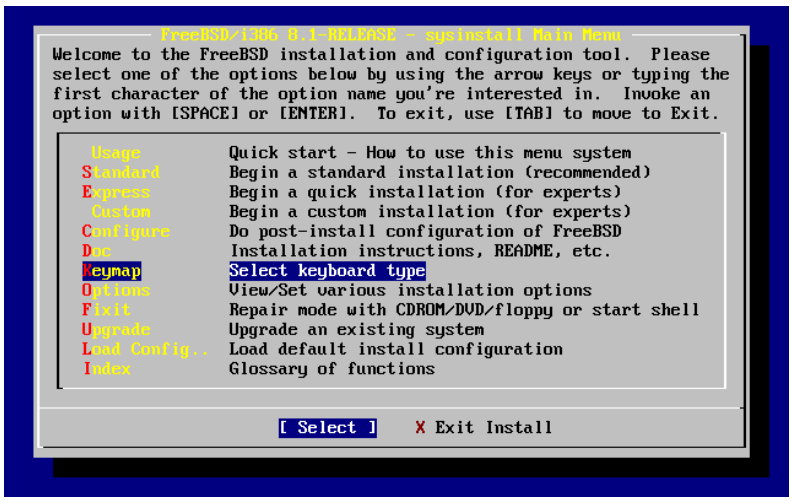
Um ein Dokument zu lesen, wählen Sie das Dokument mit den Pfeiltasten aus und drücken **Enter**. Wenn Sie das Dokument gelesen haben, kommen Sie mit der Taste **Enter** in das Dokumentationsmenü zurück.

Um in das Hauptmenü zurückzukommen, wählen Sie mit den Pfeiltasten Exit aus und drücken die Taste **Enter**.

2.5.2. Die Tastaturbelegung ändern

Um die Tastaturbelegung zu ändern, wählen Sie den Menüpunkt Keymap und drücken **Enter**. Dies ist nur erforderlich wenn Sie eine nicht standard-konforme Tastatur oder eine andere als eine amerikanische Tastatur einsetzen.

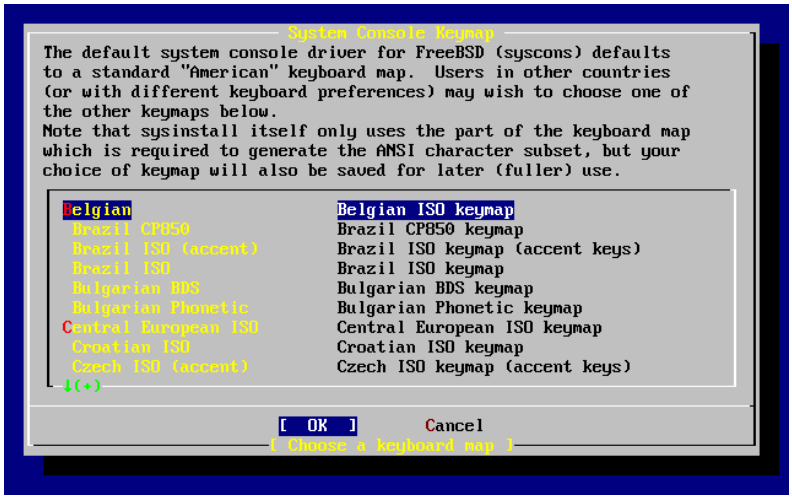
Abbildung 2-8. Das Hauptmenü von sysinstall



Eine andere Tastaturbelegung können Sie mit den Pfeiltasten markieren und der Taste **Space** auswählen. Wenn Sie die Taste **Space** nochmals drücken wird die Auswahl aufgehoben. Haben Sie eine Tastaturbelegung ausgewählt, markieren Sie mit den Pfeiltasten [OK] und drücken Sie **Enter**.

Der Bildschirmabzug zeigt nur einen der verfügbaren Belegungen an. Mit der Taste **Tab** markieren Sie die Schaltfläche [Cancel], die mit der Vorgabe-Belegung wieder in das Hauptmenü zurückführt.

Abbildung 2-9. Sysinstall Keymap Menu



2.5.3. Installationsoptionen einstellen

Wählen Sie Options aus und rücken die Taste **Enter**.

Abbildung 2-10. Das Hauptmenü von sysinstall

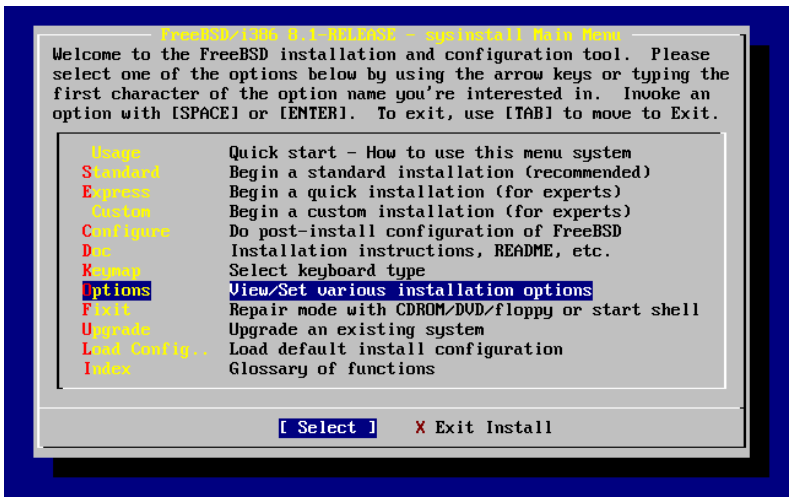
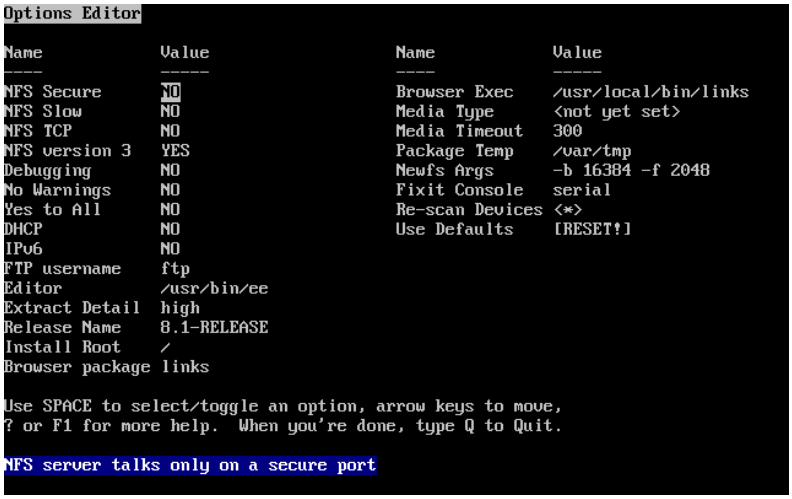


Abbildung 2-11. Optionen von sysinstall



Für die meisten Benutzer sind die voreingestellten Werte völlig ausreichend und brauchen daher nicht geändert werden. Der Name des Releases variiert mit der zu installierenden Version von FreeBSD.

Eine Beschreibung der ausgewählten Option erscheint blau hervorgehoben am unteren Ende des Bildschirms. Mit der Option Use Defaults können Sie alle Optionen auf die Vorgabewerte zurückstellen.

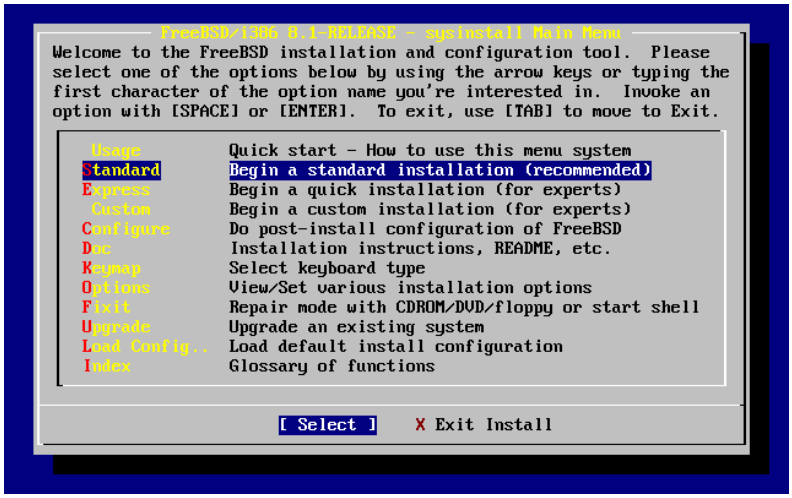
Wenn Sie die Hilfeseite zu den verschiedenen Optionen lesen wollen, drücken Sie die Taste **F1**.

Die Taste **Q** führt in das Hauptmenü zurück.

2.5.4. Eine Standard-Installation starten

Die Standard-Installation sollte von allen UNIX- oder FreeBSD-Anfängern benutzt werden. Markieren Sie mit den Pfeiltasten Standard und drücken Sie **Enter**, um die Installation zu starten.

Abbildung 2-12. Die Standard-Installation starten



2.6. Plattenplatz für FreeBSD bereitstellen

Ihre erste Aufgabe ist, FreeBSD Plattenplatz bereitzustellen und den Plattenplatz für **sysinstall** kenntlich zu machen (*label*). Sie müssen daher wissen, wie FreeBSD mit Platten umgeht.

2.6.1. Nummerierung der Laufwerke im BIOS

Bevor Sie FreeBSD installieren und konfigurieren, sollten Sie Sie einen wichtigen Punkt beachten. Dies gilt insbesondere dann, wenn Sie mehrere Festplatten besitzen.

In einem PC, der unter einem vom BIOS abhängigen Betriebssystem, wie MS-DOS oder Microsoft Windows läuft, kann das BIOS die normale Reihenfolge der Laufwerke verändern und das Betriebssystem beachtet diese Änderung. Mit dieser Funktion kann der Rechner von einem anderen Laufwerk als dem so genannten “primären Laufwerk” gestartet werden. Die Funktion ist sehr zweckmäßig für Benutzer, die Datensicherungen auf einer zweiten Platte erstellen und dafür Werkzeuge wie **Ghost®** oder **xcopy** einsetzen. Wenn die erste Platte ausfällt, von einem Virus befallen wird oder durch einen Fehler des Betriebssystems verunstaltet wird, können die Platten im BIOS logisch getauscht werden. Es sieht so aus, als wären die Laufwerke, ohne Öffnen des Gehäuses getauscht worden.

Teurere Systeme mit SCSI-Controllern haben oft BIOS-Erweiterungen, mit denen die Reihenfolge von bis zu sieben SCSI-Platten in ähnlicher Weise verändert werden kann.

Ein Benutzer, der es gewohnt ist, diese BIOS-Funktionen zu benutzen, mag überrascht sein, dass FreeBSD sich nicht wie erwartet verhält. FreeBSD verwendet das BIOS nicht und weiß daher nichts von der logischen Plattenordnung im BIOS. Dies kann zu sehr verwirrenden Situationen führen, insbesondere wenn die Platten identische Geometrien besitzen und Kopien voneinander sind.

Vor der Installation von FreeBSD sollte im BIOS die normale Nummerierung der Laufwerke eingestellt und so belassen werden. Ist es nötig, die Reihenfolge der Laufwerke zu verändern, so sollte das immer auf dem schweren Weg, also durch Öffnen des Gehäuses und Verändern der Jumper und Kabel, erfolgen.

RandnotizVon Bills und Freds ungewöhnlichen Abenteuern

Bill macht aus einer älteren Wintel Kiste ein neues FreeBSD-System für Fred. Auf einer SCSI-Platte, die er mit der SCSI-ID 0 konfiguriert, installiert Bill FreeBSD.

Nachdem Fred das System einige Tage benutzt hat, bemerkt er, dass die ältere SCSI-Platte viele Fehler meldet und beschwert sich bei Bill.

Nach einigen Tagen entschließt sich Bill, die Sache in die Hand zu nehmen. Er schnappt sich eine identische SCSI-Platte aus dem Lager im Hinterzimmer und baut diese, nachdem Sie einen Oberflächenscan überstanden hat, mit der SCSI-ID 4 ein. Anschließend kopiert er die Daten von der Platte mit der SCSI-ID 0 auf die Platte mit der SCSI-ID 4. Da die neue Platte zufriedenstellend läuft, stellt Bill im SCSI-BIOS die Reihenfolge der Platten so um, dass das System von der neuen Platte startet. Nach einem problemlosen Start von FreeBSD läuft das System und Fred ist zufrieden.

Nach einiger Zeit haben Bill und Fred Lust auf ein weiteres Abenteuer – Sie wollen das System auf eine neue FreeBSD-Version aktualisieren. Bill ersetzt die angeschlagene Platte mit der SCSI-ID 0 durch eine gleiche Platte aus dem Lager. Auf der ausgetauschten Platte installiert er problemlos mithilfe von Freds Startdisketten die neue Version von FreeBSD.

Fred braucht ein paar Tage, um die neue FreeBSD-Version zu testen und entscheidet, dass Sie für den produktiven Einsatz geeignet ist. Nun müssen die Daten von der alten Platte (mit der SCSI-ID 4) kopiert werden. Fred hängt dazu die alte Platte ein und stellt bestürzt fest, dass alle Daten verschwunden sind.

Wo sind die Daten hin?

Bill kopierte die Daten von der Platte mit der SCSI-ID 0 auf die Platte mit der SCSI-ID 4. Als Bill die Startreihenfolge im SCSI-BIOS änderte, führte er sich nur selbst an der Nase herum. FreeBSD lief weiterhin auf der Platte mit der SCSI-ID 0. Durch die Änderung der Startreihenfolge wurde nur ein Teil des Boot- und Loader-Codes von der Platte mit der SCSI-ID 4 geladen. Die Kernel-Treiber von FreeBSD ignorieren die BIOS-Einstellungen und benutzen die normale Nummerierung. Das System lief also weiterhin auf der Platte mit der SCSI-ID 0 und alle Daten von Fred befanden sich auf dieser Platte. Es schien nur so, als würde das System auf der Platte mit der SCSI-ID 4 laufen.

Wir sind erleichtert zu bemerken, dass keine Daten verloren gingen oder verändert wurden. Die alte Platte wurde im Müll wiedergefunden und Freds Daten konnten wiederhergestellt werden (Bill weiß jetzt, dass er noch viel zu lernen hat).

Obwohl in diesem Beispiel SCSI-Platten verwendet wurden, gelten die Konzepte gleichermaßen für IDE-Platten.

2.6.2. Slices mit Fdisk erstellen

> **Anmerkung:** Zu diesem Zeitpunkt werden noch keine Änderungen auf die Festplatte ausgeschrieben. Sie können daher **sysinstall** jederzeit verlassen, und erneut beginnen, wenn Sie denken, einen Fehler gemacht zu haben. Sie können **sysinstall** über die Menüs verlassen, die Taste **U** drücken oder die Option Undo wählen. Wenn Sie einmal nicht wissen, wie Sie ein Menü verlassen, können Sie den Rechner auch einfach ausschalten.

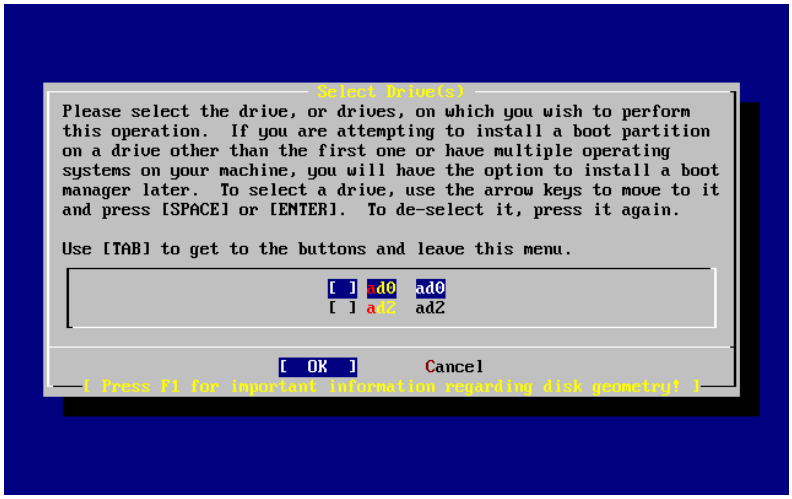
Nachdem Sie in **sysinstall** die Standard-Installation ausgewählt haben, werden Sie folgende Meldung sehen:

```
Message
```

```
In the next menu, you will need to set up a DOS-style ("fdisk")
partitioning scheme for your hard disk. If you simply wish to devote
all disk space to FreeBSD (overwriting anything else that might be on
the disk(s) selected) then use the (A)ll command to select the default
partitioning scheme followed by a (Q)uit. If you wish to allocate only
free space to FreeBSD, move to a partition marked "unused" and use the
(C)reate command.
                                [  OK  ]

                        [ Press enter or space ]
```

Drücken Sie, wie angegeben, **Enter**. Im nächsten Bildschirm werden alle Festplatten angezeigt, die der Kernel während der Geräteerkennung gefunden hat. Abbildung 2-13 zeigt ein Beispiel von einem System mit zwei IDE-Platten, die als `ad0` und `ad2` erkannt wurden.

Abbildung 2-13. Ein Laufwerk für Fdisk aussuchen



Sie fragen sich vielleicht, warum `ad1` nicht angezeigt wird. Wurde die Platte vielleicht nicht erkannt?

Stellen Sie sich ein System mit zwei IDE-Platten vor. Eine Platte ist als Master am ersten Controller, die andere als Master am zweiten Controller angeschlossen. Wenn FreeBSD die Platten in der Reihenfolge, in der sie gefunden werden, nummerieren würde, hießen die Platten `ad0` und `ad1` und alles würde funktionieren.

Wenn Sie nun am ersten IDE-Controller eine dritte Platte als Slave anschließen würden, wäre diese Platte `ad1`. Die vorher `ad1` genannte Platte würde nun `ad2` heißen. Dateisysteme werden auf Geräten wie `ad1s1a` angelegt. Daher könnte es passieren, dass auf einmal Dateisysteme nicht mehr gefunden werden und Sie FreeBSD umkonfigurieren müssten.

Um diese Probleme zu umgehen, kann der Kernel so eingestellt werden, dass er Platten nach ihrem Anschlussort anstelle der gefundenen Reihenfolge benennt. Nach diesem Schema ist die Master-Platte am zweiten IDE-Controller *immer* `ad2`, auch wenn es die Geräte `ad0` oder `ad1` gar nicht gibt.

Dieses Verhalten ist in FreeBSD voreingestellt und der Grund warum im Beispiel die Geräte `ad0` und `ad2` angezeigt werden. Der Rechner, von dem die gezeigte Ausgabe stammt, hatte zwei IDE-Platten, die beide als Master konfiguriert waren, und keine Slave-Platten.

Wählen Sie die Platte aus, auf die Sie FreeBSD installieren wollen und drücken Sie [OK]. Anschließend startet **Fdisk** und zeigt einen Bildschirm wie den in Abbildung 2-14.

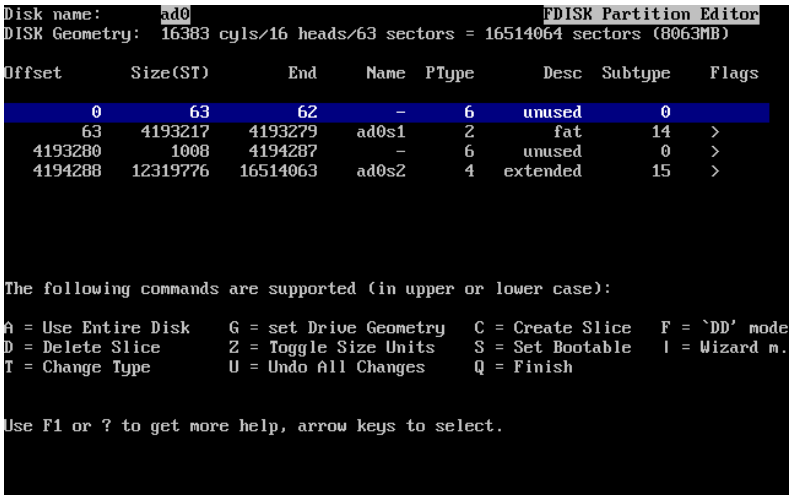
Der Bildschirm von **Fdisk** ist in drei Abschnitte unterteilt.

Der erste Abschnitt umfasst die ersten beiden Zeilen der Anzeige. Er enthält Einzelheiten über die aktuell ausgewählte Platte, unter anderem den FreeBSD-Gerätenamen, die Plattengeometrie und die Kapazität der Platte.

Der zweite Abschnitt zeigt die auf der Platte befindlichen Slices. Angezeigt wird der Anfang und das Ende der Slice, die Größe der Slice, der FreeBSD-Gerätename, eine Beschreibung und der Subtyp. Im Beispiel sehen Sie zwei unbenutzte Slices, die durch die Plattenbelegung auf PCs entstehen. Weiterhin sehen Sie eine große FAT-Slice, die ziemlich sicher unter MS-DOS/Windows als Laufwerk `C:` auftaucht und eine erweiterte Slice, die unter MS-DOS/Windows weitere Laufwerke enthalten kann.

Im dritten Abschnitt sind die Kommandos von **Fdisk** zusammengefasst.

Abbildung 2-14. Typischer Fdisk-Bildschirm vor dem Editieren



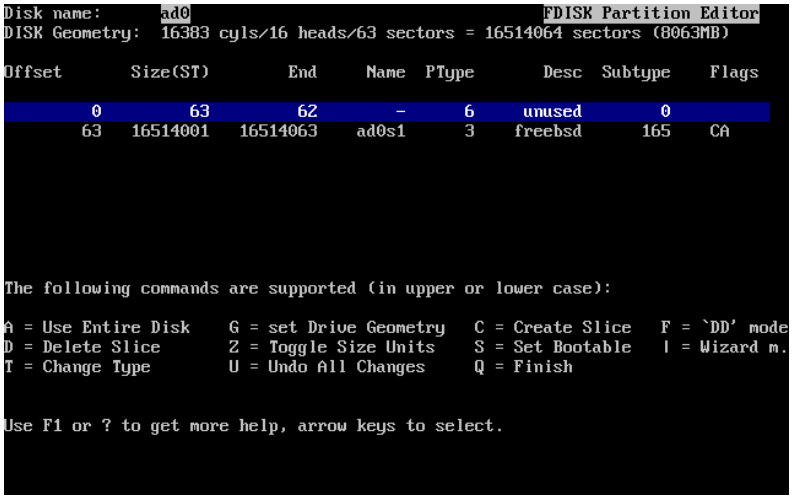
Die nächsten Schritte hängen von der beabsichtigten Einteilung der Festplatte ab.

Wenn Sie die gesamte Festplatte für FreeBSD verwenden wollen, drücken Sie die Taste **A** (entspricht dem Menüpunkt Use Entire Disk). Später im Installationsverlauf müssen Sie diese Auswahl bestätigen, danach werden alle bisherigen Daten von der Festplatte gelöscht. Diese Auswahl löscht vorher vorhandene Slices und ersetzt sie durch einen kleinen unbenutzten Bereich (der wieder durch das PC-Design bedingt ist) und eine große Slice für FreeBSD. Wählen Sie dann die neu erstellte Slice mit den Pfeiltasten aus und drücken Sie die Taste **S**, um die Slice als startfähig (bootbar) zu markieren. Abbildung 2-15 zeigt den Bildschirm zu diesem Zeitpunkt. Beachten Sie das `A` in der Spalte `Flags`. Dies zeigt an, dass die Slice *aktiv* ist und das System von dieser Slice starten wird.

Um Platz für FreeBSD zu schaffen, können Sie auch bestehende Slices löschen. Markieren Sie dazu die Slice mit den Pfeiltasten und drücken Sie die Taste **D**. Danach legen Sie eine neue Slice mit der Taste **C** an. Sie werden nach der Größe der zu erstellenden Slice gefragt; der Vorgabewert entspricht der größten Slice, die angelegt werden kann (entspricht entweder dem größten freien Bereich auf der Festplatte oder der ganzen Festplatte).

Wenn Sie schon Platz für FreeBSD geschaffen haben (beispielsweise mit **PartitionMagic**), können Sie eine neue Slice direkt mit der Taste **C** anlegen. Sie werden wieder nach der Größe der anzulegenden Slice gefragt.

Abbildung 2-15. Eine Partition über die gesamte Platte



Drücken Sie die Taste **Q**, wenn Sie fertig sind. **Sysinstall** merkt sich die Änderungen, schreibt sie aber noch nicht auf die Festplatte.

2.6.3. Einen Boot-Manager installieren

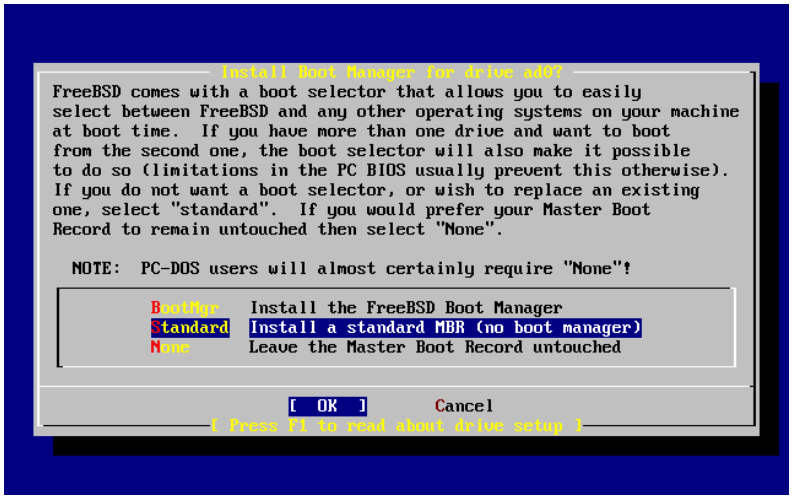
Sie können nun einen Boot-Manager installieren. Unter folgenden Umständen sollten Sie den FreeBSD-Boot-Manager installieren:

- Das System besitzt mehr als ein Laufwerk und FreeBSD ist auf einem anderen Laufwerk als dem ersten Laufwerk installiert.
- FreeBSD teilt sich das Laufwerk mit einem anderen Betriebssystem. Beim Systemstart wollen Sie auswählen, welches Betriebssystem gestartet wird.

Wird der Rechner ausschließlich mit FreeBSD betrieben und FreeBSD ist auf dem ersten Laufwerk installiert, dann genügt der Standard-Boot-Manager. Wenn Sie einen anderen Boot-Manager benutzen, der FreeBSD starten kann, wählen Sie bitte None aus.

Nachdem Sie die Auswahl getroffen haben, drücken Sie die Taste **Enter**.

Abbildung 2-16. Sysinstall Boot-Manager-Menü



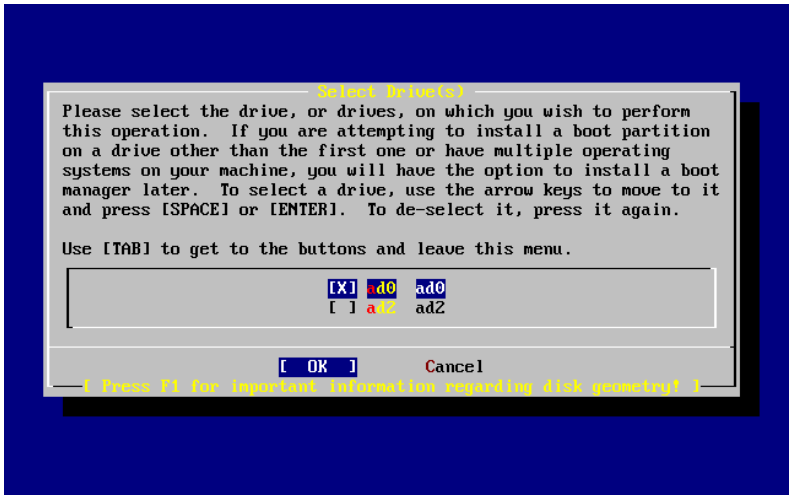
In der Hilfe, die Sie mit der Taste **F1** aufrufen, werden Probleme beschrieben, die entstehen können, wenn sich zwei Betriebssysteme ein Laufwerk teilen.

2.6.4. Slices auf einem anderen Laufwerk anlegen

Wenn das System mehr als ein Laufwerk besitzt, kehrt die Installationsprozedur nach der Auswahl des Boot-Managers zum Bildschirm `Select Drives` zurück. Sie können hier ein anderes Laufwerk auswählen und auf diesem Laufwerk mit **Fdisk** weitere Slices anlegen.

> **Wichtig:** Wenn Sie FreeBSD auf einem anderen Laufwerk als dem ersten Laufwerk installieren, müssen Sie den FreeBSD-Boot-Manager auf beiden Laufwerken installieren.

Abbildung 2-17. Die Laufwerksauswahl verlassen



Die Taste **Tab** wechselt zwischen dem zuletzt ausgewählten Laufwerk und den Schaltflächen [OK] und [Cancel].

Drücken Sie einmal die Taste **Tab**, um [OK] auszuwählen und drücken Sie anschließend **Enter** um die Installation weiterzuführen.

2.6.5. Partitionen mit Bsdlabel anlegen

In jeder angelegten Slice müssen Sie Partitionen anlegen. Die Partitionen werden mit Buchstaben von a bis h gekennzeichnet. Die Buchstaben b, c und d haben eine besondere Bedeutung, die Sie beachten sollten.

Einige Anwendungen profitieren von einer besonderen Aufteilung der Partitionen, insbesondere wenn das System mehr als ein Laufwerk besitzt. Bei der ersten FreeBSD-Installation sollten Sie sich allerdings nicht zu viele Gedanken über die Partitionen machen. Wichtiger ist, dass Sie FreeBSD installieren und benutzen. Wenn Sie mehr Erfahrung mit FreeBSD gesammelt haben, können Sie FreeBSD jederzeit mit anderen Partitionen installieren.

Das folgende Schema legt vier Partitionen an: Eine Partition für den Auslagerungsbereich (*swap space*) und drei Partitionen für Dateisysteme.

Tabelle 2-2. Partitionen auf dem ersten Laufwerk

Partition	Dateisystem	Größe	Beschreibung
a	/	1 GB	Das Root-Dateisystem. Jedes andere Dateisystem wird irgendwo unterhalb von diesem Dateisystem eingehangen. 1 GB ist eine vernünftige Größe für dieses Dateisystem. Sie werden hier wenig Daten speichern und FreeBSD benötigt ungefähr 128 MB Platz auf diesem Dateisystem. Der Rest ist für temporäre Daten und die Reserve, falls künftige Versionen von FreeBSD mehr Platz in / benötigen.

Partition	Dateisystem	Größe	Beschreibung
`b`	N/A	2-3 x RAM	Der Auslagerungsbereich befindet sich auf der `b`-Partition. Es ist schon fast eine Kunst, die Größe des Auslagerungsbereichs richtig zu bestimmen. Eine gute Daumenregel ist, den Auslagerungsbereich zwei bis dreimal größer als den Hauptspeicher (RAM) anzulegen. Sie sollten mindestens 64 MB für den Auslagerungsbereich vorsehen. Wenn das System also weniger als 32 MB Hauptspeicher besitzt, richten Sie einen 64 MB großen Auslagerungsbereich ein. Besitzt das System mehr als ein Laufwerk, können Sie auf jedem Laufwerk Auslagerungsbereiche anlegen. Da FreeBSD alle Auslagerungsbereiche benutzt, wird der Vorgang des Auslagerns durch mehrere Bereiche beschleunigt. Berechnen Sie in diesem Fall die Größe des benötigten Auslagerungsbereichs, beispielsweise 128 MB, und teilen Sie die Größe durch die Anzahl der Laufwerke. Dies gibt die Größe des Auslagerungsbereichs auf jedem Laufwerk. Mit zwei Platten ergibt das in diesem Beispiel 64 MB Auslagerungsbereich pro Platte.
`e`	`/var`	512 MB bis 4096 MB	Das Verzeichnis `/var` enthält Dateien, die sich dauernd ändern (Protokolldateien und Dateien für Verwaltungszwecke) und auf die im Normalbetrieb oft zugegriffen wird. Liegen diese Dateien in einem gesonderten Dateisystem, kann FreeBSD den Zugriff auf die Dateien optimieren, ohne den Zugriff auf Dateien mit einem anderen Zugriffmuster zu stören.
`f`	`/usr`	Der Rest des Laufwerks (mindestens 8 GB)	Alle anderen Dateien werden normalerweise im Verzeichnis `/usr` oder einem Unterverzeichnis von `/usr` abgelegt.

Warnung: Die eben genannten Werte dienen nur als Beispiel und sollten nur von erfahrenen Benutzern editiert werden. Wir empfehlen Ihnen, die vom Partitionseditor vorgeschlagene Aufteilung (`Auto Defaults`) zu verwenden.

Wenn Sie FreeBSD auf mehr als einem Laufwerk installieren, müssen Sie noch weitere Partitionen in den Slices auf den anderen Laufwerken anlegen. Am einfachsten legen Sie pro Laufwerk zwei Partitionen an: eine für den Auslagerungsbereich und eine andere für ein Dateisystem.

Tabelle 2-3. Partitionen auf weiteren Laufwerken

Partition	Dateisystem	Größe	Beschreibung

Partition	Dateisystem	Größe	Beschreibung
b	-	-	Wie schon besprochen, können Sie den Auslagerungsbereich auf mehrere Platten verteilen. Auch wenn die a-Partition frei ist, sollte der Auslagerungsbereich entsprechend der Konvention auf der b-Partition angelegt werden.
e	/disk*n*	Der Rest des Laufwerks	Der Rest der Platte wird von einer großen Partition eingenommen. Sie könnten für diese Partition die a-Partition anstelle der e-Partition benutzen. Allerdings ist die a-Partition per Konvention für das Root-Dateisystem (/) reserviert. Sie brauchen die Konvention nicht zu beachten, da aber **sysinstall** die Konvention beachtet, ist die Installation sauberer, wenn Sie das auch tun. Sie können das Dateisystem irgendwo einhängen. Das Beispiel schlägt die Verzeichnisse /disk*n* vor, wobei *n* die Laufwerke nummeriert. Sie können ein anderes Schema verwenden, wenn Sie möchten.

Wenn Sie die Aufteilung der Partitionen festgelegt haben, können Sie die Partitionen mit **sysinstall** anlegen. Es erscheint die nachstehende Meldung:

```
                              Message
Now, you need to create BSD partitions inside of the fdisk
partition(s) just created. If you have a reasonable amount of disk
space (1GB or more) and don't have any special requirements, simply
use the (A)uto command to allocate space automatically. If you have
more specific needs or just don't care for the layout chosen by
(A)uto, press F1 for more information on manual layout.

                                  [  OK  ]
                            [ Press enter or space ]
```

Drücken Sie **Enter**, um den FreeBSD-Partitionseditor, der **Disklabel** heißt, zu starten.

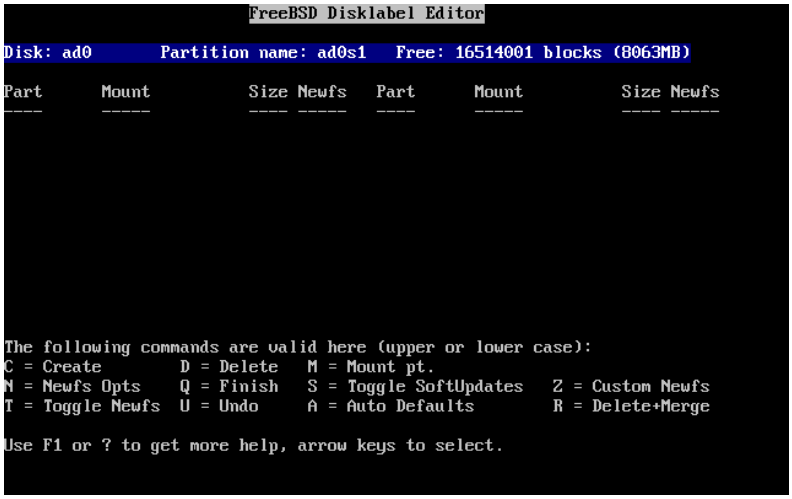
Abbildung 2-18 zeigt den Einstiegsbildschirm von **Disklabel**. Der Bildschirm ist in drei Bereiche geteilt.

Die ersten Zeilen zeigen den Namen des Laufwerks, das Sie gerade bearbeiten und die Slice, die die erstellten Partitionen enthält (**Disklabel** spricht hier von Partitionen anstatt von Slices). Der freie Platz einer Slice, der noch keiner Partition zugeordnet ist, wird ebenfalls angezeigt.

In der Mitte des Bildschirms werden die angelegten Partitionen, der Name des Dateisystems, das sich in der Partition befindet, dessen Größe und die Optionen zum Erstellen des Dateisystems angezeigt.

Das untere Drittel des Bildschirms zeigt die in **Disklabel** gültigen Tastenkombinationen.

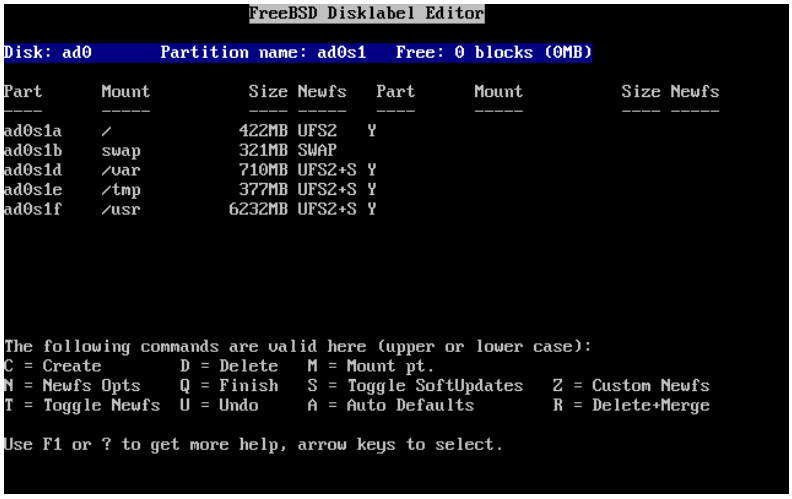
Abbildung 2-18. Sysinstall Disklabel-Editor



Disklabel kann für Sie automatisch Partitionen mit vorgegebenen Größen erstellen (diese Standardgrößen werden durch einen internen Partitionierungsalgorithmus ermittelt, der auf der Plattengröße beruht). Probieren Sie das bitte jetzt aus und drücken Sie die Taste **A**. Der Bildschirm sieht danach ähnlich wie in Abbildung 2-19 aus. Abhängig von der Größe des Laufwerks können die Vorgabewerte richtig oder falsch sein. Da Sie die Vorgaben nicht akzeptieren müssen, spielt das keine Rolle.

> **Anmerkung:** FreeBSD legt das Verzeichnis `/tmp` in einer eigenen Partition an. Dies verhindert, dass sich die Root-Partition mit temporären Dateien füllt.

Abbildung 2-19. Sysinstall Disklabel-Editor mit automatischen Vorgaben

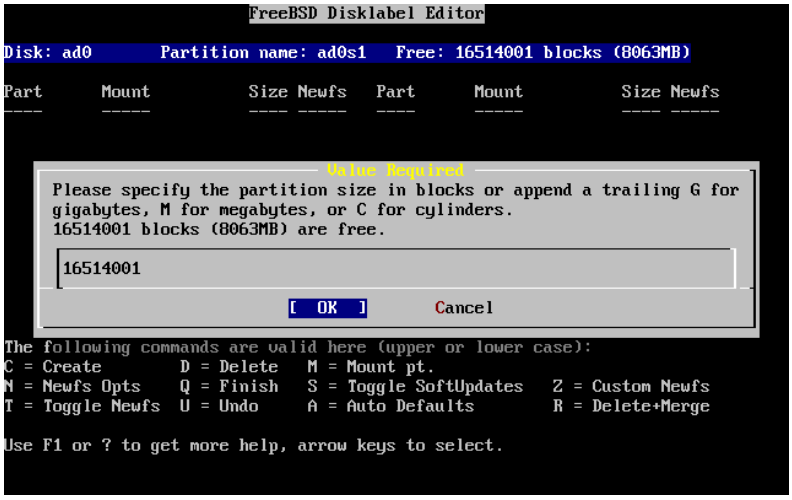


Wollen Sie die vorgegebenen Partitionen nicht verwenden und durch eigene ersetzen, markieren Sie mit den Pfeiltasten die erste Partition und drücken Sie die Taste **D**, um die Partition zu löschen. Wiederholen Sie dies für alle

vorgegebenen Partitionen.

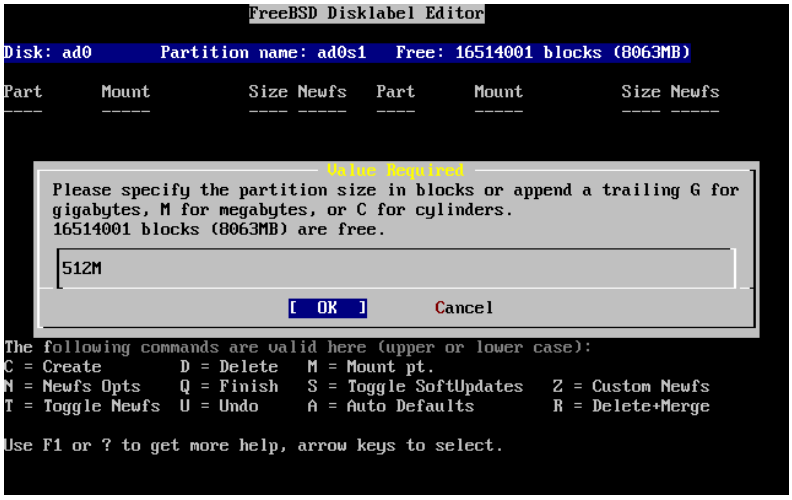
Um die erste Partition (a), die als `/` eingehangen wird, zu erstellen, drücken Sie die Taste **C**. Stellen Sie dabei sicher, dass die richtige Slice im oberen Teil des Bildschirms markiert ist. Wie in Abbildung 2-20, erscheint ein Fenster, in dem Sie die Größe der Partition angeben müssen. Sie können die Größe in Blöcken oder einer Zahl gefolgt von `M` für Megabyte, `G` für Gigabyte oder `C` für Zylinder angeben.

Abbildung 2-20. Die Größe einer Partition festlegen



Die vorgegebene Größe erstellt eine Partition, die den Rest der Slice ausfüllt. Wenn Sie die Größen aus dem früheren Beispiel verwenden, löschen Sie die vorgeschlagene Größe mit der Taste **Backspace** und tragen Sie **`512M`** ein, wie in Abbildung 2-21 gezeigt. Drücken Sie anschließend [OK].

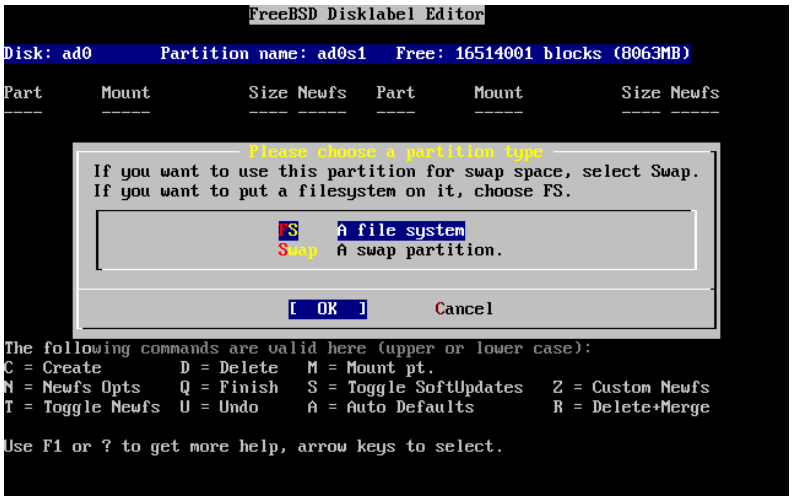
Abbildung 2-21. Die Größe einer Partition ändern



Nachdem Sie die Größe der Partition festgelegt haben, werden Sie gefragt, ob die Partition ein Dateisystem oder einen Auslagerungsbereich enthalten soll (siehe Abbildung 2-22). Die erste Partition enthält ein Dateisystem, wählen

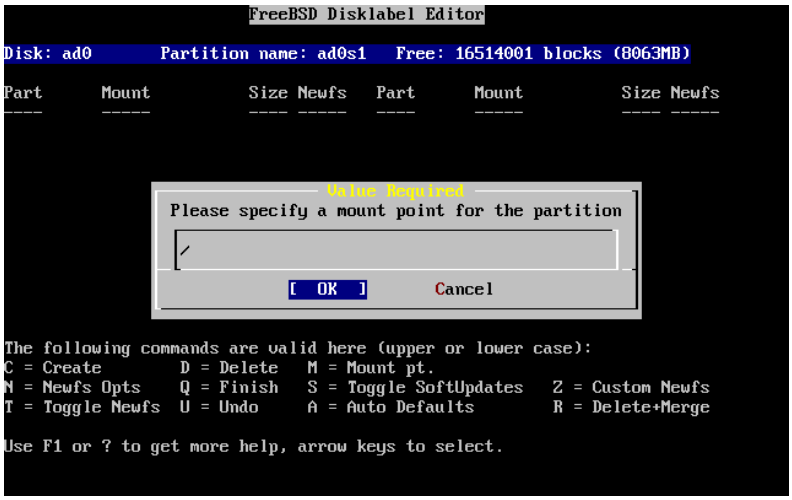
Sie FS aus und drücken Sie die Taste **Enter**.

Abbildung 2-22. Den Partitionstyp festlegen



Abschließend müssen Sie, weil Sie ein Dateisystem erstellen, angeben, wo das Dateisystem eingehangen wird. Die Eingabe ist in Abbildung 2-23 dargestellt. Das Root-Dateisystem wird in `/` eingehangen, geben Sie daher `/` ein und drücken Sie die Taste **Enter**.

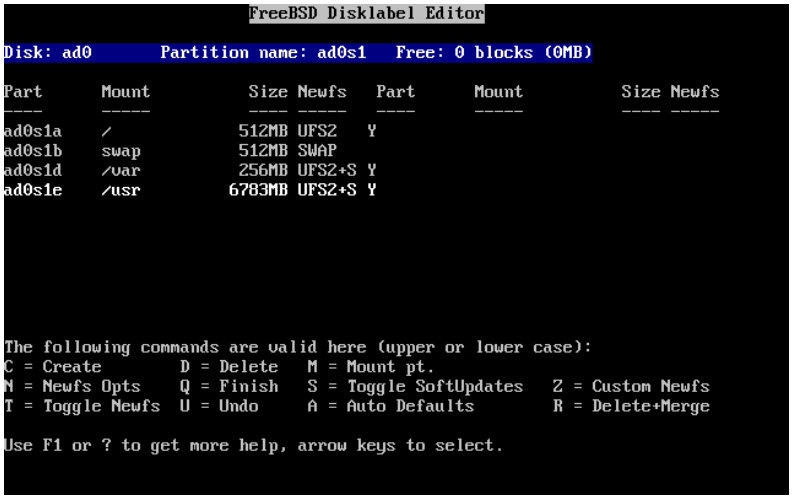
Abbildung 2-23. Den Mountpoint festlegen



Auf dem Bildschirm wird jetzt die neu angelegte Partition angezeigt. Wiederholen Sie diese Prozedur für die restlichen Partitionen. Beim Anlegen des Auslagerungsbereichs werden Sie nicht nach einem Mountpoint gefragt, da ein Auslagerungsbereich nie eingehangen wird. Wenn Sie die letzte Partition anlegen, `/usr`, können Sie die vorgeschlagene Größe stehen lassen. Das Dateisystem wird dann den Rest der Slice einnehmen.

Der letzte Bildschirm von **Disklabel** sieht wie in Abbildung 2-24 aus (Ihre Werte werden von den gezeigten Werten abweichen). Drücken Sie die Taste **Q**, um **Disklabel** zu verlassen.

Abbildung 2-24. Sysinstall Disklabel-Editor



2.7. Den Installationsumfang bestimmen

2.7.1. Die Distribution auswählen

Welche Software Sie installieren, hängt hauptsächlich vom Zweck des Rechners und dem zur Verfügung stehenden Plattenplatz ab. Die vorgegebenen Distributionen reichen von der minimalen Installation bis hin zu einer kompletten Installation. Anfänger sollten eine der vorgegebenen Distributionen auswählen, erfahrene Benutzer können die zu installierende Distribution anpassen.

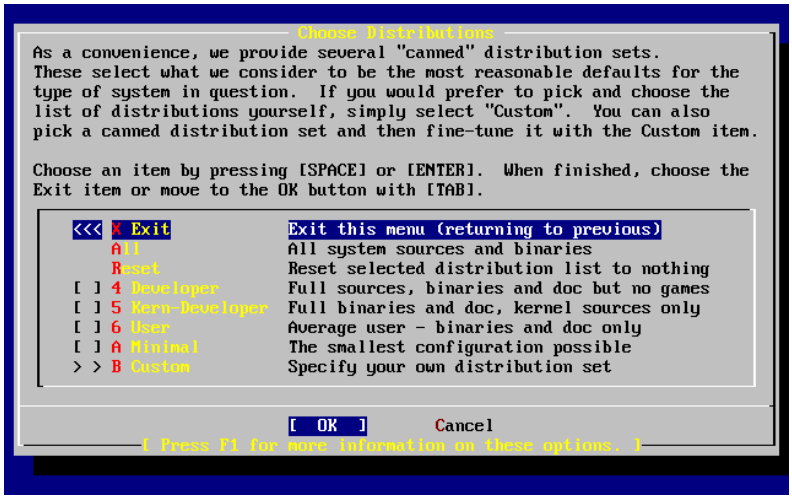
Die Taste **F1** führt zu einem Hilfebildschirm, der die Distributionen und deren Inhalte beschreibt. Drücken Sie **Enter**, um die Hilfe zu verlassen und zur Auswahl der Distribution zurückzukehren.

Wenn Sie eine graphische Benutzeroberfläche installieren wollen, müssen Sie die Konfiguration des X-Servers und die Auswahl der Benutzeroberfläche nach erfolgreicher Installation durchführen. Die Installation und Konfiguration des X-Servers wird in Kapitel 6 besprochen.

Wenn Sie einen angepassten Kernel erstellen wollen, wählen Sie eine Distribution aus, die den Quellcode (*source code*) enthält. Warum und wie Sie einen angepassten Kernel erstellen, erfahren Sie in Kapitel 9.

Natürlich ist das flexibelste System das, auf dem alles installiert ist. Wenn das System über ausreichend Plattenplatz verfügt, wählen Sie mit den Pfeiltasten die Option All aus (siehe Abbildung 2-25) und drücken die Taste **Enter**. Wenn Sie Bedenken haben, dass der Plattenplatz nicht ausreicht, wählen Sie eine Distribution, die weniger Software enthält. Machen Sie sich keine unnötigen Sorgen um die richtige Distribution, ausgelassene Distribution können später nachinstalliert werden.

Abbildung 2-25. Die Distribution auswählen



2.7.2. Die Ports-Sammlung installieren

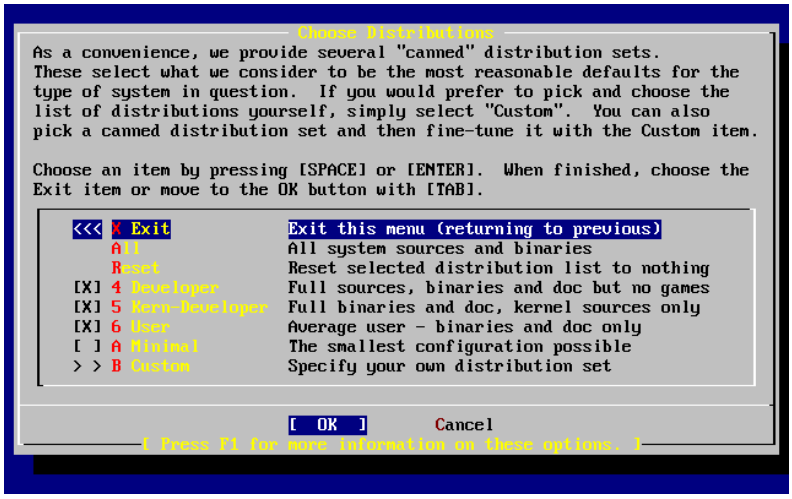
Nach der Auswahl der Distribution haben Sie Gelegenheit, die FreeBSD-Ports-Sammlung zu installieren. Mit der Ports-Sammlung lässt sich Software Dritter auf einfache Art und Weise installieren. Der Quellcode der zu installierenden Software ist nicht in der Ports-Sammlung enthalten. Stattdessen enthält die Ports-Sammlung Dateien, die den Installationsprozess (herunterladen, übersetzen und installieren) automatisieren. Die Ports-Sammlung wird in Kapitel 5 besprochen.

Der Installationsprozess prüft nicht, ob ausreichend Platz für die Ports-Sammlung vorhanden ist. Wählen Sie die Ports-Sammlung bitte nur aus, wenn das System über ausreichenden Platz verfügt. In FreeBSD 9.1 nimmt die Ports-Sammlung ungefähr 500 MB Plattenplatz in Anspruch. Neuere Versionen von FreeBSD benötigen mit Sicherheit noch mehr Platz.

```
                        User Confirmation Requested
Would you like to install the FreeBSD ports collection?

This will give you ready access to over 24,000 ported software packages,
at a cost of around 500 MB of disk space when "clean" and possibly much
more than that if a lot of the distribution tarballs are loaded
(unless you have the extra CDs from a FreeBSD CD/DVD distribution
available and can mount it on /cdrom, in which case this is far less
of a problem).

The ports collection is a very valuable resource and well worth having
on your /usr partition, so it is advisable to say Yes to this option.

For more information on the ports collection & the latest ports,
visit:
    http://www.FreeBSD.org/ports

                              [ Yes ]     No
```

Wählen Sie mit den Pfeiltasten [Yes] aus, um die Ports-Sammlung zu installieren. Wählen Sie [No] aus, um die Ports-Sammlung auszulassen. Drücken Sie danach die Taste **Enter**, es erscheint wieder das Distributionsmenü.

Abbildung 2-26. Die Distributionen bestätigen



Wenn Sie mit den ausgewählten Optionen zufrieden sind, wählen Sie mit den Pfeiltasten Exit aus (stellen Sie sicher, dass [OK] aktiv ist) und drücken Sie die Taste **Enter**.

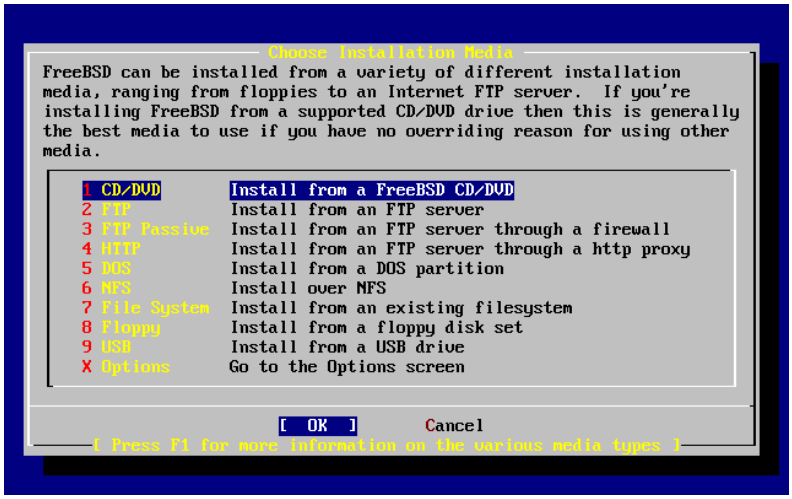
2.8. Das Installationsmedium auswählen

Wenn Sie von einer CD-ROM oder einer DVD installieren, wählen Sie bitte Install from a FreeBSD CD/DVD aus. Stellen Sie sicher, dass [OK] aktiv ist und drücken Sie dann die Taste **Enter**, um mit der Installation fortzufahren.

Wenn Sie ein anderes Installationsmedium benutzen, wählen Sie die passende Option aus und folgen den angezeigten Anweisungen.

Die Hilfeseiten über Installationsmedien erreichen Sie mit der Taste **F1**. Drücken Sie **Enter**, um zur Auswahl des Installationsmediums zurückzukehren.

Abbildung 2-27. Das Installationsmedium auswählen



FTP-Installationsmodi: Sie können zwischen drei FTP-Installationsmodi wählen: Active-FTP, Passive-FTP oder über einen HTTP-Proxy.

FTP Active: Install from an FTP server

Diese Option führt alle FTP-Operationen im Active-Mode aus. Dieser Modus funktioniert nicht durch Firewalls, er funktioniert aber mit alten FTP-Servern, die den Passive-Mode nicht beherrschen. Wenn die Verbindung im Passive-Mode (das ist die Vorgabe) hängt, versuchen Sie den Active-Mode.

FTP Passive: Install from an FTP server through a firewall

Mit dieser Option benutzt **sysinstall** den Passive-Mode für alle FTP-Operationen. In diesem Modus funktionieren Verbindungen durch Firewalls, die einkommende Pakete auf beliebigen TCP-Ports blockieren.

FTP via a HTTP proxy: Install from an FTP server through a http proxy

Diese Option weist **sysinstall** an, alle FTP-Operationen mit HTTP über einen Proxy (wie ein Web-Browser) durchzuführen. Der Proxy leitet die Anfragen an den richtigen FTP-Server weiter. Mit dieser Option passieren Sie eine Firewall, die FTP-Verbindungen verbietet, aber einen HTTP-Proxy anbietet. Neben dem FTP-Server müssen Sie in diesem Fall den Proxy-Server angeben.

Bei einem FTP-Proxy-Server müssen Sie normalerweise den Ziel-FTP-Server als Teil des Benutzernamens hinter dem Klammeraffen (“@”) angeben. Der Proxy-Server übernimmt die Kommunikation mit dem Ziel-FTP-Server. Nehmen wir an, Sie wollen von `ftp.FreeBSD.org` über den FTP-Proxy `foo.example.com` auf Port `1234` installieren.

Wählen Sie das Menü Options aus und setzen Sie dort den FTP-Benutzernamen (*username*) auf `ftp@ftp.FreeBSD.org`. Als Passwort geben Sie bitte Ihre E-Mail-Adresse an. Setzen Sie das Installationsmedium auf Active-FTP oder Passive-FTP, je nachdem welchen Modus der Proxy-Server unterstützt. Für die URL geben Sie `ftp://foo.example.com:1234/pub/FreeBSD` an.

Der Proxy-Server `foo.example.com` leitet Zugriffe auf das Verzeichnis `/pub/FreeBSD` an den Server `ftp.FreeBSD.org` weiter. Daher können `foo.example.com` als FTP-Server angeben.

2.9. Die Installation festschreiben

Wenn Sie wünschen, kann die Installation nun beginnen. Dies ist die letzte Gelegenheit, die Installation abzubrechen und Änderungen auf der Festplatte zu vermeiden.

```
                      User Confirmation Requested
 Last Chance! Are you SURE you want to continue the installation?

 If you're running this on a disk with data you wish to save then WE
 STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

 We can take no responsibility for lost disk contents!

                             [ Yes ]    No
```

Wählen Sie [Yes] aus und drücken Sie **Enter**, um weiter zu machen.

Die Installationsdauer hängt von den ausgewählten Distributionen, dem Installationsmedium und der Geschwindigkeit des Rechners ab. Während der Installation wird der Fortgang mit Statusmeldungen angezeigt.

Die Installation ist beendet, wenn die folgende Meldung erscheint:

```
                                 Message

Congratulations! You now have FreeBSD installed on your system.

We will now move on to the final configuration questions.
For any option you do not wish to configure, simply select No.

If you wish to re-enter this utility after the system is up, you may
do so by typing: /usr/sbin/sysinstall.

                                  [ OK ]

                       [  Press enter or space  ]
```

Drücken Sie die Taste **Enter**, um die Nacharbeiten durchzuführen.

Wenn Sie [No] auswählen und **Enter** drücken wird die Installation abgebrochen und das System wird nicht verändert. Die nachstehende Meldung wird angezeigt:

```
                                 Message
Installation complete with some errors.  You may wish to scroll
through the debugging messages on VTY1 with the scroll-lock feature.
You can also choose "No" at the next prompt and go back into the
installation menus to retry whichever operations have failed.

                                  [ OK ]
```

Die Meldung wird angezeigt, weil nichts installiert wurde. Drücken Sie **Enter**, um in das Hauptmenü zurückzukehren. Dort können Sie die Installationsprozedur verlassen.

2.10. Arbeiten nach der Installation

Nach einer erfolgreichen Installation wird das System konfiguriert. Sie können das System direkt konfigurieren oder nach einem Neustart. Nach einem Neustart rufen Sie `sysinstall` auf und wählen den Menüpunkt Configure.

2.10.1. Netzwerkkonfiguration

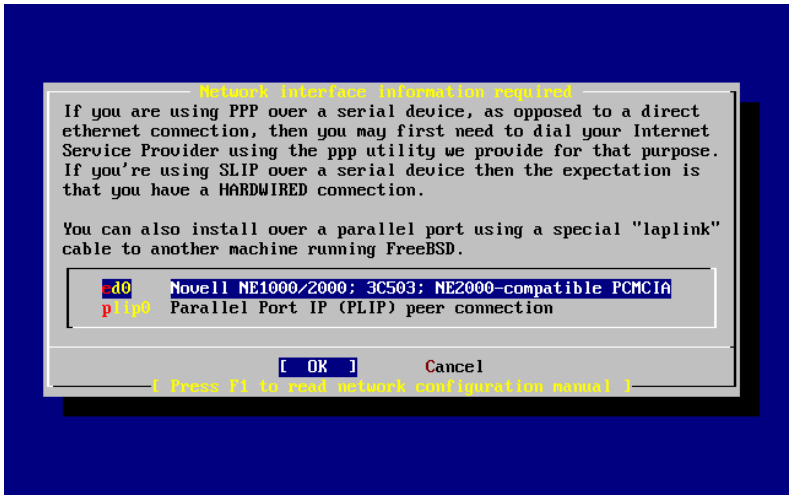
Wenn Sie schon PPP für eine FTP-Installation konfiguriert haben, erscheint dieser Bildschirm nicht. Sie können die Konfiguration später in **sysinstall** vornehmen.

Netzwerke und die Konfiguration von FreeBSD als Gateway oder Router werden eingehend im Kapitel Weiterführende Netzwerkthemen behandelt.

```
                    User Confirmation Requested
  Would you like to configure any Ethernet or PPP network devices?

                              [ Yes ]   No
```

Wenn Sie eine Netzwerkkarte konfigurieren wollen, wählen Sie [Yes] aus und drücken Sie die Taste **Enter**. Wählen Sie [No], um die Netzwerkkonfiguration zu überspringen.

Abbildung 2-28. Eine Netzwerkkarte auswählen



Wählen Sie die zu konfigurierende Karte mit den Pfeiltasten aus und drücken Sie die Taste **Enter**.

```
                    User Confirmation Requested
       Do you want to try IPv6 configuration of the interface?

                              Yes   [ No ]
```

Für das gezeigte Installationsbeispiel genügte das momentan verwendete Internet-Protokoll (IPv4). Daher wurde mit den Pfeiltasten [No] ausgewählt und mit der Taste **Enter** bestätigt.

Wenn Sie durch einen RA-Server mit einem IPv6-Netzwerk verbunden sind, wählen Sie bitte [Yes] und drücken die Taste **Enter**. Die Suche nach den RA-Servern dauert einige Sekunden.

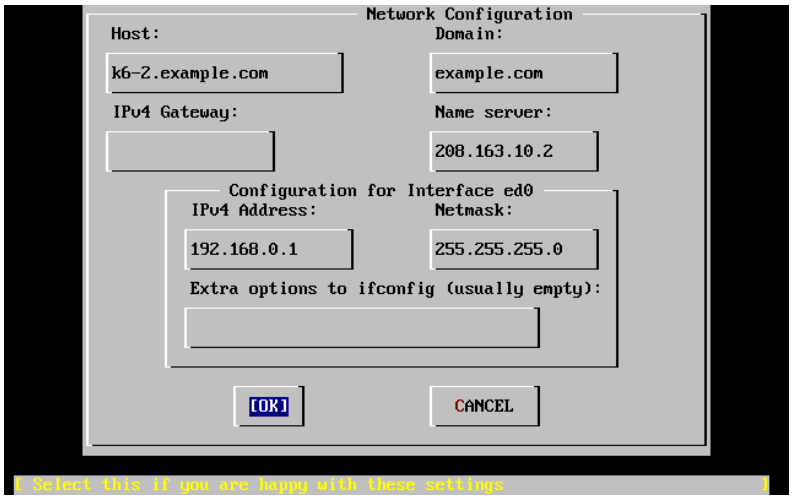
```
                    User Confirmation Requested
Do you want to try DHCP configuration of the interface?

                          Yes   [ No ]
```

Falls Sie das Dynamic Host Configuration Protocol (DHCP) nicht verwenden, wählen Sie [No] aus und drücken Sie **Enter**.

Wenn Sie [Yes] auswählen, wird das Programm **dhclient** ausgeführt und bei Erfolg die Netzwerkkarte konfiguriert. Mehr über DHCP können Sie in Abschnitt 30.5 nachlesen.

Der nächste Bildschirmabzug zeigt die Netzwerkkonfiguration eines Systems, das Gateway für das lokale Netz ist.

Abbildung 2-29. Die Netzwerkkarte ed0 konfigurieren



Tragen Sie in die Felder, die Sie mit der Taste **Tab** auswählen können, die richtige Konfiguration ein.

Host

Der vollständige Rechnername (*fully-qualified hostname*), wie in diesem Beispiel `k6-2.example.com`.

Domain

Der Domain-Name, in dem sich der Rechner befindet. Im Beispiel ist das `example.com`.

IPv4 Gateway

Die IP-Adresse des Rechners, der Pakete an entfernte Netze weiterleitet. Sie müssen dieses Feld ausfüllen, wenn der sich der Rechner in einem Netzwerk befindet. *Lassen Sie das Feld leer*, wenn der Rechner der Gateway in das Internet ist. Der IPv4-Gateway wird auch *default gateway* oder *default route* genannt.

Name server

Die IP-Adresse des lokalen DNS-Servers. Im Beispiel gibt es keinen lokalen DNS-Server, daher wurde der DNS-Server des Providers (`208.163.10.2`) benutzt.

IPv4 address

Die IP-Adresse der Netzwerkkarte (`192.168.0.1`).

Netmask (Netzmaske)

Im Beispiel werden Adressen aus einem Klasse C Netz (`192.168.0.0` bis `192.168.0.255`) benutzt. Standardmäßig besitzt ein Klasse C Netz die Netzmaske `255.255.255.0`.

Extra options to ifconfig (Optionen für ifconfig)

Zusätzliche Optionen für den Befehl `ifconfig`, die spezifisch für die verwendete Netzwerkkarte sind. Im Beispiel sind keine Optionen angegeben.

Wenn Sie alle Werte eingegeben haben, wählen Sie mit **Tab** [OK] aus und drücken Sie **Enter**.

```
                  User Confirmation Requested
       Would you like to bring the ed0 interface up right now?

                          [ Yes ]   No
```

Wenn Sie [Yes] auswählen und **Enter** drücken, wird die Netzwerkkonfiguration aktiviert. Allerdings bringt dies zu diesem Zeitpunkt nicht viel, da der Rechner noch neu gestartet werden muss.

2.10.2. Gateway einrichten

```
                   User Confirmation Requested
      Do you want this machine to function as a network gateway?

                           [ Yes ]    No
```

Wählen Sie [Yes], wenn der Rechner ein Gateway für ein lokales Netz ist und Pakete an andere Netze weiterleitet. Wenn der Rechner ein normaler Netzknoten ist, wählen Sie [No] aus. Bestätigen Sie die auswahl mit der Taste **Enter**.

2.10.3. IP-Dienste einrichten

```
                     User Confirmation Requested
Do you want to configure inetd and the network services that it provides?

                              Yes   [ No ]
```

Wenn [No] ausgewählt wird, werden Dienste wie **telnetd** nicht aktiviert. Benutzer können sich dann von entfernten Rechnern nicht mit **telnet** an dieser Maschine anmelden. Lokale Benutzer können aber auf entfernte Rechner mit **telnet** zugreifen.

Die Dienste können Sie nach der Installation aktivieren, indem Sie die Datei `/etc/inetd.conf` editieren. Dies wird in Abschnitt 30.2.1 beschrieben.

Wenn Sie jetzt weitere Dienste aktivieren möchten, wählen Sie [Yes] aus. Es erscheint die nachstehende Rückfrage:

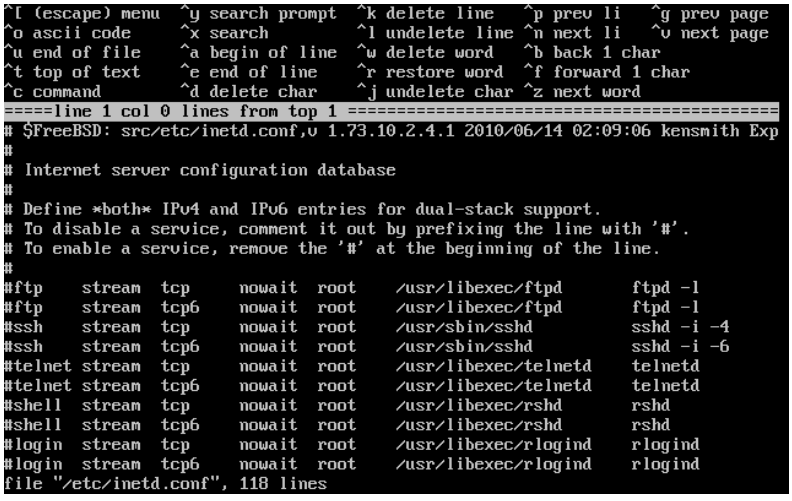
```
                      User Confirmation Requested
The Internet Super Server (inetd) allows a number of simple Internet
services to be enabled, including finger, ftp and telnetd.  Enabling
these services may increase risk of security problems by increasing
the exposure of your system.

With this in mind, do you wish to enable inetd?

                              [ Yes ]   No
```

Bestätigen Sie die Rückfrage mit [Yes].

```
                      User Confirmation Requested
inetd(8) relies on its configuration file, /etc/inetd.conf, to determine
which of its Internet services will be available.  The default FreeBSD
inetd.conf(5) leaves all services disabled by default, so they must be
specifically enabled in the configuration file before they will
function, even once inetd(8) is enabled.  Note that services for
IPv6 must be separately enabled from IPv4 services.

Select [Yes] now to invoke an editor on /etc/inetd.conf, or [No] to
use the current settings.

                              [ Yes ]   No
```

Wenn Sie [Yes] auswählen, können Sie Dienste aktivieren, in dem Sie das Zeichen # am Zeilenanfang entfernen.

Abbildung 2-30. `inetd.conf` editieren



Wenn Sie die gewünschten Dienste aktiviert haben, drücken Sie die Taste **Esc**. Es erscheint ein Menü, in dem Sie die Änderungen abspeichern und den Editor verlassen können.

2.10.4. SSH aktivieren

```
                    User Confirmation Requested
                 Would you like to enable SSH login?
                          Yes       [  No  ]
```

Durch die Auswahl von [Yes], wird sshd(8), der **OpenSSH**-Daemon aktiviert. Danach ist es möglich, sich über eine verschlüsselte Verbindung auf Ihrem System anzumelden. Weitere Informationen über **OpenSSH** finden Sie in Abschnitt 15.10 des FreeBSD-Handbuchs.

2.10.5. Anonymous-FTP

```
                      User Confirmation Requested
 Do you want to have anonymous FTP access to this machine?

                               Yes    [ No ]
```

2.10.5.1. Anonymous-FTP verbieten

Wenn Sie die vorgegebene Auswahl [No] mit der Taste **Enter** bestätigen, können Benutzer, die ein Konto und ein Passwort auf dem System besitzen, immer noch mit FTP auf das System zugreifen.

2.10.5.2. Anonymous-FTP erlauben

Wenn Sie Anonymous-FTP erlauben, darf jeder auf Ihr System zugreifen. Bedenken Sie die Folgen für die Systemsicherheit (siehe Kapitel 15) bevor Sie diese Option aktivieren.

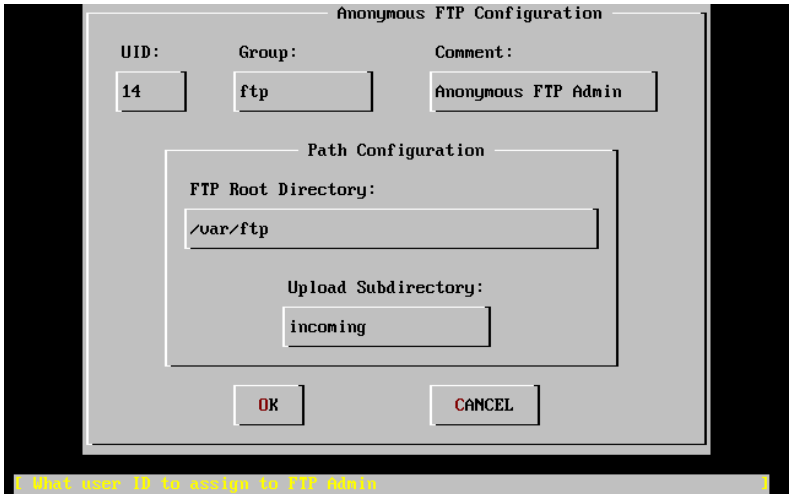
Um Anonymous-FTP zu aktivieren, wählen Sie mit den Pfeiltasten [Yes] aus und drücken Sie die Taste **Enter**. Es erscheint folgende Meldung:

```
                       User Confirmation Requested
 Anonymous FTP permits un-authenticated users to connect to the system
 FTP server, if FTP service is enabled.  Anonymous users are
 restricted to a specific subset of the file system, and the default
 configuration provides a drop-box incoming directory to which uploads
 are permitted.  You must separately enable both inetd(8), and enable
 ftpd(8) in inetd.conf(5) for FTP services to be available.  If you
 did not do so earlier, you will have the opportunity to enable inetd(8)
 again later.

 If you want the server to be read-only you should leave the upload
 directory option empty and add the -r command-line option to ftpd(8)
 in inetd.conf(5)

 Do you wish to continue configuring anonymous FTP?

                            [ Yes ]         No
```

Diese Nachricht informiert Sie darüber, dass der FTP-Dienst auch in der Datei `/etc/inetd.conf` aktiviert werden muss, wenn Sie anonyme FTP-Verbindungen erlauben wollen (lesen Sie dazu auch Abschnitt 2.10.3 des

FreeBSD-Handbuchs). Wählen Sie [Yes] und drücken Sie **Enter**, um fortzufahren. Danach erscheint der folgende Bildschirm:

Abbildung 2-31. Anonymous-FTP konfigurieren



Mit der Taste **Tab** wechseln Sie zwischen den Feldern, in die Sie die benötigten Informationen eingeben.

UID

Die User-ID, die dem anonymen FTP-Benutzer zugewiesen werden soll. Alle hochgeladenen Dateien werden diesem User-ID gehören.

Group

Die Gruppe, zu der der anonyme FTP-Benutzer gehören soll.

Comment

Eine Beschreibung dieses Benutzers in der Datei `/etc/passwd`.

FTP Root Directory

Ort, an dem Dateien für anonymen FTP-Zugang bereitgestellt werden sollen.

Upload Subdirectory

Das Verzeichnis, in dem von einem anonymen FTP-Benutzer hochgeladene Dateien gespeichert werden.

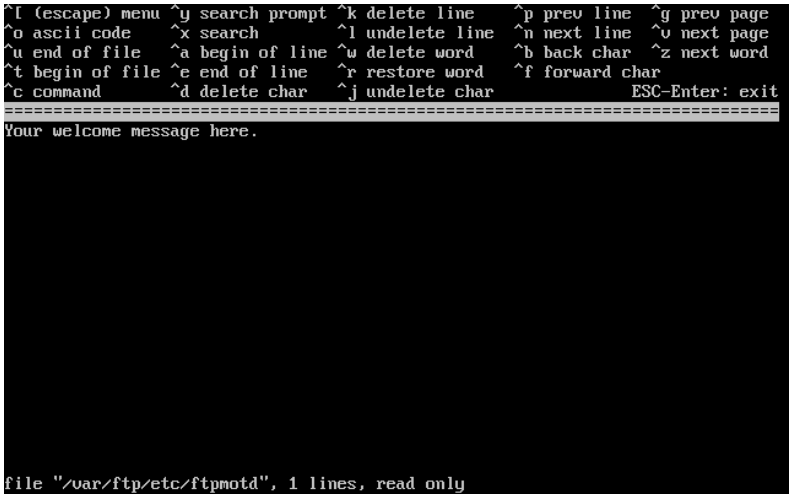
Das FTP-Wurzelverzeichnis wird per Voreinstellung in `/var` angelegt. Wenn in `/var` zu wenig Platz vorhanden ist, können Sie das FTP-Wurzelverzeichnis beispielsweise nach `/usr/ftp` verlegen.

Wenn Sie mit den Einstellungen zufrieden sind, drücken Sie die Taste **Enter**.

```
                    User Confirmation Requested
        Create a welcome message file for anonymous FTP users?

                              [ Yes ]    No
```

Wenn Sie [Yes] auswählen und mit **Enter** bestätigen, können Sie die Begrüßungsmeldung des FTP-Servers in einem Editor ändern.

Abbildung 2-32. Begrüßungsmeldung des FTP-Servers editieren



Der Editor, in dem Sie sich befinden, heißt `ee`. Folgen Sie den Anweisungen, um die Meldung zu editieren. Sie können die Meldung auch später in einem Editor Ihrer Wahl editieren. Merken Sie sich dazu den Dateinamen, der im Editor unten angezeigt wird.

Wenn Sie die Taste **Esc** drücken, erscheint ein Menü, in dem a) leave editor vorgewählt ist. Drücken Sie die Taste **Enter**, um den Editor zu verlassen. Falls Sie Änderungen vorgenommen haben, bestätigen Sie die Änderungen nochmals mit **Enter**.

2.10.6. Network-File-System einrichten

Mit dem Network-File-System (NFS) können Sie über ein Netzwerk auf Dateien zugreifen. Ein Rechner kann NFS-Server, NFS-Client oder beides sein. NFS wird in Abschnitt 30.3 besprochen.

2.10.6.1. NFS-Server einrichten

```
                      User Confirmation Requested
 Do you want to configure this machine as an NFS server?

                                Yes    [ No ]
```

Wenn Sie keinen NFS-Server benötigen, wählen Sie [No] aus und bestätigen Sie mit **Enter**.

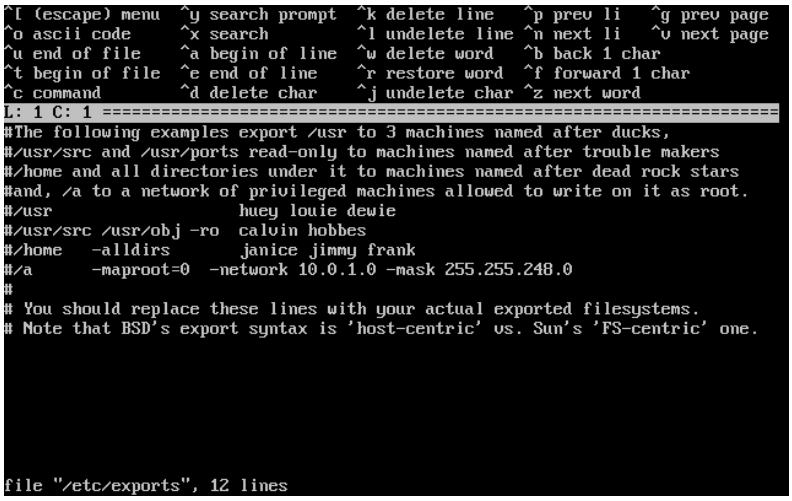
Wenn Sie [Yes] auswählen, erscheint der Hinweis, dass die Datei `exports` angelegt werden muss.

```
                               Message
Operating as an NFS server means that you must first configure an
/etc/exports file to indicate which hosts are allowed certain kinds of
access to your local filesystems.
```

```
Press [Enter] now to invoke an editor on /etc/exports
                                [ OK ]
```

Drücken Sie **Enter** und es wird ein Editor gestartet, in dem Sie die Datei `exports` editieren können.

Abbildung 2-33. `exports` editieren



Folgen Sie den Anweisungen, um Dateisysteme zu exportieren. Sie können die Datei auch später in einem Editor Ihrer Wahl editieren. Merken Sie sich dazu den Dateinamen, der im Editor unten angezeigt wird.

Drücken Sie die Taste **Esc** und es erscheint ein Menü, in dem a) leave editor vorgewählt ist. Drücken Sie die Taste **Enter**, um den Editor zu verlassen.

2.10.6.2. NFS-Client einrichten

Mit einem NFS-Client können Sie auf NFS-Server zugreifen.

```
                        User Confirmation Requested
 Do you want to configure this machine as an NFS client?

                                  Yes   [ No ]
```

Wählen Sie entweder [Yes] oder [No] aus und drücken Sie **Enter**.

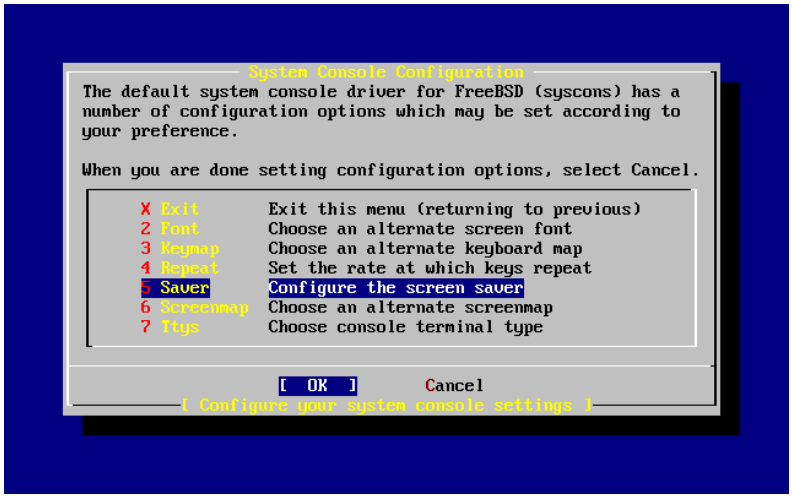
2.10.7. Die Systemkonsole einrichten

Sie können verschiedene Merkmale der Systemkonsole anpassen.

```
                      User Confirmation Requested
       Would you like to customize your system console settings?

                                  [ Yes ]  No
```

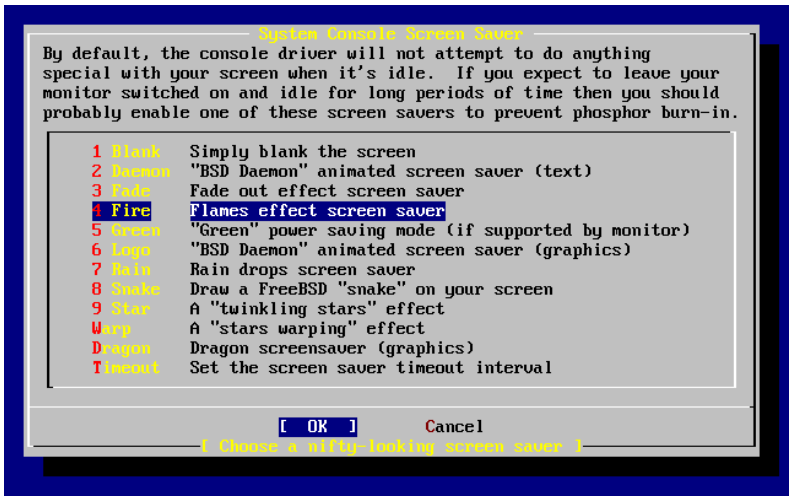
Wenn Sie die Merkmale der Systemkonsole anpassen wollen, wählen Sie [Yes] aus und drücken Sie die Taste **Enter**.

Abbildung 2-34. Merkmale der Systemkonsole



Oft wird ein Bildschirmschoner auf der Konsole aktiviert. Wälen Sie mit den Pfeiltasten Saver aus und drücken Sie die Taste **Enter**.

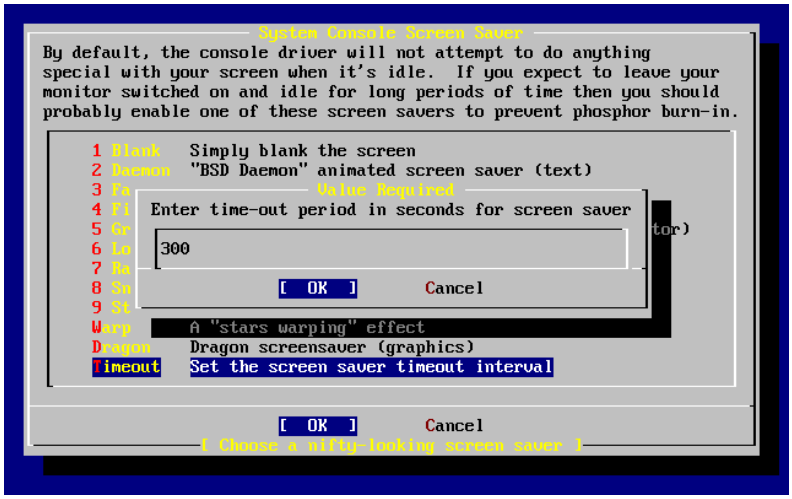
Abbildung 2-35. Bildschirmschoner auswählen



Wählen Sie den gewünschten Bildschirmschoner mit den Pfeiltasten aus und drücken Sie **Enter**. Das Konfigurationsmenü der Systemkonsole erscheint wieder.

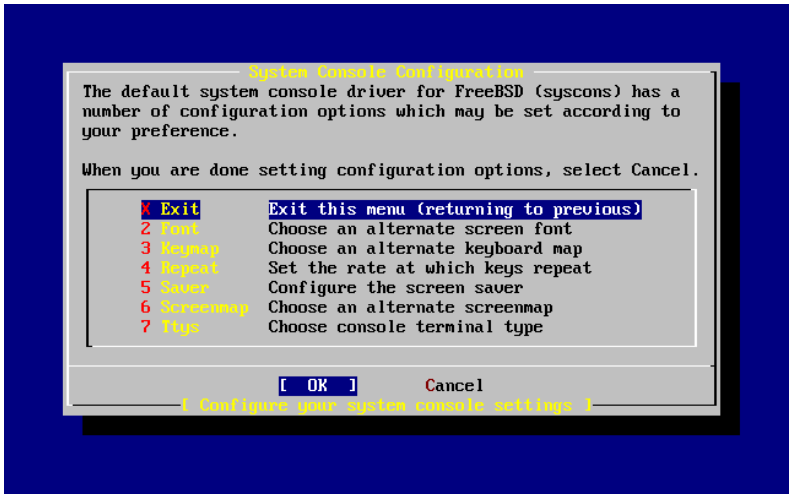
In der Voreinstellung wird der Bildschirmschoner nach 300 Sekunden aktiviert. Um diese Zeitspanne zu ändern, wählen Sie wieder Saver aus. Mit den Pfeiltasten wählen Sie dann Timeout aus und drücken **Enter**. Es erscheint ein Eingabefenster:

Abbildung 2-36. Den Bildschirmschoner einstellen



Ändern Sie die Zeitspanne und wählen Sie [OK] aus. Mit **Enter** kehren Sie in das Konfigurationsmenü der Systemkonsole zurück.

Abbildung 2-37. Die Konfiguration der Systemkonsole verlassen



Um die Nacharbeiten fortzuführen, wählen Sie Exit aus und drücken Sie **Enter**.

2.10.8. Die Zeitzone einstellen

Wenn Sie die Zeitzone richtig einstellen, kann Ihr Rechner automatisch regional bedingte Zeitumstellungen ausführen und andere von der Zeitzone abhängige Funktionen handhaben.

Das folgende Beispiel gilt für den Osten der USA. Ihre Auswahl hängt vom geographischen Standort Ihres Rechners ab.

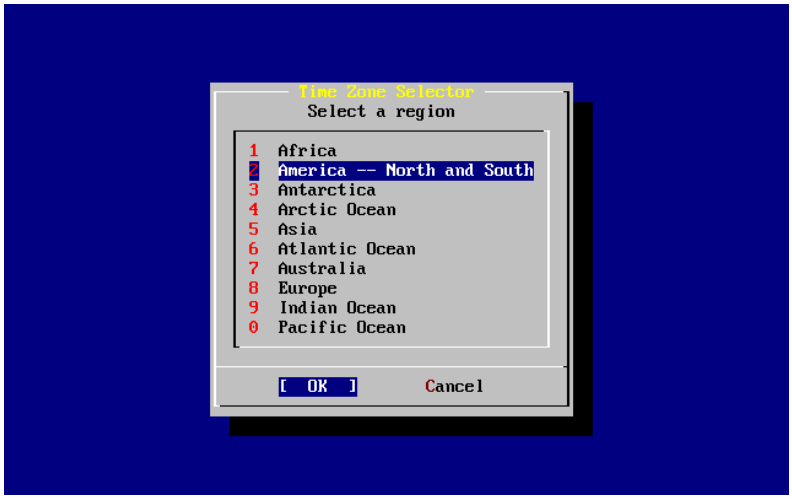
```
                    User Confirmation Requested
          Would you like to set this machine's time zone now?

                              [ Yes ]   No
```

Um die Zeitzone einzustellen, wählen Sie [Yes] und drücken **Enter**.

```
                         User Confirmation Requested
 Is this machine's CMOS clock set to UTC? If it is set to local time
 or you don't know, please choose NO here!

                                  Yes   [ No ]
```

Je nachdem ob die Systemzeit die Zeitzone UTC verwendet, wählen Sie [Yes] oder [No] aus. Bestätigen Sie die Auswahl mit der Taste **Enter**.

Abbildung 2-38. Das Gebiet auswählen

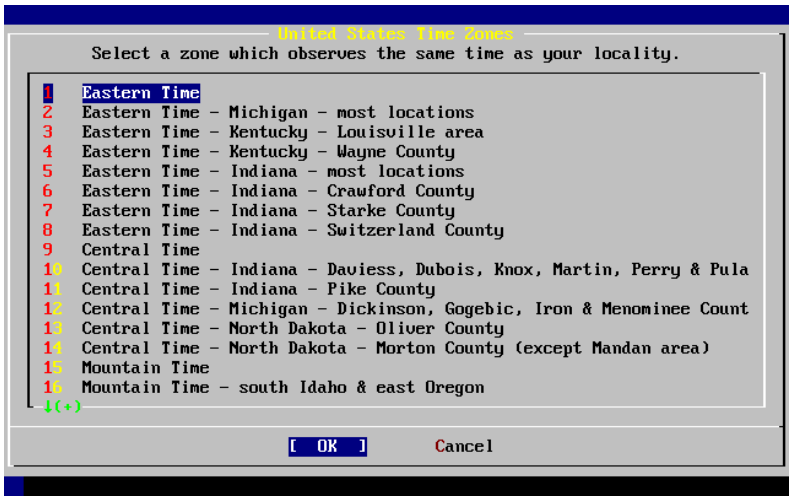


Wählen Sie mit den Pfeiltasten das richtige Gebiet aus und drücken Sie **Enter**.

Abbildung 2-39. Das Land auswählen

Wählen Sie mit den Pfeiltasten das richtige Land aus und drücken Sie **Enter**.

Abbildung 2-40. Die Zeitzone auswählen



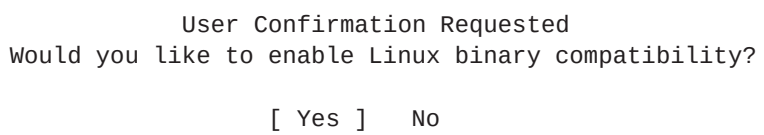
Wählen Sie mit den Pfeiltasten die richtige Zeitzone aus drücken Sie **Enter**.

```
                    Confirmation
Does the abbreviation 'EDT' look reasonable?

                    [ Yes ]   No
```

Wenn die angezeigte Abkürzung der Zeitzone richtig ist, bestätigen Sie diese mit der Taste **Enter**.

2.10.9. Linux-Kompatibilität

Anmerkung: Die folgenden Anweisungen sind nur für FreeBSD 7.X gültig. Installieren Sie eine FreeBSD 8.X-Version, wird der folgende Bildschirm nicht angezeigt.

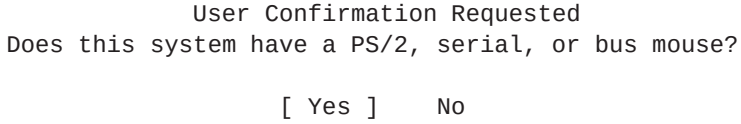


Wenn Sie [Yes] auswählen und **Enter** drücken, können Sie Linux-Software auf FreeBSD laufen lassen. Später wird dazu die notwendige Software installiert.

Wenn Sie über FTP installieren, müssen Sie mit dem Internet verbunden sein. Einige FTP-Server bieten nicht alle verfügbare Software an. Es kann sein, dass die nötige Software für die Linux-Kompatibilität nicht installiert werden kann, dies können Sie später jedoch nachholen.

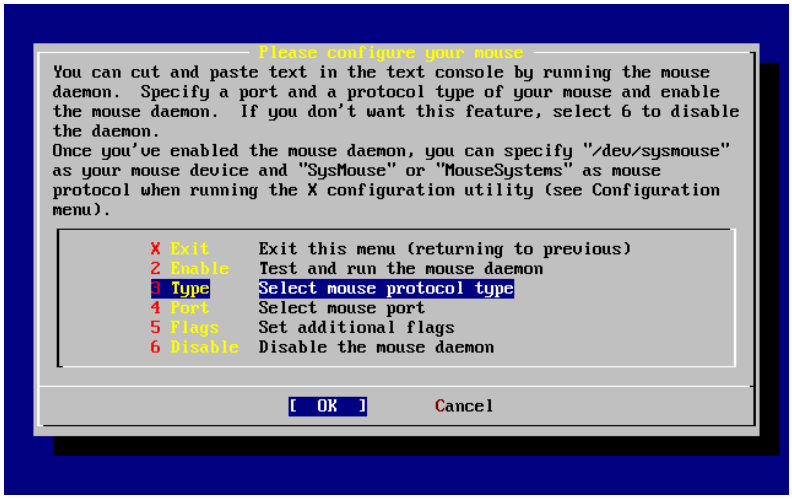
2.10.10. Die Maus konfigurieren

Mit einer 3-Tasten-Maus können Sie Texte auf der Konsole und in Programmen markieren und einfügen (*cut and paste*). Wenn Sie eine 2-Tasten-Maus besitzen, können Sie eine 3-Tasten-Maus emulieren. Lesen Sie dazu nach der Installation die Hilfeseite moused(8). Das folgende Beispiel zeigt die Konfiguration einer nicht-USB-Maus (PS/2 oder serielle Maus):



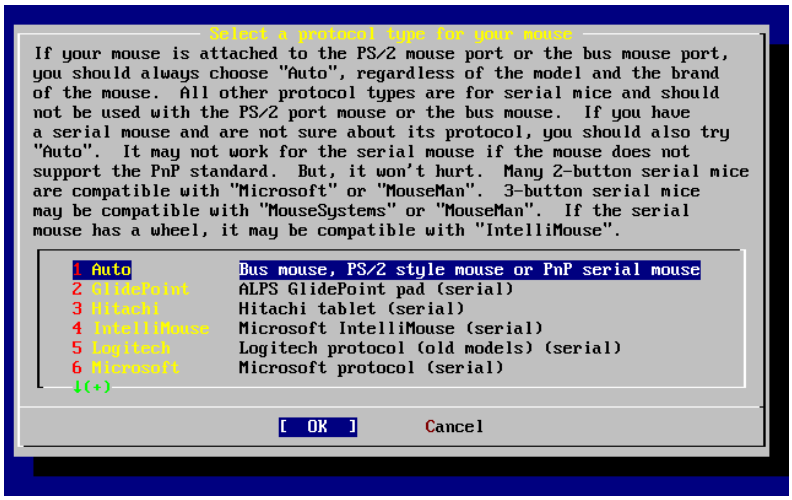
Wählen Sie [Yes] für eine PS/2-, eine serielle oder eine Bus-Maus. Haben Sie hingegen eine USB-Maus, wählen Sie [No]. Danach drücken Sie **Enter**.

Abbildung 2-41. Das Mausprotokoll festlegen



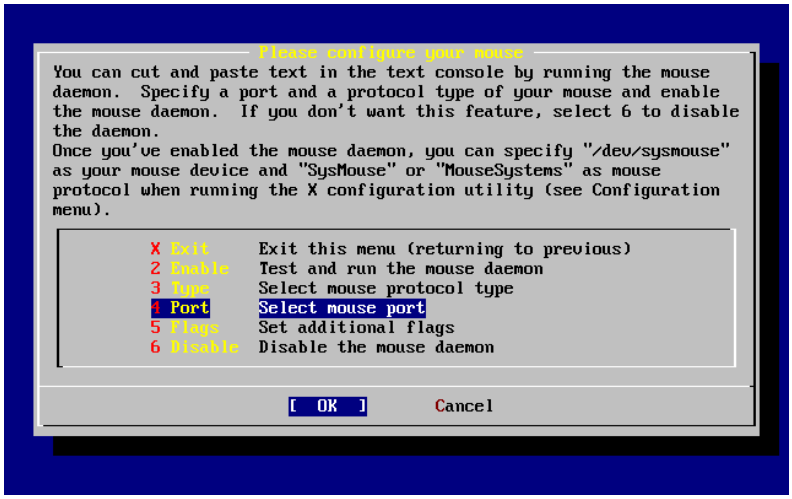
Markieren Sie mit den Pfeiltasten Type und drücken Sie press **Enter**.

Abbildung 2-42. Das Mausprotokoll festlegen



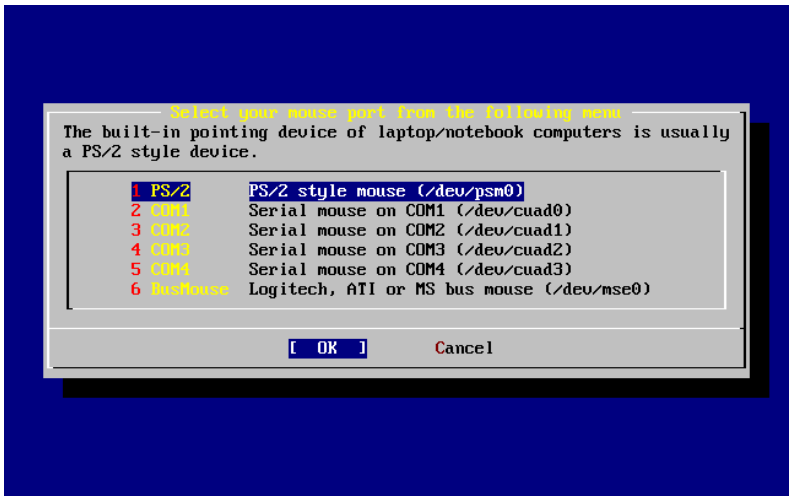
Im Beispiel wurde eine PS/2-Maus verwendet, sodass die Vorgabe Auto passend war. Sie können das Protokoll mit den Pfeiltasten ändern. Stellen Sie sicher, dass [OK] aktiviert ist und verlassen Sie das Menü mit der Taste **Enter**.

Abbildung 2-43. Den Mausport einstellen



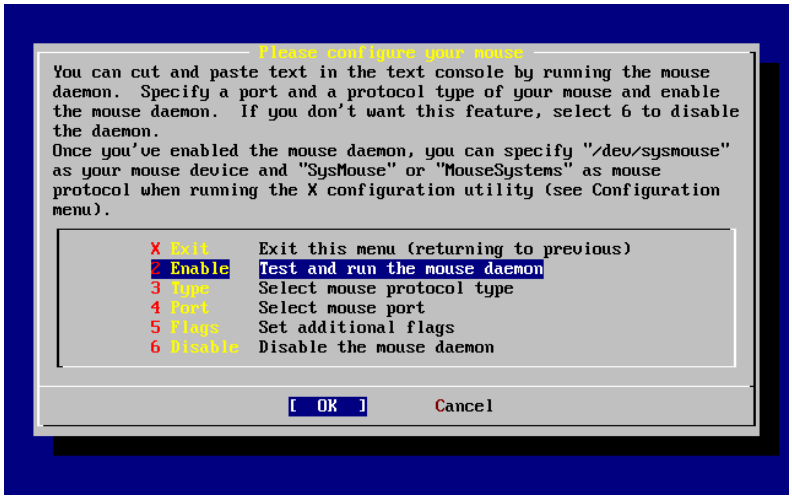
Wählen Sie mit den Pfeiltasten Port und drücken Sie die Taste **Enter**.

Abbildung 2-44. Den Mausport einstellen



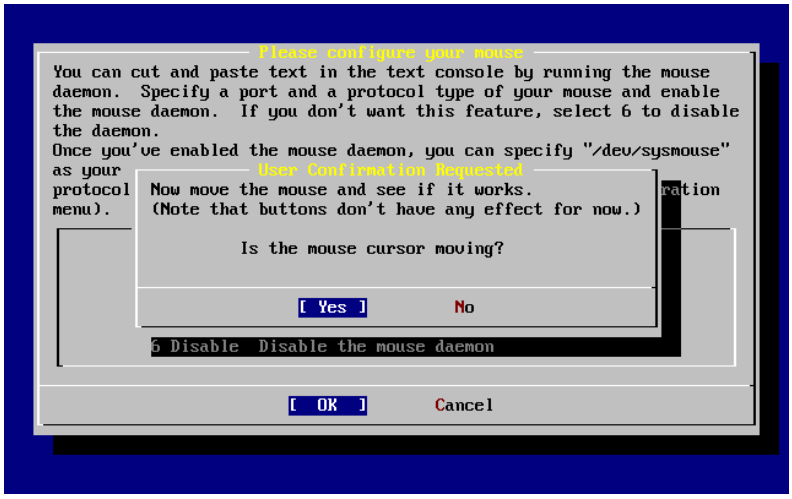
Im Beispiel wurde eine PS/2-Maus verwendet, sodass die Vorgabe PS/2 richtig war. Sie können den Port mit den Pfeiltasten ändern. Bestätigen Sie die Auswahl mit der Taste **Enter**.

Abbildung 2-45. Den Mouse-Daemon aktivieren



Wählen Sie nun mit den Pfeiltasten Enable aus und drücken Sie die Taste **Enter**, um den Mouse-Daemon zu aktivieren und zu testen.

Abbildung 2-46. Den Mouse-Daemon testen



Bewegen Sie die Maus hin und her und prüfen Sie, dass sich der Mauszeiger entsprechend bewegt. Wenn alles in Ordnung ist, wählen Sie [Yes] aus und drücken Sie **Enter**. Wenn sich die Maus nicht richtig verhält, wurde sie nicht korrekt konfiguriert. Wählen Sie in diesem Fall [No] und versuchen Sie, die Einstellungen zu korrigieren.

Um mit den Nacharbeiten fortzufahren, wählen Sie mit den Pfeiltasten Exit aus und drücken Sie **Enter**.

2.10.11. Pakete installieren

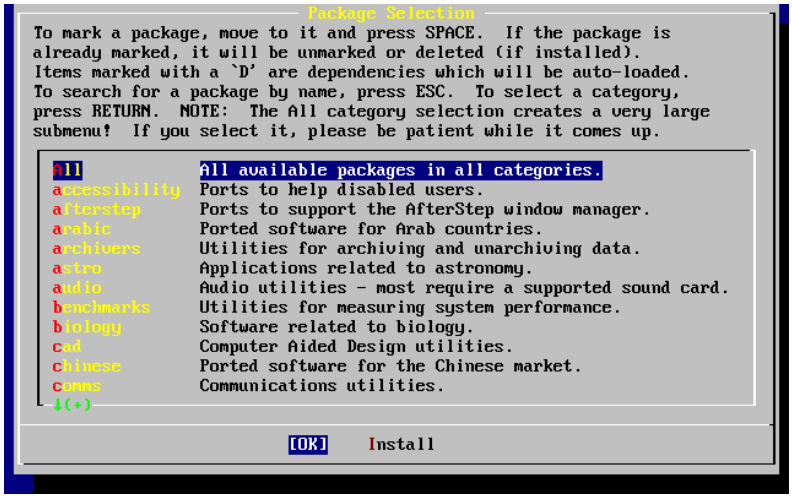
Pakete (*packages*) sind schon übersetzte Programme und sind ein zweckmäßiger Weg, Programme zu installieren.

Beispielhaft wird im Folgenden die Installation eines Paketes gezeigt. In diesem Schritt können auch weitere Pakete installiert werden. Nach der Installation können Sie mit `sysinstall` zusätzliche Pakete installieren.

```
                    User Confirmation Requested
The FreeBSD package collection is a collection of hundreds of
ready-to-run applications, from text editors to games to WEB servers
and more. Would you like to browse the collection now?

                              [ Yes ]   No
```

Nachdem Sie [Yes] ausgewählt und **Enter** gedrückt haben, gelangen Sie in die Paketauswahl:

Abbildung 2-47. Die Paketkategorie aussuchen

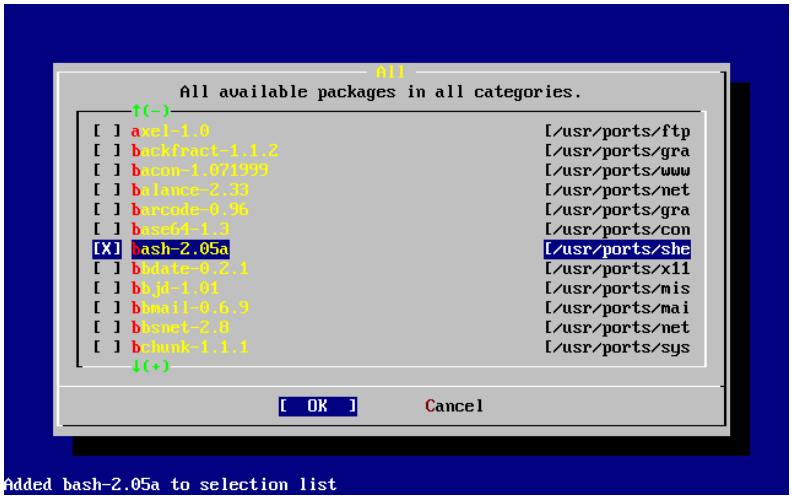


Es stehen nur die Pakete zur Auswahl, die sich auf dem momentanen Installationsmedium befinden.

Wenn Sie All auswählen, werden alle Pakete angezeigt. Sie können die Anzeige auf die Pakete einer Kategorie beschränken. Wählen Sie mit den Pfeiltasten die Kategorie aus und drücken Sie die Taste **Enter**.

Ein Menü mit allen Paketen der ausgewählten Kategorie erscheint:

Abbildung 2-48. Pakete auswählen



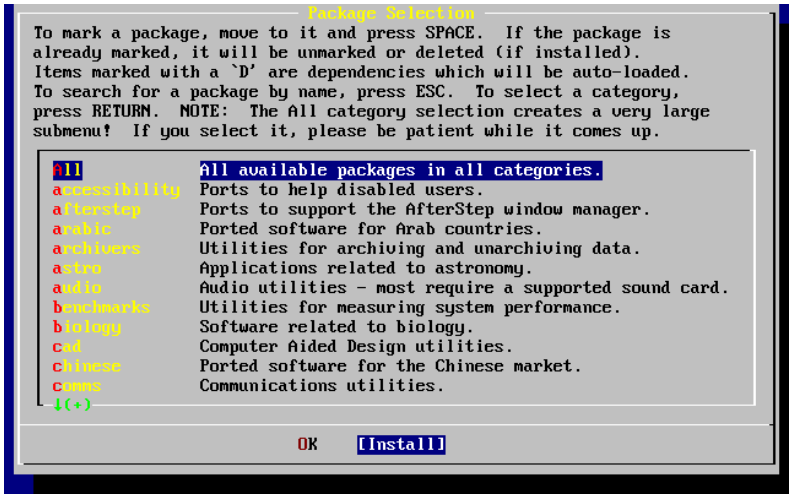
Im gezeigten Bildschirm ist das Paket **bash** ausgewählt. Sie können weitere Pakete auswählen, indem Sie die Pakete mit den Pfeiltasten markieren und die Taste **Space** drücken. In der unteren linken Ecke des Bildschirms wird eine Kurzbeschreibung des ausgewählten Pakets angezeigt.

Die Taste **Tab** wechselt zwischen dem zuletzt ausgesuchten Paket, [OK] und [Cancel].

Wenn Sie die zu installierenden Pakete ausgewählt haben, drücken Sie einmal **Tab**, um [OK] zu markieren. Drücken Sie dann **Enter**, um wieder in die Paketauswahl zu gelangen.

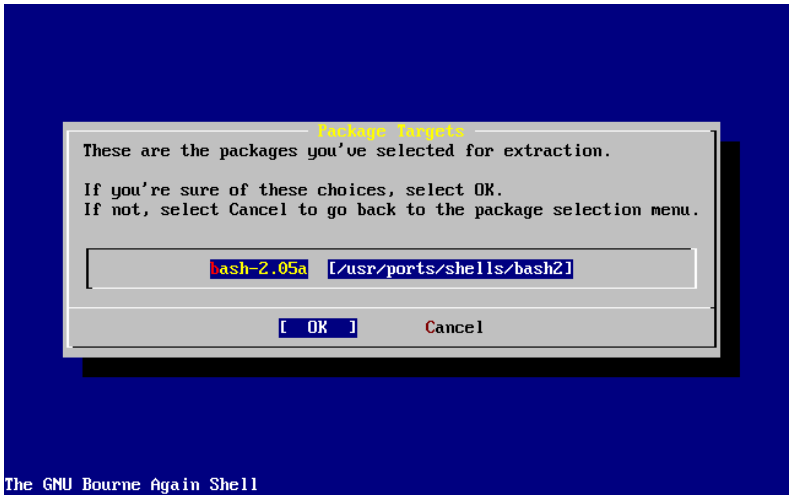
Die rechte und die linke Pfeiltaste wechseln ebenfalls zwischen [OK] und [Cancel]. Mit diesen Tasten können Sie auch [OK] auswählen und dann mit **Enter** zur Paketauswahl zurückkehren.

Abbildung 2-49. Pakete installieren



Benutzen Sie die Taste **Tab** und die Pfeiltasten um [Install] auszuwählen. Drücken Sie anschließend die Taste **Enter**. Sie müssen jetzt die Installation der Pakete bestätigen:

Abbildung 2-50. Paketinstallation bestätigen



Die Paketinstallation wird gestartet, wenn Sie [OK] auswählen und **Enter** drücken. Den Verlauf der Installation können Sie anhand der angezeigten Meldungen verfolgen; achten Sie dabei auf Fehlermeldungen.

Nach der Paketinstallation können Sie die Nacharbeiten fortsetzen. Wenn Sie keine Pakete ausgewählt haben und die Nacharbeiten fortsetzen möchten, wählen Sie trotzdem [Install] aus.

2.10.12. Benutzer und Gruppen anlegen

Während der Installation sollten Sie mindestens ein Benutzerkonto anlegen, sodass Sie das System ohne das Konto `root` benutzen können. Normalerweise ist die Root-Partition recht klein und läuft schnell voll, wenn Sie Anwendungen unter dem `root`-Konto laufen lassen. Vor der größten Gefahr warnt der nachstehende Hinweis:

```
                    User Confirmation Requested
 Would you like to add any initial user accounts to the system? Adding
 at least one account for yourself at this stage is suggested since
 working as the "root" user is dangerous (it is easy to do things which
 adversely affect the entire system).

                             [ Yes ]   No
```

Der Bildschirm auf Deutsch:

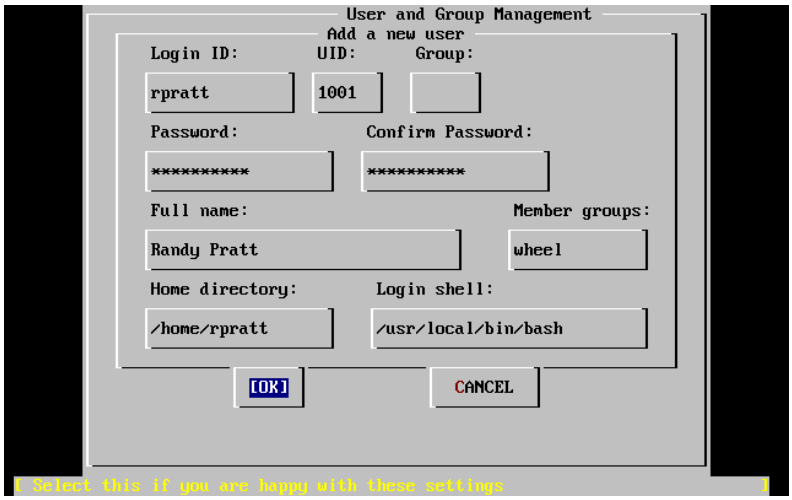
```
                     Bestätigung erforderlich
 Wollen Sie Benutzerkonten anlegen?  Wir empfehlen, mindestens
 ein Konto für sich selbst anzulegen, da es gefährlich
 ist, unter "root" zu arbeiten (es ist leicht, Befehle einzugeben,
 die das System nachhaltig beeinträchtigen).

                             [ Yes ]   No
```

Um ein Benutzerkonto anzulegen, wählen Sie [Yes] aus und drücken **Enter**.

Abbildung 2-51. Benutzerkonto auswählen

Markieren Sie User mit den Pfeiltasten und drücken Sie die Taste **Enter**.

Abbildung 2-52. Benutzerkonto anlegen



Wählen Sie die Felder zum Ausfüllen mit der Taste **Tab** aus. Zur Hilfe werden die nachstehenden Beschreibungen werden im unteren Teil des Bildschirms angezeigt:

Login ID

Der Name des Benutzerkontos (verpflichtend).

UID

Die numerische ID dieses Kontos. Wenn Sie das Feld leer lassen, wird eine ID automatisch zugeteilt.

Group

Die diesem Konto zugeordnete Login-Gruppe. Wenn Sie das Feld leer lassen, wird automatisch eine Gruppe zugeteilt.

Password

Das Passwort des Benutzerkontos. Füllen Sie dieses Feld sehr sorgfätig aus.

Full name

Der vollständige Name des Benutzers (Kommentarfeld).

Member groups

Die Gruppen, in denen dieses Konto Mitglied ist (das Konto erhält Zugriffsrechte auf Dateien dieser Gruppe).

Home directory

Das Heimatverzeichnis des Benutzerkontos. Wenn Sie das Feld leer lassen, wird das Verzeichnis automatisch festgelegt.

Login shell

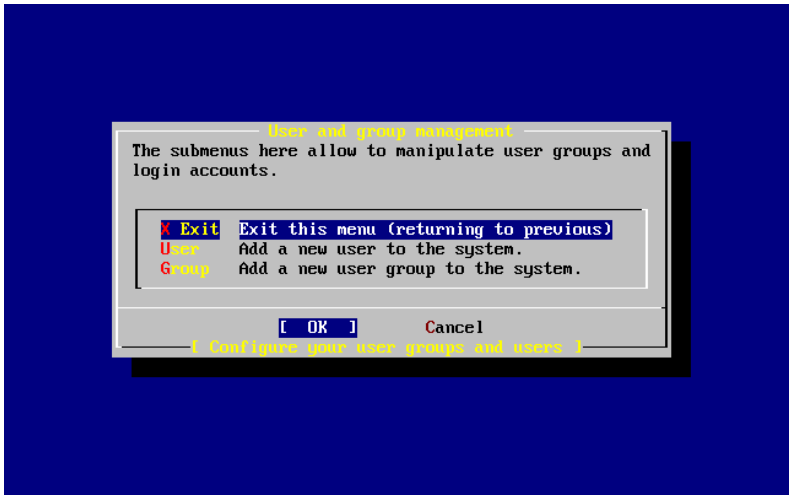
Die Login-Shell des Kontos. Wenn Sie das Feld leer lassen, wird `/bin/sh` als Login-Shell festgesetzt.

Im Beispiel wurde die Login-Shell von `/bin/sh` zu der vorher installierten `/usr/local/bin/bash` geändert. Tragen Sie keine Shell ein, die nicht existiert, da sich sonst nicht anmelden können. In der BSD-Welt wird häufig die C-Shell benutzt, die Sie mit `/bin/tcsh` angeben können.

Damit ein Wechsel auf den Superuser `root` möglich ist, wurde dem Benutzerkonto die Gruppe `wheel` zugeordnet.

Wenn Sie zufrieden sind, drücken Sie [OK]. Es erscheint wieder das Benutzer-Menü:

Abbildung 2-53. Benutzermenü verlassen



Weitere Gruppen können, wenn Sie die Anforderungen schon kennen, zu diesem Zeitpunkt angelegt werden. Nach der Installation können Sie Gruppen mit dem Werkzeug `sysinstall` anlegen.

Wenn Sie alle Benutzer angelegt haben, wählen Sie mit den Pfeiltasten Exit aus und drücken Sie die Taste **Enter**.

2.10.13. Das `root`-Passwort festlegen

```
                        Message
Now you must set the system manager's password.
This is the password you'll use to log in as "root".

                          [ OK ]

                [ Press enter or space ]
```

Um das `root`-Passwort festzulegen, drücken Sie die Taste **Enter**.

Sie müssen das Passwort zweimal eingeben. Stellen Sie sicher, dass Sie das Passwort nicht vergessen. Beachten Sie, dass bei der Eingabe das Passwort weder ausgegeben wird noch Sterne angezeigt werden.

```
New password :
Retype new password :
```

Nach der erfolgreichen Eingabe des Passworts kann die Installation fortgesetzt werden.

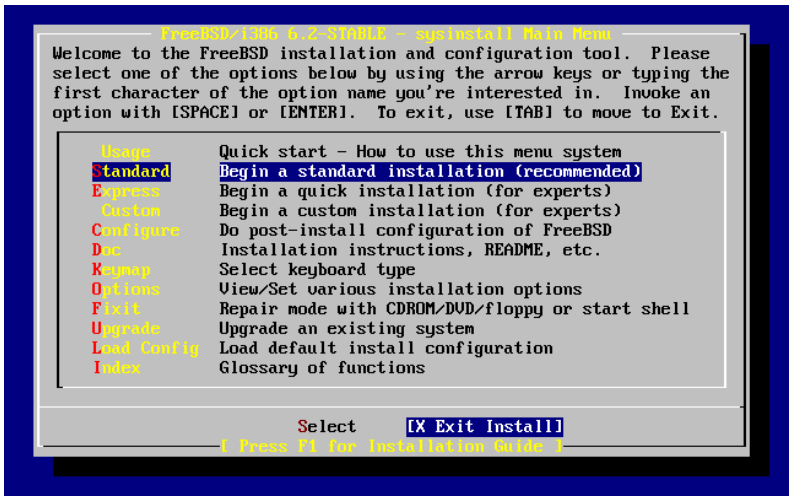
2.10.14. Die Installation beenden

Wenn Sie noch weitere Netzwerkkarten konfigurieren oder weitere Einstellungen vornehmen wollen, können Sie das jetzt tun. Sie können die Einstellungen auch nach der Installation mit `sysinstall` vornehmen.

```
                    User Confirmation Requested
Visit the general configuration menu for a chance to set any last
options?

                              Yes   [ No ]
```

Um in das Hauptmenü zurückzukehren, wählen Sie mit den Pfeiltasten [No] aus und drücken Sie **Enter**.

Abbildung 2-54. Die Installation beenden



Wählen Sie mit den Pfeiltasten [X Exit Install] aus und drücken Sie die Taste **Enter**. Sie müssen das Beenden der Installation bestätigen:

```
                     User Confirmation Requested
 Are you sure you wish to exit? The system will reboot.

                              [ Yes ]   No
```

Wählen Sie [Yes]. Wenn Sie von einer CD-ROM gestartet haben, erhalten Sie die folgende Meldung, die Sie daran erinnert, die CD-ROM aus dem Laufwerk zu entfernen:

```
                     Message
Be sure to remove the media from the drive.

                     [ OK ]
            [ Press enter or space ]
```

Das CD-Laufwerk ist bis zum Neustart des Systems verriegelt. Entfernen Sie die CD zügig, wenn der Rechner startet.

Achten Sie beim Neustart des Systems auf eventuell auftauchende Fehlermeldungen (lesen Sie Abschnitt 2.10.16 für weitere Informationen).

2.10.15. Weitere Netzwerkdienste einrichten

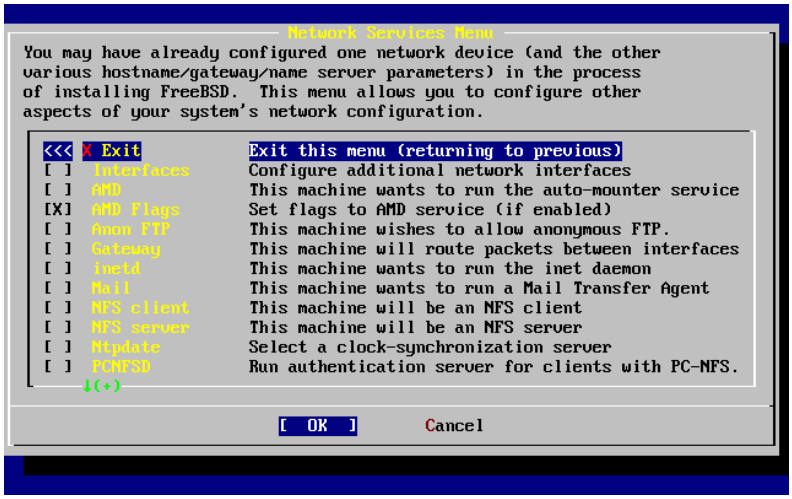
Beigetragen von Tom Rhodes.

Anfänger ohne Vorwissen finden das Einrichten von Netzwerkdiensten oft deprimierend. Netzwerke und das Internet sind für moderne Betriebssysteme von entscheidender Bedeutung. Es ist daher wichtig, die Netzwerkfunktionen von FreeBSD zu kennen. Die von FreeBSD angebotenen Netzwerkdienste können Sie während der Installation kennen lernen.

Netzwerkdienste sind Programme, die Eingaben aus dem Netzwerk entgegennehmen. Es wird große Mühe darauf verwendet, dass diese Programme keinen Schaden verursachen. Leider können auch Programmierern Fehler unterlaufen und es gibt Fälle, in denen Fehler in Netzwerkdiensten von Angreifern ausgenutzt wurden. Es ist daher wichtig, dass Sie nur Dienste aktivieren, die Sie benötigen. Im Zweifallsfall sollten Sie einen Dienst solange nicht aktivieren, bis Sie herausfinden, dass Sie den Dienst benötigen. Einen Dienst können Sie später immer noch mit **sysinstall** oder in der Datei `/etc/rc.conf` aktivieren.

Wählen Sie den Menüpunkt Networking und es erscheint ein Menü wie das nachstehende:

Abbildung 2-55. Netzwerkdienste – obere Hälfte



Die erste Option, Interfaces, wurde schon in Abschnitt 2.10.1 konfiguriert. Sie können daher diesen Punkt überspringen.

Der Punkt AMD aktiviert einen Dienst, der automatisch Dateisysteme einhängt. Normalerweise wird der Dienst zusammen mit dem NFS-Protokoll (siehe unten) verwendet, um automatisch entfernte Dateisysteme einzuhängen. Dieser Menüpunkt erfordert keine weitere Konfiguration.

Der nächste Menüpunkt ist AMD Flags. Wenn Sie den Punkt auswählen, erscheint ein Fenster, in dem Sie AMD-spezifische Optionen eingeben können. Die nachstehenden Optionen sind schon vorgegeben:

```
-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

Die Option `-a` legt das Verzeichnis fest (hier `/.amd_mnt`), unter dem Dateisysteme eingehangen werden. Die Option `-l` legt die Protokolldatei fest. Wenn **syslogd** verwendet wird, werden alle Meldungen an den Daemon **syslogd** gesendet. Das Verzeichnis `/host` dient zum Zugriff auf exportierte Verzeichnisse von entfernten Rechnern, das Verzeichnis `/net` dient zum Zugriff auf exportierte Verzeichnisse von entfernten IP-Adressen. Die Datei `/etc/amd.map` enthält die Einstellungen für von AMD verwaltete Dateisysteme.

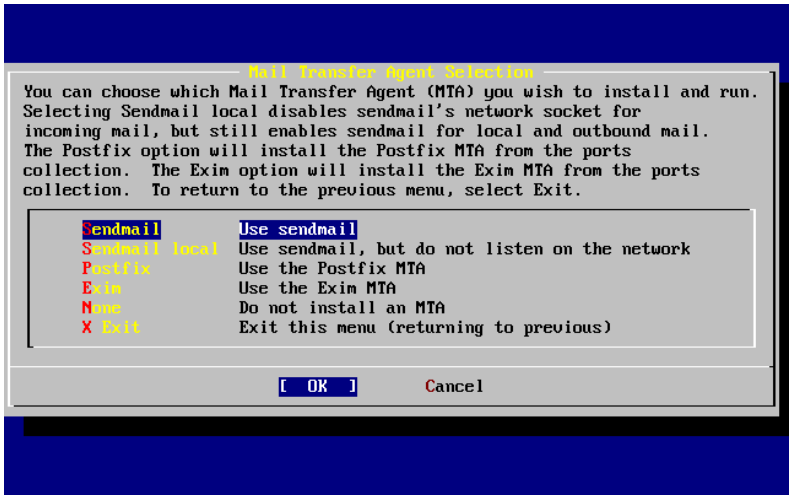
Die Auswahl Anon FTP erlaubt Anonymous-FTP-Verbindungen. Wählen Sie diese Option, wenn Sie einen Anonymous-FTP-Server einrichten wollen. Seien Sie sich über die Sicherheitsrisiken bewusst, wenn Sie Anonymous-FTP erlauben. Die Sicherheitsrisiken und die Konfiguration von Anonymous-FTP werden in einem gesonderten Fenster erklärt, das aufgeht, wenn Sie diese Option auswählen.

Der Menüpunkt Gateway konfiguriert das System, wie vorher erläutert, als Gateway. Wenn Sie während der Installation den Rechner aus Versehen als Gateway konfiguriert haben, können Sie dies hier wieder rückgängig machen.

Der Menüpunkt Inetd konfiguriert, wie schon oben besprochen, den Daemon inetd(8).

Die Auswahl Mail konfiguriert den Mail Transfer Agent (MTA) des Systems. Wenn Sie diesen Punkt auswählen, erscheint das folgende Menü:

Abbildung 2-56. Den MTA festlegen



In diesem Menü wählen Sie aus, welcher MTA installiert und benutzt wird. Ein MTA ist ein Mail-Server, der E-Mails an lokale Empfänger oder an Empfänger im Internet ausliefert.

Die Auswahl Sendmail installiert das verbreitete **sendmail** (in FreeBSD die Voreinstellung). Die Auswahl Sendmail local verwendet **sendmail** als MTA, deaktiviert aber den Empfang von E-Mails aus dem Internet. Postfix und Exim sind ähnlich wie Sendmail. Beide Programme liefern E-Mails aus und einige Anwender verwenden lieber eines der beiden Programme als MTA.

Nachdem Sie einen MTA ausgewählt haben (oder beschlossen haben, keinen MTA zu benutzen), erscheint wieder das Menü Netzwerkdienste. Der nächste Menüpunkt ist NFS client.

Die Auswahl NFS client erlaubt es dem System, mit einem NFS-Server zu kommunizieren. Ein NFS-Server stellt mithilfe des NFS-Protokolls Dateisysteme für andere Systeme auf dem Netzwerk bereit. Wenn der Rechner alleine für sich steht, können Sie diesen Menüpunkt auslassen. Wahrscheinlich müssen Sie noch weitere Einstellungen vornehmen; der Abschnitt 30.3 beschreibt die Einstellungen für NFS-Server und NFS-Clients.

Der Menüpunkt NFS server richtet einen NFS-Server auf dem Rechner ein. Durch die Auswahl dieses Punktes werden die für Remote-Procedure-Call (RPC) benötigten Dienste gestartet. Mit RPC werden Routinen auf entfernten Rechnern aufgerufen.

Der nächste Punkt, Ntpdate, konfiguriert die Zeitsynchronisation. Wenn Sie diesen Punkt auswählen, erscheint das folgende Menü:

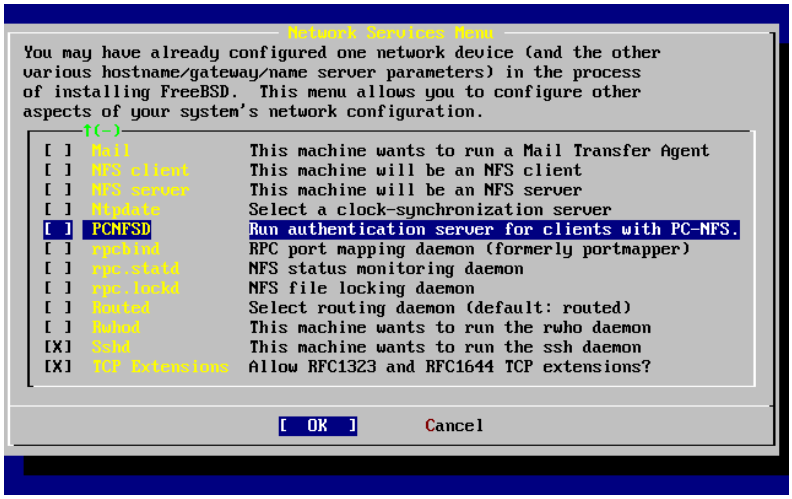
Abbildung 2-57. Ntpdate konfigurieren

Wählen Sie aus diesem Menü einen nahe liegenden Server aus. Die Zeitsynchronisation mit einem nahe liegenden Server ist, wegen der geringeren Latenzzeit, genauer als die Synchronisation mit einem weiter entfernten Server.

Der nächste Menüpunkt ist PCNFSD. Wenn Sie diesen Punkt auswählen, wird `net/pcnfsd` aus der Ports-Sammlung installiert. Dieses nützliche Werkzeug stellt NFS-Authentifizierungsdienste für Systeme bereit, die diese Dienste nicht anbieten (beispielsweise Microsofts MS-DOS).

Um die nächsten Menüpunkte zu sehen, müssen Sie herunterblättern:

Abbildung 2-58. Netzwerkdienste – untere Hälfte



Die Programme rpcbind(8), rpc.statd(8) und rpc.lockd(8) werden für Remote-Procedure-Calls (RPC) benutzt. Das Programm `rpcbind` verwaltet die Kommunikation zwischen NFS-Servern und NFS-Clients und ist für den Betrieb eines NFS-Servers erforderlich. Der Daemon **rpc.statd** hält zusammen mit dem Daemon **rpc.statd** des entfernten Rechners den Status der Verbindung. Der Status einer Verbindung wird normalerweise in der Datei `/var/db/statd.status` festgehalten. Der nächste Menüpunkt ist rpc.lockd, der Dateisperren (*file locks*)

bereitstellt. **rpc.lockd** wird normalerweise zusammen mit dem Daemon **rpc.statd** benutzt, der festhält welche Rechner Sperren anfordern und wie oft Sperren angefordert werden. Beide Dienste sind wunderbar zur Fehlersuche geeignet, doch werden Sie zum Betrieb von NFS-Servern und NFS-Clients nicht benötigt.

Der nächste Punkt in der Auswahl ist Routed, der Routing-Daemon. Das Programm routed(8) verwaltet die Routing-Tabelle, entdeckt Multicast-Router und stellt die Routing-Tabelle auf Anfrage jedem mit dem Netz verbundenen Rechner zur Verfügung. Der Daemon wird hauptsächlich auf Gateways eines lokalen Netzes eingesetzt. Wenn Sie den Punkt auswählen müssen Sie den Ort des Programms angeben. Die Vorgabe können Sie mit der Taste **Enter** übernehmen. Anschließend werden Sie nach den Kommandozeilenoptionen für `routed` gefragt. Vorgegeben ist die Option `-q`.

Der nächste Menüpunkt ist Rwhod. Wenn Sie diesen Punkt auswählen, wird während des Systemstarts der Daemon rwhod(8) gestartet. Das Kommando `rwhod` schickt Broadcast-Meldungen in das Netz oder empfängt diese im Consumer-Mode. Die Funktion der Werkzeuge wird in den Hilfeseiten ruptime(1) und rwho(1) beschrieben.

Der vorletzte Menüpunkt aktiviert den Daemon sshd(8), den **OpenSSH** Secure-Shell-Server. Wo möglich sollte SSH anstelle von **telnet** und FTP eingesetzt werden. Der Secure-Shell-Server erstellt verschlüsselte und daher sichere Verbindungen zwischen zwei Rechnern.

TCP Extensions ist der letzte Menüpunkt. Diese Auswahl aktiviert die TCP-Erweiterungen aus RFC 1323 und RFC 1644. Obwohl dies auf vielen Rechnern die Verbindungsgeschwindigkeit erhöht, können durch diese Option auch Verbindungsabbrüche auftreten. Auf Servern sollte diese Option nicht aktiviert werden, auf Einzelmaschinen kann diese Option nützlich sein.

Wenn Sie die Netzwerkdienste eingerichtet haben, blättern Sie zum Menüpunkt Exit hoch, um die Nacharbeiten fortzusetzen oder verlassen Sie **sysinstall**, indem Sie zweimal X Exit und danach [X Exit Install] wählen.

2.10.16. FreeBSD starten

2.10.16.1. Start von FreeBSD auf FreeBSD/i386

Wenn alles funktioniert hat, laufen viele Meldungen über den Bildschirm und schließlich erscheint ein Anmeldeprompt. Um sich die Meldungen anzusehen. drücken Sie die Taste **Scroll-Lock**. Sie können dann mit den Tasten **PgUp** und **PgDn** blättern. Wenn Sie erneut **Scroll-Lock** drücken, kehren Sie zum Anmeldeprompt zurück.

Es kann sein, dass der Puffer zu klein ist, um alle Meldungen anzuzeigen. Nachdem Sie sich angemeldet haben, können Sie sich mit dem Kommando `dmesg` alle Meldungen ansehen.

Melden Sie sich bitte mit dem Benutzerkonto an (`rpratt` im Beispiel), das Sie während der Installation eingerichtet haben. Arbeiten Sie mit `root` nur dann wenn es erforderlich ist.

Die nachfolgende Abbildung zeigt typische Startmeldungen (Versionsangaben entfernt):

```
Copyright (c) 1992-2002 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.

Timecounter "i8254"  frequency 1193182 Hz
CPU: AMD-K6(tm) 3D processor (300.68-MHz 586-class CPU)
  Origin = "AuthenticAMD"  Id = 0x580  Stepping = 0
  Features=0x8001bf<FPU,VME,DE,PSE,TSC,MSR,MCE,CX8,MMX>
  AMD Features=0x80000800<SYSCALL,3DNow!>
real memory  = 268435456 (262144K bytes)
```

```
config> di sn0
config> di lnc0
config> di le0
config> di ie0
config> di fe0
config> di cs0
config> di bt0
config> di aic0
config> di aha0
config> di adv0
config> q
avail memory = 256311296 (250304K bytes)
Preloaded elf kernel "kernel" at 0xc0491000.
Preloaded userconfig_script "/boot/kernel.conf" at 0xc049109c.
md0: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0: <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci0
usb0: <VIA 83C572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr 1
uhub0: 2 ports with 2 removable, self powered
chip1: <VIA 82C586B ACPI interface> at device 7.3 on pci0
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xe800-0xe81f irq 9 at
device 10.0 on pci0
ed0: address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5” drive> on fdc0 drive 0
atkbdc0: <keyboard controller (i8042)> at port 0x60-0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq 1 on atkbdc0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: model Generic PS/2 mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x1 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
```

```
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
ppbus0: IEEE1284 device found /NIBBLE
Probing for PnP devices on ppbus0:
plip0: <PLIP network interface> on ppbus0
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
ppi0: <Parallel I/O> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master using UDMA33
ad2: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata1-master using UDMA33
acd0: CDROM <DELTA OTC-H101/ST3 F/W by OIPD> at ata0-slave using PIO4
Mounting root from ufs:/dev/ad0s1a
swapon: adding /dev/ad0s1b as swap device
Automatic boot in progress...
/dev/ad0s1a: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1a: clean, 48752 free (552 frags, 6025 blocks, 0.9% fragmentation)
/dev/ad0s1f: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1f: clean, 128997 free (21 frags, 16122 blocks, 0.0% fragmentation)
/dev/ad0s1g: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0s1g: clean, 3036299 free (43175 frags, 374073 blocks, 1.3% fragmentation)
/dev/ad0s1e: filesystem CLEAN; SKIPPING CHECKS
/dev/ad0s1e: clean, 128193 free (17 frags, 16022 blocks, 0.0% fragmentation)
Doing initial network setup: hostname.
ed0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        inet6 fe80::5054::5ff::fede:731b%ed0 prefixlen 64 tentative scopeid 0x1
        ether 52:54:05:de:73:1b
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x8
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
Additional routing options: IP gateway=YES TCP keepalive=YES
routing daemons:.
additional daemons: syslogd.
Doing additional network setup:.
Starting final network daemons: creating ssh RSA host key
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
cd:76:89:16:69:0e:d0:6e:f8:66:d0:07:26:3c:7e:2d root@k6-2.example.com
 creating ssh DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
f9:a1:a9:47:c4:ad:f9:8d:52:b8:b8:ff:8c:ad:2d:e6 root@k6-2.example.com.
setting ELF ldconfig path: /usr/lib /usr/lib/compat /usr/X11R6/lib
/usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout /usr/X11R6/lib/aout
starting standard daemons: inetd cron sshd usbd sendmail.
```

```
Initial rc.i386 initialization:.
rc.i386 configuring syscons: blank_time screensaver moused.
Additional ABI support: linux.
Local package initialization:.
Additional TCP options:.

FreeBSD/i386 (k6-2.example.com) (ttyv0)

login: rpratt
Password:
```

Das Erzeugen der RSA- und DSA-Schlüssel kann auf langsamen Maschinen lange dauern. Die Schlüssel werden nur beim ersten Neustart erzeugt, spätere Neustarts sind schneller.

Wenn der X-Server konfiguriert ist und eine Oberfläche ausgewählt wurde, können Sie X mit dem Kommando `startx` starten.

2.10.17. FreeBSD herunterfahren

Es ist wichtig, dass Sie das Betriebssystem richtig herunterfahren. Wechseln Sie zunächst mit dem Befehl `su` zum Superuser; Sie müssen dazu das `root`-Passwort eingeben. Der Wechsel auf den Superuser gelingt nur, wenn der Benutzer ein Mitglied der Gruppe `wheel` ist. Ansonsten melden Sie sich direkt als Benutzer `root` an. Der Befehl `shutdown -h now` hält das System an.

```
The operating system has halted.
Please press any key to reboot.
```

Sie können den Rechner ausschalten, nachdem die Meldung `Please press any key to reboot` erschienen ist. Wenn Sie stattdessen eine Taste drücken, startet das System erneut.

Sie können das System auch mit der Tastenkombination **Ctrl**+**Alt**+**Del** neu starten. Sie sollten diese Tastenkombination allerdings nicht gewohnheitsmäßig benutzen.

2.11. Fehlersuche

Dieser Abschnitt behandelt häufig auftretende Installationsprobleme. Weiterhin enthält er Hinweise, wie FreeBSD parallel mit MS-DOS oder Windows betrieben wird.

2.11.1. Wenn etwas schief geht

Aufgrund der Beschränkungen der PC-Architektur ist eine zuverlässige Geräteerkennung nicht möglich. Falls die Geräteerkennung fehlschlägt, können Sie einige Dinge versuchen.

Sehen Sie in den Hardware Notes (http://www.FreeBSD.org/de/releases/index.html) Ihrer FreeBSD-Version nach, ob Ihre Hardware unterstützt wird.

Wenn Ihre Hardware unterstützt wird und sich der Installationsprozess aufhängt oder sonstige Probleme auftauchen, müssen Sie einen angepassten Kernel erstellen, da Ihre Hardware in diesem Fall nicht vom `GENERIC`-Kernel unterstützt wird. Der Kernel auf den Startdisketten verwendet die Werkseinstellungen für IRQs, IO-Adressen und

DMA-Kanäle. Geänderte Einstellungen müssen Sie daher in der Kernelkonfigurationsdatei angeben, damit FreeBSD diese Geräte korrekt erkennt.

Es ist auch möglich, dass die Suche nach einem nicht vorhandenen Gerät dazu führt, dass die Erkennung eines vorhandenen Geräts fehlschlägt. In diesem Fall sollten Sie nicht vorhandene Geräte, deren Einstellungen sich mit vorhandenen Geräten überschneiden, deaktivieren.

> **Anmerkung:** Einige Installationsprobleme können Sie vermeiden oder umgehen, indem Sie die Firmware der Hardware, insbesondere die Firmware der Systemplatine, aktualisieren. Die Firmware der Systemplatine ist das BIOS. Die meisten Hardware-Hersteller bieten aktuelle Firmware und Anleitungen zur Aktualisierung der Firmware auf dem Internet an.
>
> Viele Hersteller raten davon ab, ohne guten Grund das BIOS zu aktualisieren. Die Aktualisierung *kann* fehlschlagen und den BIOS-Chip dauerhaft beschädigen.

2.11.2. MS-DOS®- und Windows-Dateisysteme benutzen

Mit **Double Space™** komprimierte Dateisysteme werden zurzeit von FreeBSD nicht unterstützt. Damit FreeBSD auf die Daten zugreifen kann, müssen Sie das Dateisystem daher dekomprimieren. Rufen Sie dazu den **Compression Agent** aus dem Menü Start > Programs > System Tools auf.

FreeBSD unterstützt MS-DOS-Dateisysteme (manchmal auch als FAT-Dateisysteme bezeichnet). Der Befehl mount_msdosfs(8) bindet diese Dateisysteme in den FreeBSD-Verzeichnisbaum ein und erlaubt dadurch den Zugriff auf die darin enthaltenen Daten. mount_msdosfs(8) wird normalerweise nicht direkt, sondern über einen Eintrag in der Datei `/etc/fstab` oder durch den Aufruf des Befehls mount(8) (in Kombination mit den korrekten Parametern).

Ein typischer Eintrag in `/etc/fstab` sieht so aus:

```
/dev/ad0sN  /dos  msdosfs rw  0   0
```

> **Anmerkung:** Das Verzeichnis `/dos` muss bereits vorhanden sein, damit dieser Eintrag funktioniert. Weitere Informationen zu den Einstellungen in der Datei `/etc/fstab` finden sich in der Manualpage fstab(5).

Ein typischer Aufruf von mount(8) zum Einhängen eines MS-DOS-Dateisystems sieht so aus:

```
# mount -t msdosfs /dev/ad0s1 /mnt
```

Das MS-DOS-Dateisystem befindet sich hier auf der ersten Partition der primären Platte. Dies kann bei Ihnen anders sein. Die Anordnung der Partitionen entnehmen Sie den Ausgaben von `dmesg` und `mount`.

> **Anmerkung:** FreeBSD numeriert Platten (genauer MS-DOS-Partitionen) anders als andere Betriebssysteme. Die Nummern von erweiterten Partitionen sind in der Regel höher als die Nummern von primären Partitionen. Das Werkzeug fdisk(8) kann Ihnen dabei helfen, festzustellen, welche Partitionen zu FreeBSD und welche zu einem anderen Betriebssystem gehören.

Analog werden NTFS-Partitionen mit dem Kommando mount_ntfs(8) eingehangen.

2.11.3. Fragen und Antworten zu häufig auftretenden Problemen

1. Mein System hängt sich beim Testen der Hardware auf, oder es verhält sich seltsam während der Installation oder das Diskettenlaufwerk wird nicht getestet.

FreeBSD 5.0 und neuer machen ausgiebig Gebrauch von den ACPI-Systemdiensten zur Systemkonfiguration der i386-, amd64- und ia64-Plattformen, falls diese während des Bootvorgangs gefunden werden. Leider enthalten sowohl der ACPI-Treiber als auch manche Motherboard- und BIOS-Implementierungen für ACPI noch einige Fehler. Kommt es auf Ihrem System zu Problemen, können Sie ACPI daher deaktivieren, indem während des Bootvorganges den “Hint” `hint.acpi.0.disabled` aktivieren:

```
set hint.acpi.0.disabled="1"
```

Da diese Einstellung bei jedem Neustart verloren geht, aktivieren Sie sie dauerhaft, indem Sie die Zeile `hint.acpi.0.disabled="1"` in die Datei `/boot/loader.conf`. Weitere Informationen über den Bootloader finden Sie in Abschnitt 13.1 des FreeBSD-Handbuchs.

2. Direkt nach der Installation beginnt das System zwar zu booten, der Kernel wird geladen und meine Hardware getestet. Dann bricht der Bootvorgang aber mit der folgenden (oder einer ähnlichen) Fehlermeldung ab:

```
changing root device to ad1s1a panic: cannot mount root
```

Was läuft hier falsch? Was kann/muss ich tun?

Was soll ich mit diesem `bios_drive:interface(unit,partition)kernel_name` anfangen, das mir die Hilfefunktion ausgibt?

Dabei handelt es sich um ein lange bekanntes Problem, das nur dann auftritt, wenn es sich bei der Bootplatte nicht um die erste Platte im System handelt. Das BIOS numeriert die Festplatten anders als FreeBSD, daher ist das System manchmal nicht in der Lage, diese Numerierungen selbst automatisch in Einklang zu bringen.

Sollte Ihre Bootplatte nicht die erste Platte im System sein, können Sie FreeBSD dabei helfen, diese Platte zu finden. Es gibt zwei Situationen, in denen Sie FreeBSD mitteilen müssen, wo sich das root-Dateisystem befindet. Dazu müssen Sie die Nummer der Platte im BIOS, den Plattentyp sowie die Nummer der Platte unter FreeBSD angeben.

Im ersten Fall verfügen Sie über zwei IDE-Platten, die beide als Master an ihrem jeweiligen IDE-Controller konfiguriert sind. FreeBSD soll dabei von der zweiten Platte booten. Ihr BIOS erkennt die beiden Platten als “Platte 1” und “Platte 2”, während FreeBSD die Platten als `ad0` und `ad2` erkennt.

Für das BIOS befindet sich FreeBSD auf der Platte Nummer 1, der Typ ist `ad`, und FreeBSD erkennt die Platte als Platte Nummer 2. Daher geben Sie Folgendes ein:

```
1:ad(2,a)kernel
```

Beachten Sie, dass dieser Eintrag nicht notwendig ist, wenn die zweite Platte als Slave am primären IDE-Controller konfiguriert ist (sondern sogar falsch wäre).

Die zweite Situation entsteht, wenn Sie von einer SCSI-Platte booten und zusätzlich eine oder mehrere IDE-Platten installiert haben. In diesem Fall ist die Plattennummer unter FreeBSD kleiner als die Plattennummer im BIOS. Verfügen Sie über zwei IDE-Platten und eine SCSI-Platte, hat die SCSI-Platte im BIOS die Nummer 2, den Typ `da`, und wird von FreeBSD als Platte Nummer 0 erkannt. In diesem Fall geben Sie daher Folgendes ein:

```
2:da(0,a)kernel
```

Durch diese Zeile teilen Sie FreeBSD mit, dass Sie von der BIOS-Platte Nummer 2 booten wollen (bei der es sich um die erste SCSI-Platte Ihres Systems handelt). Verfügen Sie nur über eine IDE-Platte, geben Sie hingegen `1:` ein.

Nachdem Sie die korrekten Werte ermittelt haben, können Sie die entsprechende Zeile in exakt der gleichen Form in die Datei `/boot.config` aufnehmen. In der Voreinstellung verwendet FreeBSD den Inhalt dieser Datei als Standardantwort am `boot:`-Prompt.

3. Nach der Installation beginnt das System zu booten, der Bootmanager zeigt im Bootmenü aber immer nur `F?` an und das System startet nicht.

Sie haben bei der FreeBSD-Installation eine falsche Plattengeometrie angegeben. Starten Sie den Partitionseditor neu und geben Sie die korrekte Plattengeometrie an. Danach installieren Sie FreeBSD erneut (diesmal mit der korrekten Plattengeometrie).

Ist es Ihnen nicht möglich, die korrekte Plattengeometrie herauszufinden, hilft Ihnen vielleicht der folgende Tipp weiter: Legen Sie eine kleine MS-DOS-Partition am Beginn Ihrer Bootplatte an und installieren Sie anschließend FreeBSD auf diese Platte. Das FreeBSD-Installationsprogramm wird die MS-DOS-Partition erkennen und ist dadurch normalerweise in der Lage, die korrekte Plattenkonfiguration automatisch zu erkennen.

Die Vorgangsweise im folgenden Tipp wird zwar nicht länger empfohlen, soll aber trotzdem dokumentiert werden:

> Wenn Sie ein reines FreeBSD-System aufsetzen wollen (als Server oder als Workstation) und daher nie auf Kompatibilitält zu MS-DOS, Linux oder anderen Betriebssystemen angewiesen sein werden, haben Sie auch die Möglichkeit, die komplette Platte (durch die Wahl von `A` im Partitionseditor) für FreeBSD zu verwenden. Danach wird FreeBSD die komplette Platte vom ersten bis zum letzten Sektor verwenden und die tatsächliche Plattengeometrie ignorieren. Danach ist es allerdings nicht mehr möglich, ein anderes Betriebssystem auf die gleiche Platte zu installieren (ohne auch FreeBSD neu zu installieren).

4. FreeBSD erkennt meine ed(4)-Netzwerkkarte. Trotzdem erhalte ich weiterhein Timeout-Meldungen für dieses Gerät.

Ihre Karte verwendet wahrscheinlich einen anderen IRQ als den, der in der Datei `/boot/device.hints` angegeben wurde. Der ed(4)-Treiber verwendet in der Voreinstellung keine “Soft”-Konfiguration (also Werte, die durch EZSETUP unter MS-DOS eingeben wurden). Sie können dies allerdings erzwingen, indem Sie die Option `-1` in den “Hints” für dieses Gerät angeben.

Entweder verändern Sie die Jumper-Konfiguration der Karte (und, falls notwendig, die Kerneleinstellungen). Oder Sie geben den IRQ als `-1` an, indem Sie `hint.ed.0.irq="-1"` eingeben. Dadurch wird der Kernel angewiesen, die “Soft”-Konfiguration zu verwenden.

Prüfen Sie auch, ob Ihre Karte nicht etwa IRQ 9 verwendet, da dieser mit IRQ 2 geteilt wird. Diese Einstellung verursacht häufig Probleme (insbesondere dann, wenn IRQ 2 durch eine VGA-Grafikkarte belegt ist!). Wenn irgend möglich, sollten Sie daher IRQ 2 oder 9 nicht verwenden.

5. Wenn ich **sysinstall** aus einen X-Terminal starte, ist die gelbe Schritt auf dem grauen Hintergrund nur schwer zu erkennen. Gibt es eine Möglichkeit, den Kontrast für dieses Programm zu erhöhen?

Haben Sie X11 bereits installiert und die von **sysinstall** verwendeten Farben bereiten Ihnen beim Lesen von Text Probleme (wenn Sie ein X-Terminal verwenden), sollten Sie die Zeile `XTerm*color7: #c0c0c0` in die Datei `~/.Xdefaults` aufnehmen. Dadurch wird der Hintergrund in einem dunkleren Grauton dargestellt.

2.12. Anspruchsvollere Installationen

Beigetragen von Valentino Vaschetto. Aktualisiert von Marc Fonvieille.

Dieser Abschnitt beschreibt die Installation von FreeBSD in besonderen Situationen.

2.12.1. FreeBSD auf einem System ohne Monitor oder Tastatur installieren

Diese Methode wird als “headless install” (kopflose Installation) bezeichnet, da die Maschine, auf die FreeBSD installiert werden soll, entweder keinen Monitor angeschlossen hat oder über keine VGA-Karte verfügt. Wie kann FreeBSD dennoch installiert werden? Eben mithilfe einer seriellen Konsole. Im Wesentlichen ist eine serielle Konsole eine andere Maschine, die Ein- und Ausgaben für eine andere Maschine bereitstellt. Um über eine serielle Konsole zu installieren, erstellen Sie zunächst (wie in Abschnitt 2.3.7 beschrieben) einen bootbaren USB-Stick oder laden Sie das passende CD-ISO-Abbild herunter.

Um von diesen Medien in eine serielle Konsole booten zu können, müssen Sie die folgenden Schritte durchführen (bei Verwendung einer Boot-CD kann der erste Schritt entfallen):

1. Den USB-Stick für eine serielle Konsole anpassen

   Wenn Sie ein System mit den frisch erstellten USB-Stick starten, läuft der normale FreeBSD-Installationsprozess an. Diese Installation soll aber über die serielle Konsole gesteuert werden. Daher müssen Sie den USB-Stick mit dem Befehl mount(8) in den Verzeichnisbaum einhängen:

   ```
   # mount /dev/da0a /mnt
   ```

   Anmerkung: Passen Sie den Mountpunkt und die Gerätedatei falls nötig an Ihre Gegebenheiten an.

   Nachdem Sie den USB-Stick eingehängt haben, müssen Sie ihn rekonfigurieren, damit er in eine serielle Konsole startet. Dazu nehmen Sie in die Datei `loader.conf` des USB-Sticks eine Zeile auf, die die serielle Konsole als Systemkonsole festlegt:

   ```
   # echo 'console="comconsole"' >> /mnt/boot/loader.conf
   ```

   Damit ist Ihr USB-Stick für die Installation vorbereitet. Sie können ihn daher wieder aus dem Dateisystem aushängen:

   ```
   # umount /mnt
   ```

   Entfernen Sie nun den USB-Stick und machen Sie direkt mit Schritt 3 weiter.

2. Die Installations-CD für eine serielle Konsole anpassen

   Wenn Sie von dem soeben heruntergeladenen CD-ISO-Abbild (siehe Abschnitt 2.13.1) starten, gelangen Sie in den normalen Installationsmodus von FreeBSD. Da wir aber in eine serielle Konsole booten wollen, muss das CD-Image extrahiert, modifiziert und neu erzeugt werden, bevor Sie es auf eine CD-R brennen.

   Entpacken Sie alle Dateien des CD-ISO-Abbilds (beispielsweise `FreeBSD-9.1-RELEASE-i386-disc1.iso`) auf dem System, auf das Sie das Abbild heruntergeladen haben unter Verwendung von tar(1):

   ```
   # mkdir /path/to/headless-iso
   # tar -C /path/to/headless-iso -pxvf FreeBSD-9.1-RELEASE-i386-disc1.iso
   ```

Nun müssen Sie das entpackte ISO-Abbild rekonfigurieren, damit es künftig in eine serielle Konsole startet. Dazu nehmen Sie in die Datei `loader.conf` des entpackten ISO-Abbild eine Zeile auf, die die serielle Konsole als Systemkonsole festlegt:

```
# echo 'console="comconsole"' >> /path/to/headless-iso/boot/loader.conf
```

Damit ist der Dateibaum des entpackten ISO-Abbilds für die Installation vorbereitet und Sie können über den Befehl mkisofs(8) (das Sie über den Port `sysutils/cdrtools` installieren können) ein neues CD-ISO-Abbild erzeugen:

```
# mkisofs -v -b boot/cdboot -no-emul-boot -r -J -V "Headless_install" \
     -o Headless-FreeBSD-9.1-RELEASE-i386-disc1.iso /path/to/headless-iso
```

Dieses rekonfigurierte ISO-Abbild brennen Sie nun mit dem Brennprogramm Ihrer Wahl auf eine CD-R.

3. Das Nullmodemkabel anschließen

   Sie müssen beide Maschinen mit einem Nullmodemkabel verbinden. Schließen Sie das Nullmodemkabel an die seriellen Schnittstellen beider Maschinen an. *Sie können kein direktes serielles Kabel verwenden*, Nullmodemkabel besitzen gekreuzte Leitungen.

4. Die Installation starten

   Sie können die Installation jetzt starten. Stöpseln Sie den vorbereiteten USB-Stick ein und starten Sie Ihren Computer. Alternativ starten Sie Ihren Computer und legen die vorbereitete Installations-CD ein.

5. Die Verbindung mit der zur installierenden Maschine herstellen

   Mit dem Kommando cu(1) verbinden Sie sich mit der zu installierenden Maschine:

   ```
   # cu -l /dev/cuau0
   ```

   Unter FreeBSD 7.X verwenden Sie hingegen den folgenden Befehl:

   ```
   # cu -l /dev/cuad0
   ```

Fertig! Über die `cu`-Sitzung können Sie nun die zu installierende Maschine steuern. Der Kernel wird automatisch geladen und Sie können anschließend den Terminaltyp festlegen. Wählen Sie die `FreeBSD color console` aus und fahren wie gewohnt mit der Installation fort.

2.13. Eigene Installationsmedien herstellen

> **Anmerkung:** Im Folgenden ist mit “Installations-CD” eine CD-ROM oder DVD gemeint, die Sie gekauft oder selbst hergestellt haben.

Oft müssen Sie eigene Installationsmedien erzeugen. Dies können physische Medien wie Bänder sein oder Installationsquellen sein, aus denen **sysinstall** Dateien herunterlädt, beispielsweise ein lokaler FTP-Server oder eine MS-DOS-Partition.

Beispiele:

- Im lokalen Netzwerk befinden sich viele Maschinen, Sie besitzen allerdings nur eine Installations-CD. Den Inhalt der Installations-CD wollen Sie auf einem lokalem FTP-Server bereitstellen. Zur Installation wird der lokale FTP-Server anstelle eines Internet-Servers benutzt.
- Sie haben eine Installations-CD, allerdings erkennt FreeBSD im Gegensatz zu MS-DOS/Windows das CD- oder DVD-Laufwerk nicht. Sie können die Installationsdateien auf eine MS-DOS-Partition desselben Rechners kopieren und FreeBSD von der MS-DOS-Partition installieren.
- Der Rechner, auf dem Sie FreeBSD installieren wollen, besitzt kein CD- oder DVD-Laufwerk. Ein anderer Rechner, zu dem eine serielle oder parallele Verbindung besteht, besitzt allerdings ein CD- oder DVD-Laufwerk.
- Sie wollen ein Band erzeugen, mit dem Sie FreeBSD installieren können.

2.13.1. Eine Installations-CD-ROM erzeugen

Mit jeder Release stellt das FreeBSD-Project für jede unterstützte Architektur mindestens zwei CD-Abbilder (“ISO-Images”) zur Verfügung. Wenn Sie einen CD-Brenner besitzen, können Sie diese Abbilder brennen und damit FreeBSD installieren. Wenn Sie einen CD-Brenner besitzen und über eine gute Internet-Verbindung verfügen, ist das die preiswerteste Art, FreeBSD zu installieren.

1. Das richtige Abbild herunterladen

   Die ISO-Abbilder für jedes Releases können Sie von `ftp://ftp.FreeBSD.org/pub/FreeBSD/ISO-IMAGES-`*`arch`*`/`*`version`* oder einem nahe gelegenen Spiegel herunterladen. Ersetzen Sie *`arch`* und *`version`* durch passende Werte.

   Das Verzeichnis enthält die folgenden Abbilder:

   Tabelle 2-4. FreeBSD 7.*X* und 8.*X* ISO-Abbilder

Dateiname	Inhalt
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-bootonly.iso`	Enthält alles, was Sie benötigen, um um den FreeBSD-Kernel zu laden und das Installationsprogramm zu starten. Die zu installierenden Dateien müssen allerdings über FTP oder eine andere geeignete Quelle bezogen werden, da sie in diesem Abbild nicht enthalten sind.
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-dvd1.iso.gz`	Dieses DVD-Abbild enthält alle zur Installation von FreeBSD nötigen Dateien, eine Auswahl an Paketen Dritter sowie die Dokumentation. Zusätzlich ermöglicht es Ihnen dieses Abbild, einen “livefs”-basierten Rettungsmodus zu starten.
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-memstick.img`	Dieses Abbild kann auf einen USB-Stick geschrieben werden. Dieser kann danach als Installationsmedium verwendet werden (wenn Ihr System dies unterstützt). Zusätzlich ermöglicht es Ihnen dieses Abbild, einen “livefs”-basierten Rettungsmodus zu starten. Die FreeBSD-Dokumentation ist ebenfalls enthalten, aber keine Pakete Dritter. Dieses Abbild ist erst ab FreeBSD 8.0 verfügbar.

Dateiname	Inhalt
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-disc1.iso`	Dieses CD-Abbild enthält alle für die Installation von FreeBSD nötigen Dateien sowie die Dokumentation. Es sind allerdings keine Pakete Dritter enthalten.
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-disc2.iso`	So viele Pakete Dritter, wie auf dem Installationsmedium Platz hatten. Dieses Abbild ist für FreeBSD 8.*X* nicht mehr verfügbar.
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-disc3.iso`	Ein weiteres Abbild mit so vielen Paketen Dritter, wie auf dem Installationsmedium Platz hatten. Dieses Abbild ist für FreeBSD 8.*X* nicht mehr verfügbar.
`version``-RELEASE-`*`arch`*`-docs.iso`	Die FreeBSD-Dokumentation.
`FreeBSD-`*`version`*`-RELEASE-`*`arch`*`-livefs.iso`	Dieses Abbild enthät einen “livefs”-basierten Rettungsmodus. Eine Installation von FreeBSD ist mit diesem Abbild allerdings nicht möglich.

Anmerkung: Die Abbilder für FreeBSD 7.*X*-Releases vor FreeBSD 7.3 sowie für FreeBSD 8.0 wurden noch unterschiedlich benannt. Bei Ihnen fehlt die Bezeichnung `FreeBSD-` am Anfang des Abbildnamens.

Sie benötigen nur eines der beiden Abbilder `bootonly` oder `disc1`. Laden Sie bitte nicht beide Abbilder herunter, das `disc1`-Abbild enthält alles, was das `bootonly`-Abbild enthält.

Benutzen Sie das `bootonly`-Abbild, wenn Sie eine preiswerte Internet-Anbindung besitzen. Mit diesem Abbild können Sie FreeBSD installieren. Software Dritter können Sie anschließend mithilfe des Ports-Systems (Kapitel 5) herunterladen.

Benutzen Sie das `dvd1`-Abbild, wenn Sie FreeBSD installieren wollen und das Installationsmedium eine angemessene Auswahl an Software Dritter enthalten soll.

Die zusätzlichen Abbilder sind nützlich, aber nicht notwendig, insbesondere wenn Sie eine schnelle Internet-Verbindung besitzen.

2. Die CDs brennen

   Sie müssen die Abbilder auf eine CD brennen. Das Brennen von CDs unter FreeBSD wird in Abschnitt 19.6 erläutert (sehen Sie sich insbesondere Abschnitt 19.6.3 und Abschnitt 19.6.4 an).

   Wenn Sie die CDs unter einem anderen Betriebssystem erstellen, benutzen Sie die entsprechenden Werkzeuge des Betriebssystems. Die Abbilder sind Standard-ISO-Abbilder und können von vielen Brennprogrammen verarbeitet werden.

   Anmerkung: Wenn Sie eine angepasste Version von FreeBSD erstellen wollen, sollten Sie den Release Engineering Article (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/releng) lesen.

2.13.2. Einen lokalen FTP-Server einrichten

Die Dateien auf der Installations-CD sind genauso angeordnet wie auf den FreeBSD-FTP-Servern. Daher ist es einfach, einen lokalen FTP-Server für die FreeBSD-Installation über ein Netzwerk einzurichten.

1. Hängen Sie auf dem FTP-Server die CD-ROM in das Verzeichnis `/cdrom` ein:

   ```
   # mount /cdrom
   ```

2. Legen Sie ein Konto für Anonymous-FTP an. Dazu editieren Sie die Datei `/etc/passwd` mit dem Kommando vipw(8) und fügen die nachstehende Zeile hinzu:

   ```
   ftp:*:99:99::0:0:FTP:/cdrom:/nonexistent
   ```

3. Stellen Sie sicher, dass der FTP-Dienst in der Datei `/etc/inetd.conf` aktiviert ist.

Jeder, der Ihren Rechner über das Netzwerk erreicht, kann nun FreeBSD über FTP installieren. In **sysinstall** wird dazu FTP als Installationsmedium wählt. Der FTP-Server wird durch die Auswahl Other (andere als die vorgegebenen Server) und anschließende Eingabe von **`ftp://`*`Ihr Rechner`*** festgelegt.

> **Anmerkung:** Wenn die Version der für die FTP-Installation Ihrer Clients verwendeten Bootmedien (normalerweise Disketten) nicht exakt der von Ihnen auf Ihrem lokalen FTP-Server angebotenen Version entspricht, ist **sysinstall** nicht in der Lage, die Installation abzuschließen. Sind die Versionsnummern unterschiedlich, können Sie durch das Aufrufen des Punktes Options **sysinstall** dazu zwingen, die Installation dennoch abzuschließen. Dazu setzen Sie den Namen der Distribution auf any.

> **Warnung:** Diese Vorgehensweise ist in Ihrem lokalen Netzwerk, das durch eine Firewall geschützt ist, völlig in Ordnung. Wenn Sie FTP für Rechner auf dem Internet (und nicht für lokale Rechner) anbieten, zieht Ihr Server die Aufmerksamkeit von Crackern und andere Unannehmlichkeiten auf sich. Achten Sie in diesem Fall darauf, dass Sie gute Sicherheitsverfahren anwenden.

2.13.3. Installationsdisketten erstellen

Wenn Sie, was wir *nicht* empfehlen, von Disketten installieren müssen, weil Disketten das einzig unterstützte Installationsmedium sind oder Sie es sich einfach schwer machen wollen, müssen Sie zunächst einige Disketten vorbereiten.

Sie müssen mindestens den Inhalt des Verzeichnisses `base` auf 1.44 MB Disketten kopieren. Wenn Sie die Disketten unter MS-DOS erstellen, *müssen* Sie die Disketten mit dem MS-DOS-Kommando `FORMAT` formatieren. Unter Windows können Sie Disketten mithilfe des Explorers formatieren (klicken Sie mit der rechten Maustaste auf das `A:`-Laufwerk und wählen Sie Format aus).

Vertrauen Sie vorformatierten Disketten nicht; formatieren Sie die Disketten zur Sicherheit immer selbst. In der Vergangenheit waren vorformatierte Disketten der Verursacher vieler Probleme.

Falls Sie die Disketten auf einer FreeBSD-Maschine erstellen, sollten Sie immer noch formatieren. Allerdings brauchen Sie kein MS-DOS-Dateisystem auf den Disketten anzulegen. Mit den Kommandos `bsdlabel` und `newfs` können Sie das Dateisystem UFS verwenden, wie im nachstehenden Beispiel für 3.5" 1.44 MB Disketten gezeigt:

```
# fdformat -f 1440 fd0.1440
```

```
# bsdlabel -w fd0.1440 floppy3
# newfs -t 2 -u 18 -l 1 -i 65536 /dev/fd0
```

Anschließend können Sie die Disketten wie ein normales Dateisystem einhängen und beschreiben.

Nachdem Sie die Disketten formatiert haben, kopieren Sie die Dateien der Distribution auf die Disketten. Die Dateien der Distribution sind in Stücke geteilt, sodass fünf Dateien auf eine 1.44 MB Diskette passen. Kopieren Sie die gewünschten Distribution auf Disketten, wobei Sie so viele Dateien wie möglich auf eine Diskette kopieren. Jede Distribution wird auf der Diskette in einem eigenen Verzeichnis abgelegt, beispielsweise `a:\base\base.aa`, `a:\base\base.ab` und so weiter.

> **Wichtig:** Die Datei `base.inf` muss unbedingt auf die erste Diskette des `base`-Diskettensatzes kopiert werden, damit das Installationsprogramm feststellen kann, wie viele Disketten geladen werden müssen, um die Distribution wieder zusammenzusetzen.

Im Installationsprozess wählen Sie als Installationsmedium Floppy aus. Folgen Sie dann den gegebenen Anweisungen.

2.13.4. Von einer MS-DOS-Partition installieren

Um eine Installation von einer MS-DOS-Partition vorzubereiten, kopieren Sie Dateien der Distributionen in das Verzeichnis `freebsd` direkt unterhalb des Wurzelverzeichnisses (zum Beispiel `c:\freebsd`). In diesem Verzeichnis muss sich dieselbe Verzeichnisstruktur wie auf dem Installationsmedium befinden. Wenn Sie die Dateien von einer Installations-CD kopieren, empfehlen wir den MS-DOS-Befehl `xcopy`. Das nachstehende Beispiel bereitet eine minimale Installation von FreeBSD vor:

```
C:\> md c:\freebsd
C:\> xcopy e:\bin c:\freebsd\bin\ /s
C:\> xcopy e:\manpages c:\freebsd\manpages\ /s
```

Im Beispiel wurde angenommen, dass auf Laufwerk `C:` ausreichend Platz vorhanden ist und die CD-ROM Laufwerk `E:` ist.

Wenn Sie kein CD-Laufwerk besitzen, können Sie die Distributionen von ftp.FreeBSD.org (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/) herunterladen. Jede Distribution liegt in einem eigenen Verzeichnis. Beispielsweise liegt die Base-Distribution im Verzeichnis 9.1/base/ (ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/9.1-RELEASE/base/).

Kopieren Sie jede Distribution, die Sie von einer MS-DOS-Partition installieren wollen (und für die Platz ist) in das Verzeichnis `c:\freebsd`. Für eine minimale Installation benötigen Sie nur die Base-Distribution.

2.13.5. Ein Installationsband erstellen

Falls Sie nicht über FTP oder von einer CD-ROM installieren können, ist die Installation von Band wahrscheinlich die einfachste Methode. Das Installationsprogramm erwartet, dass sich die Distributionen im `tar`-Format auf dem Band befinden. Von den Distributions-Dateien erstellen Sie das Installationsband einfach mit dem Kommando `tar`:

```
# cd /freebsd/distdir
# tar cvf /dev/rwt0 dist1 ... dist2
```

Stellen Sie während der Installation sicher, dass Sie über genügend freien Platz in einem temporären Verzeichnis (das Sie festlegen können) verfügen. Das temporäre Verzeichnis muss den *gesamten* Inhalt des Bands aufnehmen können. Da auf Bänder nicht wahlfrei zugegriffen werden kann, benötigt diese Installationsmethode temporär sehr viel Platz.

> **Anmerkung:** Das Band muss sich vor dem Neustart mit der Startdiskette im Laufwerk befinden. Ansonsten wird das Band während der Geräteerkennung vielleicht nicht erkannt.

2.13.6. Eine Netzwerkinstallation vorbereiten

Sie können drei Verbindungsarten für eine Netzwerkinstallation benutzen: Eine Ethernet-Verbindung, eine serielle Verbindung (PPP), oder eine parallele Verbindung (PLIP, Laplink-Kabel).

Die schnellste Netzwerkinstallation ist natürlich mit einer Netzwerkkarte möglich. FreeBSD unterstützt die meisten der üblichen Netzwerkkarten. Eine Liste der unterstützten Netzwerkkarten ist in den Hardware-Notes jedes Releases enthalten. Wenn Sie eine unterstützte PCMCIA-Netzwerkkarte benutzen, stellen Sie sicher, dass die Karte eingesteckt ist, *bevor* der Laptop eingeschaltet wird. Leider unterstützt FreeBSD das Einstecken von PCMCIA-Karten während der Installation noch nicht.

Für eine Netzwerkinstallation müssen Sie Ihre IP-Adresse, die Netzwerkmaske und den Namen Ihres Rechner kennen. Wenn Sie über eine PPP-Verbindung installieren und keine feste IP-Adresse besitzen, braucht Sie der vorgehende Satz nicht zu beunruhigen. Sie können eine IP-Adresse dynamisch von Ihrem ISP beziehen. Fragen Sie Ihren Systemadministrator nach den richtigen Netzwerkeinstellungen. Wenn Sie andere Rechner über Namen anstatt über IP-Adressen erreichen wollen, brauchen Sie zudem einen Nameserver und möglicherweise die Adresse eines Gateways (mit PPP ist das die Adresse des ISPs), über den Sie den Nameserver erreichen. Wenn Sie von einem FTP-Server über einen HTTP-Proxy installieren wollen, benötigen Sie außerdem noch die Adresse des Proxy-Servers. Wenn Sie nicht alle oder zumindest die meisten der benötigten Daten kennen, sollten Sie wirklich *vor* der Installation mit Ihrem Systemadministrator oder ISP reden!

Wenn Sie ein Modem benutzen, ist PPP ziemlich sicher die einzige Wahl. Stellen Sie sicher, dass Sie die Daten Ihres Service Providers bereitliegen haben, da Sie während der Installation die Daten früh benötigen.

Wenn Sie PAP oder CHAP benutzen, um sich mit Ihrem ISP zu verbinden (wenn Sie unter Windows kein Skript benötigen, um die Verbindung herzustellen), brauchen Sie an der **ppp**-Eingabeaufforderung nur das Kommando `dial` abzusetzen. Ansonsten müssen Sie sich mit Modem-spezifischen AT-Kommandos bei Ihrem ISP einwählen (PPP stellt nur einen einfachen Terminal-Emulator zur Verfügung). Weiteres über PPP erfahren Sie im Abschnitt User-PPP des Handbuchs und im PPP-Abschnitt (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/faq/ppp.html) der FAQ. Bei Problemen können Sie mit dem Kommando `set log local` Meldungen auf den Bildschirm umleiten.

Wenn eine feste Verbindung zu einer anderen FreeBSD-Maschine besteht, sollten Sie ein paralleles Laplink-Kabel in Betracht ziehen. Über eine parallele Verbindung sind höhrere Geschwindigkeiten als über eine serielle Verbindung (typischerweise bis zu 50 kByte/s) möglich. Daher ist die Installation über eine parallele Verbindung schneller als eine Installation über eine serielle Verbindung.

2.13.6.1. Eine NFS-Installation vorbereiten

Eine NFS-Installation ist unkompliziert. Kopieren Sie einfach die Distributionen auf einen NFS-Server und wählen Sie NFS als Installationsmedium aus.

Wenn der NFS-Server nur Verbindungen über privilegierte Ports (Ports kleiner 1024) annimmt, setzen Sie vor der Installation die Option `NFS Secure` im Menü Options.

Wenn Sie eine schlechte Netzwerkkarte besitzen, die sehr langsam ist, wählen Sie die Option `NFS Slow`.

Damit die NFS-Installation funktioniert, muss der NFS-Server auch Unterverzeichnisse von exportierten Verzeichnissen zum Einhängen freigeben. Wenn beispielsweise die Distribution von FreeBSD 9.1 unter `ziggy:/usr/archive/stuff/FreeBSD` liegt, muss der Rechner `ziggy` erlauben, das Verzeichnis `/usr/archive/stuff/FreeBSD` einzuhängen. Es reicht nicht, dass `ziggy` erlaubt das Verzeichnis `/usr` oder `/usr/archive/stuff` einzuhängen.

Unter FreeBSD werden diese Freigaben in der Datei `/etc/exports` mit der Option `-alldirs` eingestellt. Die nötigen Einstellungen können auf einem anderen NFS-Server unterschiedlich sein. Wenn Sie vom NFS-Server die Fehlermeldung `permission denied` erhalten, dann haben Sie wahrscheinlich die Freigaben nicht richtig konfiguriert.

Kapitel 3. FreeBSD 9.*x* (und neuer) installieren

Restructured, reorganized, and parts rewritten by Jim Mock. The sysinstall walkthrough, screenshots, and general copy by Randy Pratt. Updated for bsdinstall by Gavin Atkinson und Warren Block. Übersetzt von Benedict Reuschling.

3.1. Übersicht

FreeBSD enthält ein text-basiertes, einfach zu verwendendes Installationsprogramm. FreeBSD 9.0-RELEASE und neuer verwendet ein Installationsprogramm genannt **bsdinstall**, während Versionen vor FreeBSD 9.0-RELEASE stattdessen **sysinstall** zur Installation einsetzten. Dieses Kapitel beschreibt die Verwendung von **bsdinstall**. Der Einsatz von **sysinstall** wird in Kapitel 2 behandelt.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- wie man FreeBSD Installationsmedien erstellt.
- wie FreeBSD Festplatten unterteilt und darauf verweist.
- wie man **bsdinstall** startet.
- welche Fragen Sie von **bsdinstall** gestellt bekommen, was sie bedeuten und und wie man diese beantwortet.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Die Liste von unterstützter Hardware lesen, die mit Ihrer zu installierenden Version von FreeBSD ausgeliefert wird, um sicherzustellen, dass Ihre Hardware auch unterstützt wird.

> **Anmerkung:** Generell wurden diese Installationsanweisungen für Rechner der i386 (“PC-kompatibel”) Architektur verfasst. An Stellen, an denen sich die Anweisungen speziell auf eine andere Plattform beziehen, wird darauf hingewiesen. Es mag kleinere Unterschiede geben zwischen dem Installationsprogramm und dem, was hier beschrieben steht. Sie sollten daher dieses Kapitel als eine Art Wegweiser und keine exakte Anleitung betrachten.

3.2. Hardware-Anforderungen

3.2.1. Minimalkonfiguration

Die Minimalkonfiguration zur Installation von FreeBSD variiert mit der Version von FreeBSD und der Hardwarearchitektur.

Eine Zusammenfassung dieser Informationen wird in den folgenden Abschnitten gegeben. Abhängig von der Installationsmethode, die Sie verwenden, um FreeBSD zu installieren, werden Sie unter Umständen ein unterstütztes CD-ROM-Laufwerk benötigen und in manchen Fällen eine Netzwerkkarte. Dies wird im Abschnitt Abschnitt 3.3.5 genauer betrachtet.

3.2.1.1. FreeBSD/i386

FreeBSD/i386 benötigt einen 486er oder einen schnelleren Prozessor und mindestens 64 MB RAM. Es sollte mindestens 1.1 GB freier Festplattenspeicher für die Installation zur Verfügung stehen.

> **Anmerkung:** Auf alten Rechnern hat die Aufrüstung von RAM und dem Festplattenplatz normalerweise einen höheren geschwindigkeitssteigernden Effekt als einen schnelleren Prozessor einzubauen.

3.2.1.2. FreeBSD/amd64

Es gibt zwei Klassen von Prozessoren, die in der Lage sind, auf FreeBSD/amd64 zu laufen. Die erste Klasse sind AMD64-Prozessoren, was sowohl AMD Athlon™64, AMD Athlon64-FX, AMD Opteron™ oder bessere Prozessoren beinhaltet.

Die zweite Klasse von Prozessoren, die FreeBSD/amd64 benutzen kann, besteht aus der Intel EM64T-Architektur. Beispiele dieser Prozessoren beinhalten die Intel Core 2 Duo, Quad, Extreme Prozessorfamilien, die Intel Xeon 3000, 5000, und 7000 Reihe von Prozessoren, sowie die Intel Core i3, i5 and i7 Prozessoren.

Sollten Sie einen Rechner basierend auf der nVidia nForce3 Pro-150 besitzen, *müssen* Sie im BIOS das IO APIC deaktivieren. Falls Sie keine solche Option zum deaktivieren besitzen, werden Sie wahrscheinlich ACPI deaktivieren müssen. Der Pro-150 Chipsatz enthält Fehler, für die wir noch keine Abhilfe gefunden haben.

3.2.1.3. FreeBSD/powerpc Apple® Macintosh®

Alle neuen Apple® Macintosh® Systeme mit eingebautem USB werden unterstützt. SMP wird auf Maschinen mit mehreren CPUs unterstützt.

Ein 32-bit Kernel kann nur die ersten 2 GB des Hauptspeichers verwenden. FireWire® wird auf den blauen und weissen PowerMac G3s nicht unterstützt.

3.2.1.4. FreeBSD/sparc64

Systeme, die von FreeBSD/sparc64 unterstützt werden, sind auf der FreeBSD/sparc64 (http://www.freebsd.org/platforms/sparc.html)-Projektseite aufgelistet.

Eine dedizierte Platte wird für FreeBSD/sparc64 benötigt. Es ist nicht möglich, eine Platte mit einem anderen Betriebssystem zur gleichen Zeit zu teilen.

3.2.2. Unterstützte Hardware

Hardwarearchitekturen und von FreeBSD unterstützte Geräte werden in der Datei mit Hardware Notes aufgelistet. Normalerweise heisst diese Datei `HARDWARE.TXT` und befindet sich im Wurzelverzeichnis des Veröffentlichungsmediums. Kopien dieser unterstützten Hardwareliste ist ebenfalls auf der Seite Release Information (http://www.FreeBSD.org/releases/index.html) der FreeBSD Webseite abrufbar.

3.3. Vor der Installation

3.3.1. Sichern Sie Ihre Daten

Sichern Sie alle wichtigen Daten auf dem Zielcomputer, auf dem FreeBSD installiert werden soll. überprüfen Sie diese Sicherungen, bevor Sie fortfahren. Die FreeBSD Installation wird Sie vor Änderungen an den Platten danach fragen, jedoch kann dies nicht mehr rückgängig gemacht werden, sobald der Prozess gestartet wurde.

3.3.2. Den Installationsort von FreeBSD festlegen

Falls FreeBSD das einzige installierte Betriebssystem sein wird und die gesamte Platte dazu verwenden kann, kann der Rest dieses Abschnitts übersprungen werden. Sollten Sie allerdings die Platte mit anderen Betriebssystemen teilen, ist ein Verständnis des Plattenlayouts hilfreich für die Installation.

3.3.2.1. Festplattenlayout für FreeBSD/i386 und FreeBSD/amd64

Festplatten können in mehrere verschiedene Bereiche aufgeteilt werden. Diese Bereiche werden *Partitionen* genannt.

Es gibt zwei Arten, eine Festplatte in mehrere Partitionen einzuteilen. Traditionell enthält ein *Master Boot Record* (MBR) eine Partitionstabelle, welche bis zu vier *primäre Partitionen* aufnehmen kann (aus historischen Gründen werden diese primären Partitionen in FreeBSD *slices* genannt). Eine Begrenzung von nur vier Partitionen ist für grosse Platten sehr beschränkt, so dass eine dieser primären Partitionen als *erweiterte Partition* eingesetzt wird. Mehrere *logische Partitionen* können dann innerhalb der erweiterten Partition angelegt werden. Dies mag etwas unhandlich erscheinen und das ist auch der Fall.

Die *GUID-Partitionstabelle* (GPT) ist eine neuere und einfachere Methode zur Partition einer Festplatte. GPT ist weitaus flexibler als die traditionelle MBR-Partitionstabelle. Geläufige GPT-Implementierungen erlauben bis zu 128 Partitionen pro Platte, was die Notwendigkeit von umständlichen Behelfen wie logische Partitionen eliminiert.

> **Warnung:** Manche älteren Betriebssysteme wie Windows XP sind mit dem GPT-Partitionsschema nicht kompatibel. Wenn sich FreeBSD die Platte mit einem solchen Betriebssystem teilen soll, werden MBR Partitionen benötigt.

FreeBSDs Standard-Bootloader benötigt entweder eine primäre oder eine GPT-Partition (lesen Sie dazu Kapitel 13 für weitere Informationen zum FreeBSD Bootvorgang). Wenn alle der primären oder GPT-Partitionen bereits in Verwendung sind, muss eine davon für FreeBSD zur Verfügung gestellt werden.

Eine Minimalinstallation von FreeBSD braucht ungefähr 1 GB Plattenplatz. Dies ist jedoch eine *sehr* minimale Installation, die fast gar keinen freien Speicherplatz übrig lässt. Eine etwas realistischere Minimalangabe sind 3 GB ohne eine graphische Umgebung und 5 GB oder mehr, falls eine graphische Benutzeroberfläche verwendet werden soll. Anwendungen von Drittanbietern benötigt sogar noch mehr Platz.

Eine Vielzahl freier und kommerzieller Werkzeuge zur Veränderung der Partitionsgrössen (http://en.wikipedia.org/wiki/List_of_disk_partitioning_software) sind verfügbar. GParted Live (http://gparted.sourceforge.net/livecd.php) ist eine freie Live-CD, die den GParted-Partitionseditor enthält. GParted ist auch in einer Vielzahl von anderen Linux Live-CD Distributionen enthalten.

> **Warnung:** Anwendungen zur Festplattenpartition kann Daten zerstören. Erstellen Sie eine Vollsicherung und überprüfen Sie deren Integrität bevor Sie die Partitionen auf der Platte verändern.

Die Veränderung der Grösse von Microsoft Vista-Partitionen kann schwierig sein. Eine Vista Installations-CD-ROM kann hilfreich sein, wenn eine solche Aktion versucht wird.

Beispiel 3-1. Eine existierende Partition verändern

Ein Windows-Computer besitzt eine einzelne 40 GB Platte, die in zwei 20 GB Partitionen aufgeteilt wurde. Windows nennt diese `C:` und `D:`. Die `C:` Partition enthält 10 GB und the `D:` Partition 5 GB an Daten.

Durch kopieren der Daten von `D:` nach `C:` macht die zweite Partition frei, so dass FreeBSD sie benutzen kann.

Beispiel 3-2. Verkleinern einer bestehenden Partition

Ein Windows-Computer besitzt eine einzelne 40 GB Platte und eine grosse Partition, welche die gesamte Platte einnimmt. Windows zeigt diese 40 GB Partition als einzelne `C:` Partition. 15 GB Plattenplatz wird verwendet. Das Ziel ist, für Windows eine 20 GB Partition einzurichten und eine weitere 20 GB-Partition für FreeBSD bereitzustellen.

Es gibt zwei Wege, dieses Ziel zu erreichen.

1. Sichern Sie Ihre Windows-Daten. Installieren Sie dann Windows neu, indem Sie eine 20 GB-Partition während der Installation anlegen.
2. Verwenden Sie ein Werkzeug zur Veränderung einer Partition wie **GParted**, um die Windows-Partition zu verkleinern und eine neue Partition im freigewordenen Plattenplatz für FreeBSD anzulegen.

Festplattenpartitionen, die unterschiedliche Betriebssysteme enthalten, ermöglichen es, jeweils eines dieser Systeme zu verwenden. Eine andere Methode, die es erlaubt, mehrere Betriebssysteme gleichzeitig einzusetzen, wird in Kapitel 23 behandelt.

3.3.3. Netzwerkparameter ermitteln

Manche FreeBSD Installationsarten benötigen eine Netzwerkverbindung, um Dateien herunter zu laden. Um zu einem Ethernet-Netzwerk (bzw. Kabel oder DSL-Modem mit einem Ethernet-Anschluss) eine Verbindung herzustellen, wird das Installationsprogramm bestimmte Information zum Netzwerk abfragen.

DHCP wird allgemein verwendet, um automatisch Netzwerkeinstellungen vorzunehmen. Falls DHCP nicht verfügbar ist, müssen diese Netzwerkeinstellungen vom lokalen Netzwerkadministrator oder Provider erfragt werden:

Informationen zum Netzwerk

1. IP-Adresse
2. Subnetz-Maske
3. Default-Router IP-Adresse

4. Domänenname des lokalen Netzwerks
5. DNS-Server IP-Adresse(n)

3.3.4. Lesen Sie die FreeBSD-Errata

Obwohl das FreeBSD Projekt sich bemüht, jede veröffentlichte Version von FreeBSD so stabil wie möglich zu machen, können sich doch gelegentlich Fehler in den Veröffentlichungsprozess einschleichen. In sehr seltenen Fällen betreffen diese Fehler den Installationsvorgang. Wenn diese Probleme entdeckt und behoben sind, werden dazu Hinweise in der FreeBSD Errata (http://www.FreeBSD.org/releases/9.0R/errata.html) auf der FreeBSD Webseite veröffentlicht. Prüfen Sie die Errata vor der Installation, um sicherzustellen, dass es keine Probleme gibt, welche die Installation betreffen.

Informationen und Errata für all diese Veröffentlichungen können über den Abschnitt release information (http://www.FreeBSD.org/releases/index.html) der FreeBSD Webseite (http://www.FreeBSD.org/index.html) abgerufen werden.

3.3.5. Die Installationsmedien beschaffen

Eine FreeBSD-Installation wird durch das starten des Computers mit einer eingelegten FreeBSD-Installations-CD/DVD oder eines USB-Sticks begonnen. Das Installationsprogramm ist kein Programm das aus einem anderen Betriebssystem heraus gestartet werden kann.

Zusätzlich zum Standardinstallationsmedium, welches Kopien von allen FreeBSD-Installationsdateien enthält, gibt es auch eine *bootonly*-Variante. Ein solches Installationsmedium besitzt keine Kopien der Installationsdateien, jedoch kann es diese während der Installation aus dem Netzwerk nachladen. Die bootonly Installations-CD ist dadurch viel kleiner und reduziert die benötigte Bandbreite während der Installation durch herunterladen der allernötigsten Dateien.

Kopien der FreeBSD-Installationsmedien sind auf der FreeBSD Webseite (http://www.FreeBSD.org/where.html#download) verfügbar.

> **Tipp:** Falls Sie bereits eine Kopie von FreeBSD auf CD-ROM, DVD oder USB-Stick besitzen, kann dieser Abschnitt übersprungen werden.

CD und DVD-Images von FreeBSD sind startfähige ISO-Dateien. Nur eine CD oder DVD wird für eine Installation benötigt. Brennen Sie ein ISO-Image auf eine startfähige CD oder DVD mit Hilfe eines CD-Brennprogramms, das für Ihr aktuelles Betriebssystem zur Verfügung steht.

Um einen startfähigen USB-Stick zu erstellen, führen Sie die folgenden Schritte durch:

1. Das Image für den USB-Stick herunterladen

   Das Image für FreeBSD 9.0-RELEASE und höhere kann von dem `ISO-IMAGES/`-Verzeichnis unter `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/`*`arch`*`/`*`arch`*`/ISO-IMAGES/`*`version`*`/FreeBSD-`*`version`*`-RELEASE-` bezogen werden. Ersetzen Sie jeweils *`arch`* und *`version`* mit der Architektur und der Versionsnummer, die Sie installieren möchten. Beispielsweise sind die USB-Stick Images für FreeBSD/i386 9.0-RELEASE verfügbar unter ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/i386/ISO-IMAGES/9.0/FreeBSD-9.0-RELEASE-i386-memstick.img.

> **Tipp:** Für FreeBSD 8.*x* und frühere Versionen wird ein anderer Pfad verwendet. Details für das Herunterladen und Installieren von FreeBSD 8.*x* und frühere werden in Kapitel 2 behandelt.

Das USB-Stick Image hat die Endung `.img`. Das `ISO-IMAGES/`-Verzeichnis enthält eine Vielzahl von verschiedenen Installations-Images und die jeweils benötigte Version von FreeBSD, sowie in manchen Fällen die Zielhardware.

> **Wichtig:** Bevor Sie fortfahren, *machen Sie Sicherungskopien* der Daten auf dem USB-Stick, da die folgende Prozedur alle Daten *löscht*.

2. Das Image auf den USB-Stick schreiben

Den USB-Stick unter FreeBSD vorbereiten

> **Warnung:** Das Beispiel unten verwendet `/dev/da0` als das Zielgerät, auf welches das Image geschrieben werden soll. Seien Sie vorsichtig, dass das richtige Gerät als das Ausgabe benutzt wird oder Sie zerstören wichtige Daten.

1. Das Image mit dd(1) schreiben

   Die `.img`-Datei ist *keine* gewöhnliche Datei. Es ist ein *Image* des kompletten späteren Inhalts des USB-Sticks. Es kann *nicht* einfach wie eine gewöhnliche Datei kopiert werden, sondern muss direkt auf das Zielgerät mit dd(1) geschrieben werden:

   ```
   # dd if=FreeBSD-9.0-RELEASE-i386-memstick.img of=/dev/da0 bs=64k
   ```

Das Image unter Windows schreiben

> **Warnung:** Versichern Sie sich, dass Sie den korrekten Laufwerksbuchstaben als Ausgabe angeben oder Sie überschreiben und zerstören bestehende Daten.

1. **Image Writer für Windows** herunterladen

   Image Writer für Windows ist eine frei verfügbare Anwendung, welche eine Imagedatei korrekt auf einen SB-Stick schreiben kann. Laden Sie diese von https://launchpad.net/win32-image-writer/ herunter und entpacken Sie sie in einen Ordner.

2. Das Image mit Image Writer auf den USB-Stick schreiben

   Klicken Sie doppelt auf das **Win32DiskImager**-Icon, um das Programm zu starten. Prüfen Sie dabei, dass der Laufwerksbuchstabe unter `Device` dem Gerät entspricht, in dem sich der USB-Stick befindet. Klicken Sie auf das Ordnersymbol und wählen Sie das Image aus, welches auf den USB-Stick geschrieben werden soll. Um den Image-Dateinamen zu akzeptieren, klicken Sie auf [Save]. Überprüfen Sie erneut, ob alles stimmt und dass keine Ordner auf dem USB-Stick in anderen Fenstern geöffnet sind. Sobald alles bereit ist, klicken Sie auf [Write], um die Imagedatei auf den USB-Stick zu schreiben.

Anmerkung: Die Installation von Disketten wird nicht mehr unterstützt.

Sie sind jetzt dazu bereit, mit der Installation von FreeBSD zu beginnen.

3.4. Die Installation starten

Wichtig: Es werden durch die Installation keine Änderungen an Ihren Festplatten durchgeführt, so lange Sie nicht die folgende Meldung sehen:

```
Your changes will now be written to disk.  If you
have chosen to overwrite existing data, it will
be PERMANENTLY ERASED. Are you sure you want to
commit your changes?
```

Die Installation kann vor dieser Warnung zu jeder Zeit abgebrochen werden, ohne dass die Inhalte der Festplatte geändert davon betroffen sind. Falls Sie besorgt sind, dass etwas falsch konfiguriert wurde, schalten Sie einfach den Computer vor diesem Punkt aus und es wird kein Schaden angerichtet.

3.4.1. Der Systemstart

3.4.1.1. Systemstart von i386 und amd64

1. Falls Sie einen “startfähigen” USB-Stick einsetzen, wie in Abschnitt 3.3.5 beschrieben ist, dann stecken Sie diesen vor dem Anschalten des Computers hinein.

   Falls Sie von einer CD-ROM starten, müssen Sie den Computer anschalten und die CD-ROM so bald wie möglich einlegen.
2. Konfigurieren Sie Ihren Rechner so, dass er entweder von der CD-ROM oder dem USB-Stick startet, abhängig davon, welches Installationsmedium Sie verwenden. Die Konfiguration im BIOS erlaubt es, das Gerät, von dem gestartet werden soll, auszuwählen. Die meisten Systeme erlauben es auch, das Startgerät während des Startvorgangs zu wählen, typischerweise durch drücken von entweder **F10**, **F11**, **F12** oder **Escape**.
3. Falls Ihr Computer wie normal startet und Ihr bestehendes Betriebssystem lädt, befolgen Sie einen der hier aufgeführten Schritte:
   1. Die Medien wurden während des Startvorgangs nicht früh genug eingelegt. Lassen Sie diese wo sie sind und versuchen Sie, den Rechner davon neu zu starten.
   2. Die Änderungen am BIOS haben nicht richtig funktioniert. Sie sollten diese erneut durchführen, um die richtige Option auszuwälen.
   3. Das von Ihnen verwendete BIOS unterstützt das starten vom gewählten Medium nicht. Der Plop Boot Manager (http://www.plop.at/en/bootmanager.html) kann in diesem Fall verwendet werden, um ältere Computer von CD or USB-Medien zu starten.

4. FreeBSD wird anfangen zu starten. Falls Sie von CD-ROM starten, werden Sie eine Anzeige ähnlich wie die folgende zu sehen bekommen (Versionsinformationen wurden hier entfernt):

```
Booting from CD-ROM...
645MB medium detected
CD Loader 1.2

Building the boot loader arguments
Looking up /BOOT/LOADER... Found
Relocating the loader and the BTX
Starting the BTX loader

BTX loader 1.00 BTX version is 1.02
Consoles: internal video/keyboard
BIOS CD is cd0
BIOS drive C: is disk0
BIOS drive D: is disk1
BIOS 636kB/261056kB available memory

FreeBSD/i386 bootstrap loader, Revision 1.1

Loading /boot/defaults/loader.conf
/boot/kernel/kernel text=0x64daa0 data=0xa4e80+0xa9e40 syms=[0x4+0x6cac0+0x4+0x88e9d]
\
```

5. Der FreeBSD-Bootloader wird angezeigt:

Abbildung 3-1. Das FreeBSD-Bootloader Menü

Warten Sie entweder zehn Sekunden oder drücken Sie **Enter**.

3.4.1.2. Systemstart beim Macintosh PowerPC®

Auf den meisten Maschinen können Sie **C** auf der Tastatur gedrückt halten, um von der CD zu starten. Andernfalls, halten Sie **Command**+**Option**+**O**+**F**, oder **Windows**+**Alt**+**O**+**F** auf nicht-Apple Tastaturen gedrückt. Geben Sie an der `0 >`-Eingabeaufforderung folgendes ein:

```
boot cd:,\ppc\loader cd:0
```

Für Xserves ohne Tastatur, lesen Sie Apples Support Webseite (http://support.apple.com/kb/TA26930) über das starten in die Open Firmware.

3.4.1.3. Systemstart für SPARC64

Die meisten SPARC64-Systeme sind so eingerichtet, dass diese automatisch von CD starten. Um FreeBSD zu installieren, müssen Sie über das Netzwerk oder von einer CD-ROM starten, was es nötig macht, in die PROM OpenFirmware einzubrechen.

Um dies zu tun, starten Sie das System neu und warten Sie bis die Startmeldungen erscheinen. Abhängig vom Modell sollte dies in etwa folgendermaßen aussehen:

```
Sun Blade 100 (UltraSPARC-IIe), Keyboard Present
Copyright 1998-2001 Sun Microsystems, Inc.  All rights reserved.
OpenBoot 4.2, 128 MB memory installed, Serial #51090132.
Ethernet address 0:3:ba:b:92:d4, Host ID: 830b92d4.
```

Falls Ihr System damit fortfährt, von diesem Zeitpunkt an von Platte zu starten, müssen Sie **L1**+**A** oder **Stop**+**A** auf der Tastatur eingeben oder ein `BREAK`-Kommando (indem Sie z.B. ~# in tip(1) oder cu(1) absetzen) über die serielle Konsole senden, um zur PROM Befehlszeile zu gelangen. Es sieht dann so aus:

```
ok     ❶
ok {0} ❷
```

❶ Dies ist die Eingabeaufforderung, welche auf Systemen mit nur einer CPU verwendet wird.

❷ Dies ist die Eingabeaufforderung auf SMP-Systemen. Die Zahl gibt die Nummer der aktiven CPU an.

An dieser Eingabeaufforderung angekommen, legen Sie nun die CD-ROM in Ihr Laufwerk und geben Sie `boot cdrom` ein.

3.4.2. Die Geräteerkennung prüfen

Die letzten hundert Zeilen, die am Bildschirm angezeigt wurden, sind gespeichert worden und können erneut abgerufen werden.

Um diesen Puffer anzusehen, drücken Sie **Scroll Lock**. Das bewirkt, dass Sie die Bildschirmanzeige hoch und runter bewegen (scrollen) können. Sie können dann die Pfeiltasten oder **PageUp** und **PageDown** benutzen, um die Meldungen zu sehen. Drücken Sie **Scroll Lock** erneut, um das scrollen zu stoppen.

Tun Sie dies jetzt, um den Text, der aus den Bildschirm gelaufen ist, als der Kernel die Geräteerkennung durchgeführt hat, erneut zu prüfen. Sie werden einen Text ähnlich zu Abbildung 3-2 sehen, obwohl sich der genaue Text, abhängig von den Geräten in Ihrem Computer, unterscheiden wird.

Abbildung 3-2. Typical Device Probe Results

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
```

```
FreeBSD is a registered trademark of The FreeBSD Foundation.
FreeBSD 9.0-RELEASE #0 r225473M: Sun Sep 11 16:07:30 BST 2011
    root@psi:/usr/obj/usr/src/sys/GENERIC amd64
CPU: Intel(R) Core(TM)2 Duo CPU     T9400  @ 2.53GHz (2527.05-MHz K8-class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  Stepping = 6
  Features=0xbfebfbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,CLF
  Features2=0x8e3fd<SSE3,DTES64,MON,DS_CPL,VMX,SMX,EST,TM2,SSSE3,CX16,xTPR,PDCM,SSE4.1>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
  TSC: P-state invariant, performance statistics
real memory  = 3221225472 (3072 MB)
avail memory = 2926649344 (2791 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <TOSHIB A0064   >
FreeBSD/SMP: Multiprocessor System Detected: 2 CPUs
FreeBSD/SMP: 1 package(s) x 2 core(s)
 cpu0 (BSP): APIC ID:  0
 cpu1 (AP): APIC ID:  1
ioapic0: Changing APIC ID to 1
ioapic0 <Version 2.0> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <TOSHIB A0064> on motherboard
acpi0: Power Button (fixed)
acpi0: reservation of 0, a0000 (3) failed
acpi0: reservation of 100000, b6690000 (3) failed
Timecounter "ACPI-safe" frequency 3579545 Hz quality 850
acpi_timer0: <24-bit timer at 3.579545MHz> port 0xd808-0xd80b on acpi0
cpu0: <ACPI CPU> on acpi0
ACPI Warning: Incorrect checksum in table [ASF!] - 0xFE, should be 0x9A (20110527/tbutils-282)
cpu1: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
vgapci0: <VGA-compatible display> port 0xcff8-0xcfff mem 0xff400000-0xff7fffff,0xe0000000-0xefff
agp0: <Intel GM45 SVGA controller> on vgapci0
agp0: aperture size is 256M, detected 131068k stolen memory
vgapci1: <VGA-compatible display> mem 0xffc00000-0xffcfffff at device 2.1 on pci0
pci0: <simple comms> at device 3.0 (no driver attached)
em0: <Intel(R) PRO/1000 Network Connection 7.2.3> port 0xcf80-0xcf9f mem 0xff9c0000-0xff9dffff,0x
em0: Using an MSI interrupt
em0: Ethernet address: 00:1c:7e:6a:ca:b0
uhci0: <Intel 82801I (ICH9) USB controller> port 0xcf60-0xcf7f irq 16 at device 26.0 on pci0
usbus0: <Intel 82801I (ICH9) USB controller> on uhci0
uhci1: <Intel 82801I (ICH9) USB controller> port 0xcf40-0xcf5f irq 21 at device 26.1 on pci0
usbus1: <Intel 82801I (ICH9) USB controller> on uhci1
uhci2: <Intel 82801I (ICH9) USB controller> port 0xcf20-0xcf3f irq 19 at device 26.2 on pci0
usbus2: <Intel 82801I (ICH9) USB controller> on uhci2
ehci0: <Intel 82801I (ICH9) USB 2.0 controller> mem 0xff9ff800-0xff9ffbff irq 19 at device 26.7 on
usbus3: EHCI version 1.0
usbus3: <Intel 82801I (ICH9) USB 2.0 controller> on ehci0
hdac0: <Intel 82801I High Definition Audio Controller> mem 0xff9f8000-0xff9fbfff irq 22 at device
pcib1: <ACPI PCI-PCI bridge> irq 17 at device 28.0 on pci0
pci1: <ACPI PCI bus> on pcib1
iwn0: <Intel(R) WiFi Link 5100> mem 0xff8fe000-0xff8fffff irq 16 at device 0.0 on pci1
```

```
pcib2: <ACPI PCI-PCI bridge> irq 16 at device 28.1 on pci0
pci2: <ACPI PCI bus> on pcib2
pcib3: <ACPI PCI-PCI bridge> irq 18 at device 28.2 on pci0
pci4: <ACPI PCI bus> on pcib3
pcib4: <ACPI PCI-PCI bridge> at device 30.0 on pci0
pci5: <ACPI PCI bus> on pcib4
cbb0: <RF5C476 PCI-CardBus Bridge> at device 11.0 on pci5
cardbus0: <CardBus bus> on cbb0
pccard0: <16-bit PCCard bus> on cbb0
isab0: <PCI-ISA bridge> at device 31.0 on pci0
isa0: <ISA bus> on isab0
ahci0: <Intel ICH9M AHCI SATA controller> port 0x8f58-0x8f5f,0x8f54-0x8f57,0x8f48-0x8f4f,0x8f44-0
ahci0: AHCI v1.20 with 4 3Gbps ports, Port Multiplier not supported
ahcich0: <AHCI channel> at channel 0 on ahci0
ahcich1: <AHCI channel> at channel 1 on ahci0
ahcich2: <AHCI channel> at channel 4 on ahci0
acpi_lid0: <Control Method Lid Switch> on acpi0
battery0: <ACPI Control Method Battery> on acpi0
acpi_button0: <Power Button> on acpi0
acpi_acad0: <AC Adapter> on acpi0
acpi_toshiba0: <Toshiba HCI Extras> on acpi0
acpi_tz0: <Thermal Zone> on acpi0
attimer0: <AT timer> port 0x40-0x43 irq 0 on acpi0
Timecounter "i8254" frequency 1193182 Hz quality 0
Event timer "i8254" frequency 1193182 Hz quality 100
atkbdc0: <Keyboard controller (i8042)> port 0x60,0x64 irq 1 on acpi0
atkbd0: <AT Keyboard> irq 1 on atkbdc0
kbd0 at atkbd0
atkbd0: [GIANT-LOCKED]
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: model GlidePoint, device ID 0
atrtc0: <AT realtime clock> port 0x70-0x71 irq 8 on acpi0
Event timer "RTC" frequency 32768 Hz quality 0
hpet0: <High Precision Event Timer> iomem 0xfed00000-0xfed003ff on acpi0
Timecounter "HPET" frequency 14318180 Hz quality 950
Event timer "HPET" frequency 14318180 Hz quality 450
Event timer "HPET1" frequency 14318180 Hz quality 440
Event timer "HPET2" frequency 14318180 Hz quality 440
Event timer "HPET3" frequency 14318180 Hz quality 440
uart0: <16550 or compatible> port 0x3f8-0x3ff irq 4 flags 0x10 on acpi0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
ppc0: cannot reserve I/O port range
est0: <Enhanced SpeedStep Frequency Control> on cpu0
p4tcc0: <CPU Frequency Thermal Control> on cpu0
est1: <Enhanced SpeedStep Frequency Control> on cpu1
p4tcc1: <CPU Frequency Thermal Control> on cpu1
Timecounters tick every 1.000 msec
hdac0: HDA Codec #0: Realtek ALC268
hdac0: HDA Codec #1: Lucent/Agere Systems (Unknown)
pcm0: <HDA Realtek ALC268 PCM #0 Analog> at cad 0 nid 1 on hdac0
```

```
pcm1: <HDA Realtek ALC268 PCM #1 Analog> at cad 0 nid 1 on hdac0
usbus0: 12Mbps Full Speed USB v1.0
usbus1: 12Mbps Full Speed USB v1.0
usbus2: 12Mbps Full Speed USB v1.0
usbus3: 480Mbps High Speed USB v2.0
ugen0.1: <Intel> at usbus0
uhub0: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus0
ugen1.1: <Intel> at usbus1
uhub1: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus1
ugen2.1: <Intel> at usbus2
uhub2: <Intel UHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus2
ugen3.1: <Intel> at usbus3
uhub3: <Intel EHCI root HUB, class 9/0, rev 2.00/1.00, addr 1> on usbus3
uhub0: 2 ports with 2 removable, self powered
uhub1: 2 ports with 2 removable, self powered
uhub2: 2 ports with 2 removable, self powered
uhub3: 6 ports with 6 removable, self powered
ugen2.2: <vendor 0x0b97> at usbus2
uhub8: <vendor 0x0b97 product 0x7761, class 9/0, rev 1.10/1.10, addr 2> on usbus2
ugen1.2: <Microsoft> at usbus1
ada0 at ahcich0 bus 0 scbus1 target 0 lun 0
ada0: <Hitachi HTS543225L9SA00 FBEOC43C> ATA-8 SATA 1.x device
ada0: 150.000MB/s transfers (SATA 1.x, UDMA6, PIO 8192bytes)
ada0: Command Queueing enabled
ada0: 238475MB (488397168 512 byte sectors: 16H 63S/T 16383C)
ada0: Previously was known as ad4
ums0: <Microsoft Microsoft 3-Button Mouse with IntelliEyeTM, class 0/0, rev 1.10/3.00, addr 2> on
SMP: AP CPU #1 Launched!
cd0 at ahcich1 bus 0 scbus2 target 0 lun 0
cd0: <TEAC DV-W28S-RT 7.0C> Removable CD-ROM SCSI-0 device
cd0: 150.000MB/s transfers (SATA 1.x, ums0: 3 buttons and [XYZ] coordinates ID=0
UDMA2, ATAPI 12bytes, PIO 8192bytes)
cd0: cd present [1 x 2048 byte records]
ugen0.2: <Microsoft> at usbus0
ukbd0: <Microsoft Natural Ergonomic Keyboard 4000, class 0/0, rev 2.00/1.73, addr 2> on usbus0
kbd2 at ukbd0
uhid0: <Microsoft Natural Ergonomic Keyboard 4000, class 0/0, rev 2.00/1.73, addr 2> on usbus0
Trying to mount root from cd9660:/dev/iso9660/FREEBSD_INSTALL [ro]...
```

Prüfen Sie die Ergebnisse der Geräteerkennung genau, um sicher zu stellen, dass FreeBSD alle Geräte, die Sie erwarten, auch gefunden hat. Falls ein Gerät nicht gefunden wurde, wird es auch nicht aufgelistet. Kernelmodule erlauben es, Unterstützung für Geräte, die nicht im `GENERIC`-Kernel vorhanden sind, hinzuzufügen.

Nach der Geräteerkennungsprozedur, werden Sie Abbildung 3-3 sehen. Das Installationsmedium kann auf drei Arten verwendet werden: um FreeBSD zu installieren, als eine "live CD" oder um einfach eine FreeBSD-Shell zu öffnen. Benutzen Sie die Pfeiltasten, um eine Option auszuwählen und druücken Sie **Enter** zum bestätigen.

Abbildung 3-3. Auswahl der Verwendung des Installationsmediums

Wählen Sie hier [Install], gelangen Sie in das Installationsprogramm.

3.5. Das bsdinstall-Werkzeug

bsdinstall ist ein textbasiertes FreeBSD Installationsprogramm, geschrieben von Nathan Whitehorn und im Jahr 2011 für FreeBSD 9.0 vorgestellt wurde.

> **Anmerkung:** Kris Moores **pc-sysinstall** ist in PC-BSD (http://pcbsd.org) enthalten und kann ebenfalls verwendet werden, um FreeBSD zu installieren (http://wiki.pcbsd.org/index.php/Use_PC-BSD_Installer_to_Install_FreeBSD). Obwohl es manchmal mit **bsdinstall** verwechselt wird, sind die beiden Programme nicht miteinander verwandt.

Das **bsdinstall** Menüsystem wird durch die Pfeiltasten gesteuert, **Enter**, **Tab**, **Space** und andere Tasten.

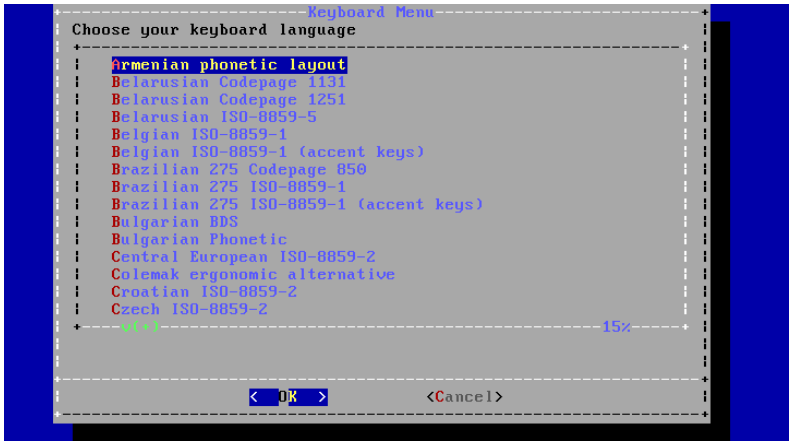
3.5.1. Die Tastaturbelegung auswählen

Abhängig davon, welche Systemkonsole verwendet wird, fragt **bsdinstall** am Anfang ab, ob eine nicht-Standard Tastaturbelegung festgelegt werden soll.

Abbildung 3-4. Tastaturbelegung festlegen

Wenn [YES] ausgewählt wird, wird der folgende Tastaturauswahlbildschirm angezeigt. Andernfalls wird dieser Auswahlbildschirm nicht gezeigt und eine Standardtastaturbelegung genutzt.

Abbildung 3-5. Tastaturauswahlbildschirm



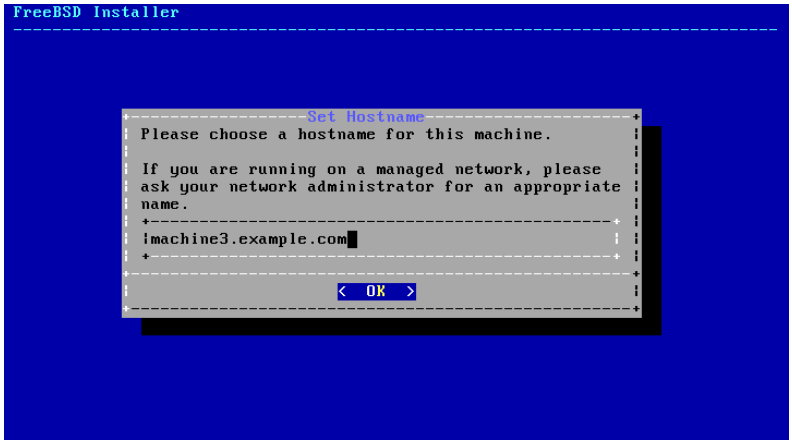
Wählen Sie die Tastenbelegung, die Ihrer am System angeschlossenen Tastatur am nächsten kommt, indem Sie die Pfeiltasten Hoch/Runter verwenden und anschliessend **Enter** drücken.

> **Anmerkung:** Durch drücken von **Esc** wird die Standardbelegung eingestellt. United States of America ISO-8859-1 ist eine sichere Option, falls Sie sich unsicher sind, welche Auswahl Sie treffen sollen.

3.5.2. Den Rechnernamen festlegen

Als nächstes fragt Sie **bsdinstall** nach dem Rechnernamen, der in dem neu zu installierenden System verwendet werden soll.

Abbildung 3-6. Festlegen des Rechnernamens

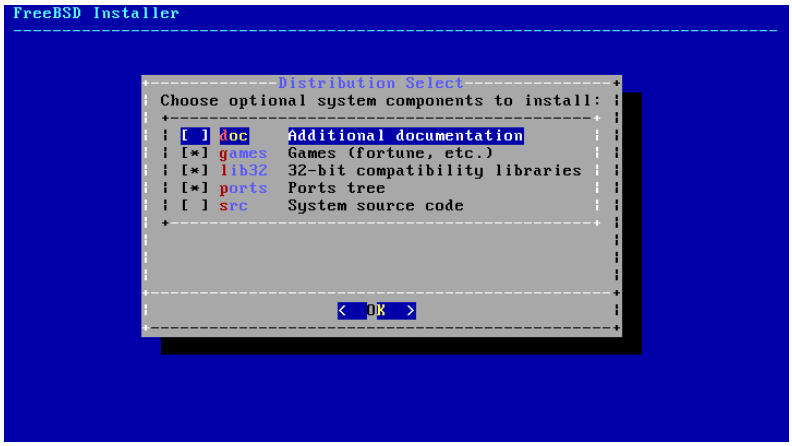


Der eingegebene Rechnername sollte ein voll-qualifizierter Rechnername sein, so wie z.B. `machine3.example.com`

3.5.3. Auswahl der zu installierenden Komponenten

Im nächsten Schritt fragt Sie **bsdinstall**, die optionalen Komponenten für die Installation auszuwählen.

Abbildung 3-7. Komponenten für die Installation auswählen



Die Entscheidung, welche Komponenten auszuwählen sind, hängt grösstenteils davon ab, für was das System künftig eingesetzt werden soll und der zur Verfügung stehende Plattenplatz. Der FreeBSD-Kernel und die Systemprogramme (zusammengenommen auch als “Basissystem” bezeichnet) werden immer installiert.

Abhängig vom Typ der Installation, werden manche dieser Komponenten nicht erscheinen.

Optionale Komponenten

- `doc` - Zusätzliche Dokumentation, meistens eher von historischem Interesse. Dokumentation, wie Sie vom FreeBSD Dokumentationsprojekt bereitgestellt wird, kann zu einem späteren Zeitpunkt noch installiert werden.

- `games` - Mehrere traditionelle BSD-Spiele, sowohl **fortune**, **rot13** und andere.
- `lib32` - Kompatibilitäts-Bibliotheken, um 32-bit Anwendungen auf der 64-bit Version von FreeBSD laufen zu lassen.
- `ports` - Die FreeBSD Ports-Sammlung.

  Die Ports-Sammlung stellt eine einfache und praktische Art dar, Software zu installieren. Die Ports-Sammlung enthält nicht den nötigen Quellcode, um die Software zu erstellen. Stattdessen handelt es sich um einen Sammlung von Dateien, die das herunterladen, erstellen und installieren von Drittanbietersoftware automatisiert. Kapitel 5 behandelt die Verwendung der Ports-Sammlung.

  > **Warnung:** Das Installationsprogramm prüft nicht, ob genügend Plattenplatz zur Verfügung steht. Wählen Sie diese Option nur, wenn Sie über ausreichend Festplattenspeicher verfügen. Seit FreeBSD 9.0, nimmt die Ports-Sammlung etwa ports.size; Plattenplatz ein. Sie können für neuere Versionen von FreeBSD einen grösseren Wert annehmen.

- `src` - Quellcode für das System.

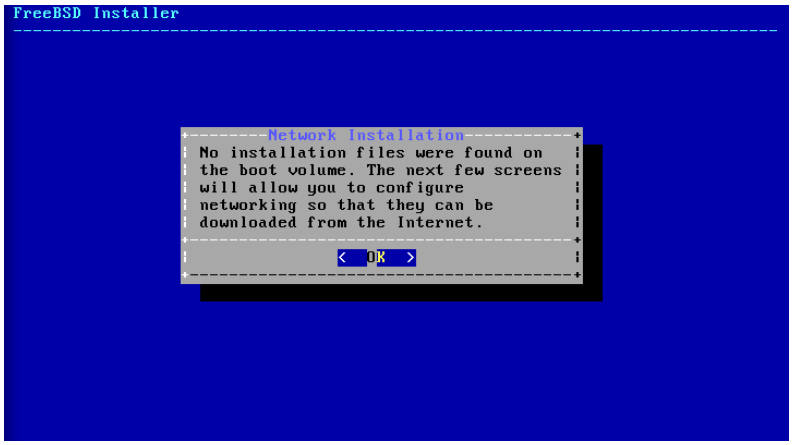
  FreeBSD wird mit allen Quellen für den Kernel und die Systemprogramme ausgeliefert. Obwohl dies für die meisten Anwendungen nicht benötigt wird, kann es doch für manche Software, die als Quellcode verbreitet wird (beispielsweise Gerätetreiber oder Kernelmodule), oder um an FreeBSD selbst mitzuentwickeln, notwendig sein.

  Der komplette Quellcodebaum benötigt 1 GB Plattenplatz und um das gesamte Betriebssystem neu zu erstellen, werden zusätzliche 5 GB Platz benötigt.

3.6. Installation aus dem Netzwerk

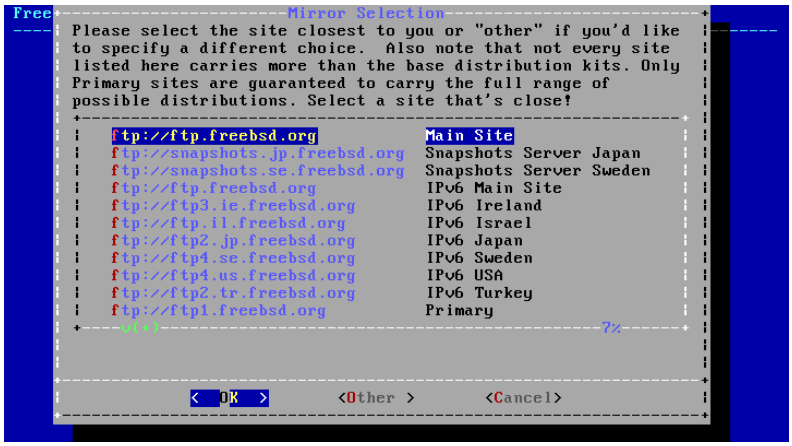
Die *bootonly*-Installationsmedien enthält keine Kopien der Installationsdateien. Wenn eine *bootonly*-Installationsmethode verwendet wird, müssen die Dateien über eine Netzwerkverbindung übertragen werden, sobald diese benötigt werden.

Abbildung 3-8. Installation über das Netzwerk



Nachdem die Netzwerkverbindung wie in Abschnitt 3.9.2 konfiguriert wurde, kann ein Spiegelserver ausgewählt werden. Spiegelserver dienen zur Zwischenspeicherung von Kopien der FreeBSD-Dateien. Wählen Sie einen Spiegelserver, welcher in der gleichen Region auf der Welt beheimatet ist, wie der Computer, auf dem FreeBSD installiert werden soll. Dateien können so viel schneller übertragen werden, wenn der Spiegelserver sich n¨her am Zielcomputer befindet und die Installationszeit wird somit reduziert.

Abbildung 3-9. Einen Spiegelserver wählen

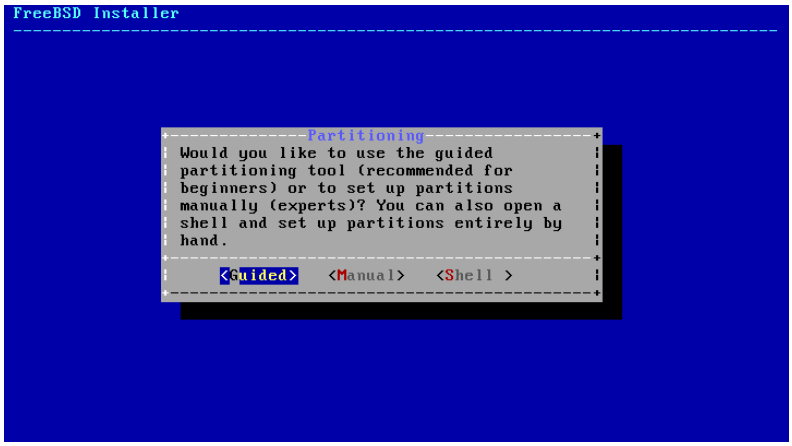


Die Installation wird auf die gleiche Weise fortfahren, als würden die Installationsdateien auf einem lokalen Medium vorliegen.

3.7. Plattenplatz bereitstellen

Es gibt drei Arten, Plattenplatz für FreeBSD zur Verfügung zu stellen. *Geführte* Partitionierung richtet Partitionen automatisch ein, während *manuelle* Partitionierung es fortgeschrittenen Anwendern erlaubt, selbstgewählte Partitionen zu erzeugen. Schliesslich gbt es noch die Option eine Shell zu starten, auf der Kommandozeilenprogramme wie gpart(8), fdisk(8) und bsdlabel(8) direkt ausgeführt werden können.

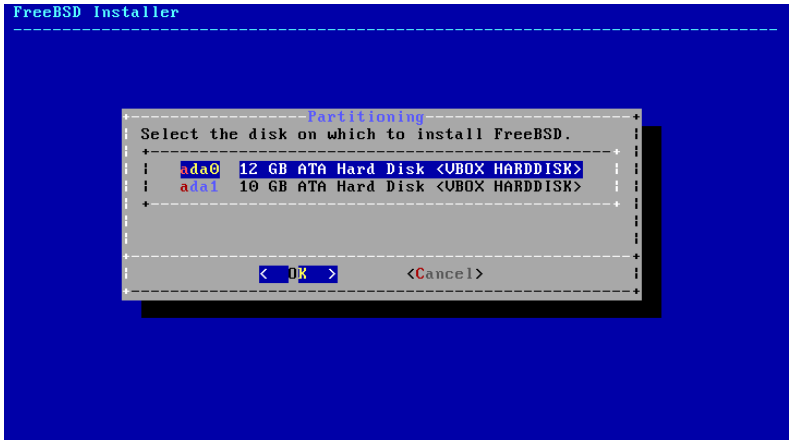
Abbildung 3-10. Geführte oder manuelle Partitionierung auswählen



3.7.1. Geführte Partitionierung

Sollten mehrere Platten angeschlossen sein, wählen Sie diejenige aus, auf der FreeBSD installiert werden soll.

Abbildung 3-11. Aus mehreren Platten eine auswählen

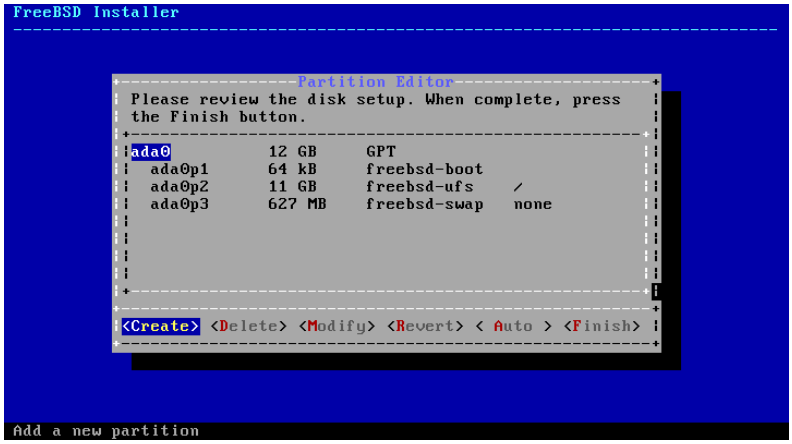


Die gesamte Festplatte oder nur ein Teil davon kann für FreeBSD verwendet werden. Ein allgemeines Partitionslayout, das die gesamte Platte einnimmt wird erstellt, wenn [Entire Disk] ausgewählt wird. Durch die Wahl von [Partition] wird ein Partitionslayout im unbenutzten Speicherplatz der Platte eingerichtet.

Abbildung 3-12. Auswahl der gesamten Platte oder einer Partition

Nachdem das Partitionslayout nun erstellt wurde, sollten Sie es danach noch einmal auf Korrektheit prüfen. Sollten Sie einen Fehler gemacht haben, können Sie durch Auswahl von [Revert] wieder die ursprünglichen Partitionen setzen oder durch [Auto] die automatischen FreeBSD Partitionen wiederherstellen. Partitionen können manuell erstellt, geändert oder gelöscht werden. Sollte die Partitionierung richtig sein, wählen Sie [Finish] aus, um mit der Installation fortzufahren.

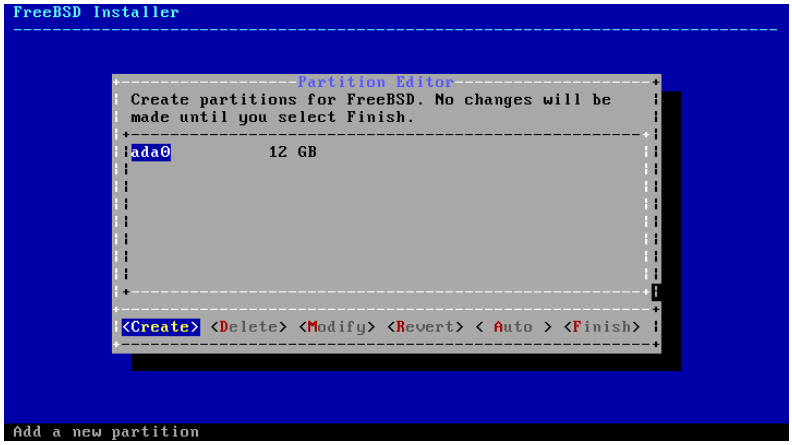
Abbildung 3-13. Überprüfen der erstellen Partitionen



3.7.2. Manuelle Partitionierung

Manuelle Partitionierung führt Sie direkt zum Partitionseditor.

Abbildung 3-14. Partitionen manuell erstellen



Durch hervorheben einer Platte (in diesem Fall `ada0`) und die Auswahl von [Create], wird ein Menü zur Wahl des *Partitionierungsschemas* angezeigt.

Abbildung 3-15. Partitionen manuell anlegen

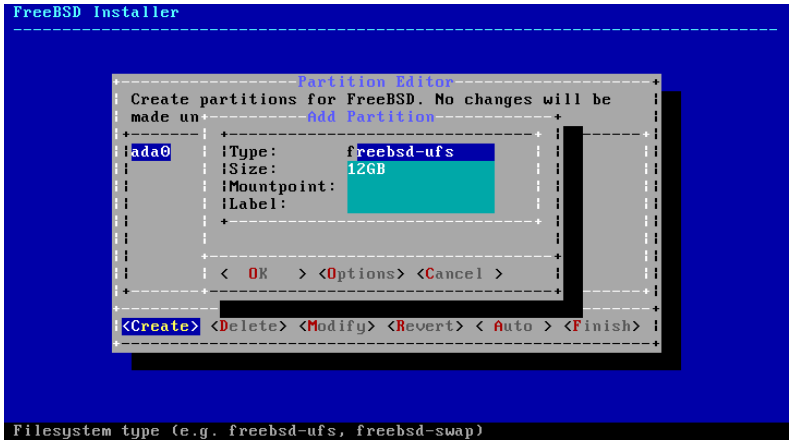
GPT-Partitionierung ist normalerweise die passendste Auswahl für PC-kompatible Rechner. Ältere PC Betriebssysteme, die nicht mit GPT kompatibel und benötigen stattdessen MBR-Partitionen. Die anderen Partitionsschemata werden für gewöhnlich für ältere Computersysteme benutzt.

Tabelle 3-1. Partitionierungsschemas

Abkürzung	Beschreibung
APM	Apple Partition Map, von PowerPC® Macintosh verwendet. (http://support.apple.com/kb/TA21692)
BSD	BSD-Labels ohne einen MBR, manchmal auch "dangerously dedicated mode" genannt. Lesen Sie dazu bsdlabel(8).
GPT	GUID Partition Table. (http://en.wikipedia.org/wiki/GUID_Partition_Table)
MBR	Master Boot Record. (http://en.wikipedia.org/wiki/Master_boot_record)
PC98	MBR-Variante, verwendet von NEC PC-98 Computern. (http://en.wikipedia.org/wiki/Pc9801)
VTOC8	Volume Table Of Contents, von Sun SPARC64 und UltraSPARC Computern verwendet.

Nachdem das Partitionierungsschema ausgewählt und erstellt wurde, werden durch erneute Auswahl von [Create] neue Partitionen erzeugt.

Abbildung 3-16. Partitionen manuell erzeugen



Eine FreeBSD-Standardinstallation mit GPT legt mindestens die folgenden drei Partitionen an:

Standard-FreeBSD GPT-Partitionen

- `freebsd-boot` - FreeBSD-Bootcode. Diese Partition muss die erste auf der Festplatte sein.
- `freebsd-ufs` - Ein FreeBSD UFS-Dateisystem.
- `freebsd-swap` - FreeBSD Auslagerungsbereich (swap space).

Mehere Dateisystempartitionen können benutzt werden und manche Leute ziehen es vor, ein traditionelles Layout mit getrennten Partitionen für die Dateisysteme `/`, `/var`, `/tmp` und `/usr` zu erstellen. Lesen Sie dazu Beispiel 3-3, um ein Beispiel zu erhalten.

Lesen Sie gpart(8) für eine vollständige Liste von verfügbaren GPT-Partitionstypen.

Grössenangaben können mit gängigen Abkürzungen eingegeben werden: *K* für Kilobytes, *M* für Megabytes oder *G* für Gigabytes.

> **Tipp:** Korrekte Sektorausrichtung ermöglicht grösstmögliche Geschwindigkeit und das Anlegen von Partitionsgrössen als vielfaches von 4K-Bytes hilft, die passende Ausrichtung auf Platten mit entweder 512-Bytes oder 4K-Bytes Sektorgrössen, festzulegen. Generell sollte die Verwendung von Partitionsgrössen, die sogar vielfache von 1M oder 1G sind, den einfachsten Weg darstellen, um sicher zu stellen, dass jede Partition an einem vielfachen von 4K beginnt. Eine Ausnahme gibt es: momentan sollte die *freebsd-boot*-Partition aufgrund von Beschränkungen im Bootcode nicht grösser sein als 512K.

Ein Einhägepunkt wird benötigt, falls diese Partition ein Dateisystem enthält. Falls nur eine einzelne UFS-Partition erstellt wird, sollte der Einhängepunkt `/` lauten.

Ein *label* wird ebenfalls benötigt. Ein Label ist ein Name, durch den diese Partition angesprochen wird. Festplattennamen oder -nummern können sich ändern, falls die Platte einmal an einem anderen Controller oder Port angeschlossen sein sollte, doch das Partitionslabel ändert sich dadurch nicht. Anstatt auf Plattennamen und Partitionsnummern in Dateien wie `/etc/fstab` zu verweisen, sorgen Labels dafür, dass das System Hardwäreänderungen eher toleriert. GPT-Labels erscheinen in `/dev/gpt/`, wenn eine Platte angeschlossen wird. Andere Partitionierungsschemas besitzen unterschiedliche Fähigkeiten, Labels zu verwenden und diese erscheinen in anderen `/dev/`-Verzeichnissen.

Tipp: Vergeben Sie ein einzigartiges Label auf jedem Dateisystem um Konflikte mit identischen Labels zu verhindern. Ein paar Buchstaben des Computernamens, dessen Verwendungszweck oder Ortes kann dem Label hinzugefügt werden. Beispielsweise "labroot" oder "rootfs-lab" für die UFS root-Partition auf einem Laborrechner.

Beispiel 3-3. Ein traditionelles, partitioniertes Dateisystem erstellen

Für ein traditionelles Partitionslayout, in dem sich `/`, `/var`, `/tmp` und `/usr` in getrennten Partitionen befinden sollen, erstellen Sie ein GPT-Partitionsschema und anschliessend die Partitionen selbst. Die gezeigten Partitionsgrössen sind typisch für eine Festplatte von 20 G. Falls mehr Platz verfügbar ist, sind grössere Swap oder `/var`-Partitionen nützlich. Den hier gezeigten Beschreibungen sind `bsp` für "Beispiel" vorangestellt, jedoch sollten Sie andere, einzigartige Beschreibungen verwenden, wie oben beschrieben.

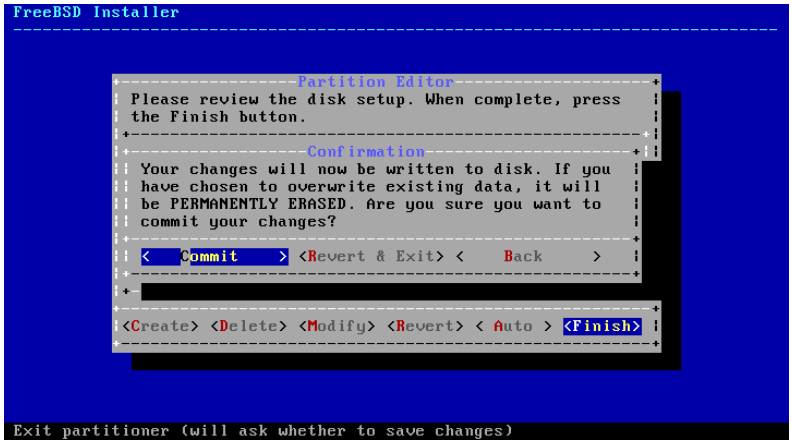
Partitionstyp	Grösse	Eingehängt als	Beschreibung
`freebsd-boot`	`512K`		
`freebsd-ufs`	`2G`	`/`	`bsprootfs`
`freebsd-swap`	`4G`		`bspswap`
`freebsd-ufs`	`2G`	`/var`	`bspvarfs`
`freebsd-ufs`	`1G`	`/tmp`	`bsptmpfs`
`freebsd-ufs`	Akzeptieren Sie die Standardeinstellungen (Rest der Platte)	`/usr`	`bspusrfs`

Nachdem die selbstgewählten Partitionen erzeugt wurden, wählen Sie [Finish], um mit der Installation fortzusetzen.

3.8. Die Installation festschreiben

Dies ist die letzte Chance, die Installation abzubrechen, ohne Änderungen an den Festplatten vorzunehmen.

Abbildung 3-17. Letzte Bestätigung



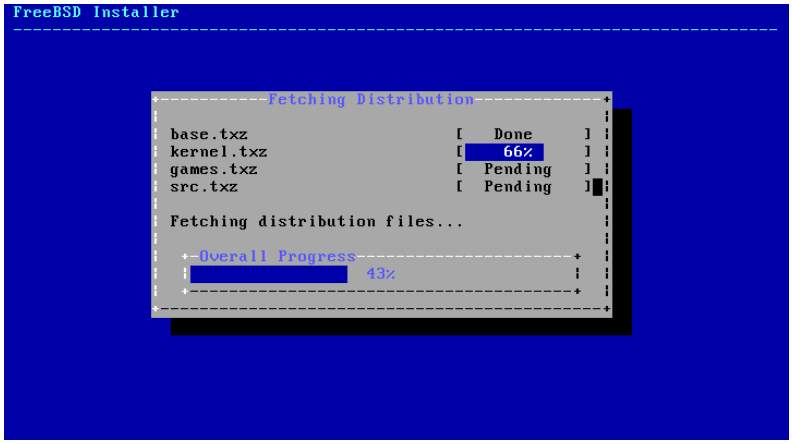
Wählen Sie [Commit] und drücken Sie **Enter**, um fortzufahren. Fall noch Änderungen zu machen sind, wählen Sie [Back], um zum Partitionseditor zurück zu gelangen. Mittels [Revert & Exit] wird das Installationsprogramm beendet, ohne Änderungen an den Festplatten durchzuführen.

Die Installationsdauer hängt von den gewählten Distributionen, dem Installationsmedium und der Geschwindigkeit des Computers ab. Eine Reihe von Nachrichten werden angezeigt, um den Fortschritt darzustellen.

Zuerst wird das Installationsprogramm die Partitionen auf die Platte schreiben und den Befehl `newfs` ausführen, um die Partitionen zu initialisieren.

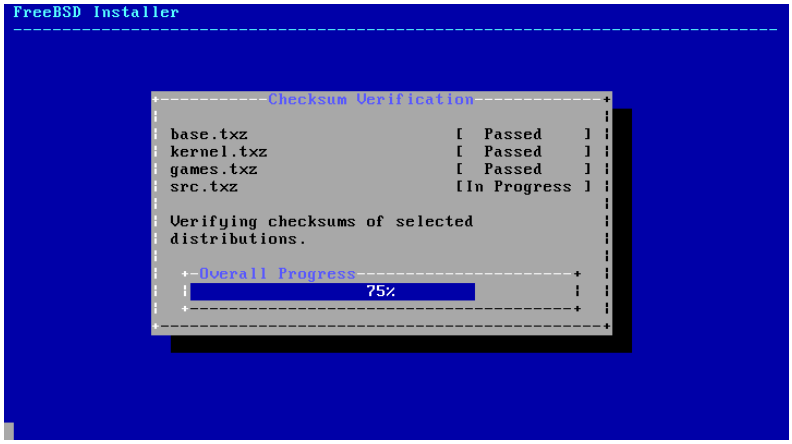
Falls Sie eine Netzwerkinstallation vornehmen, wird **bsdinstall** dann mit dem herunterladen der benötigten Distributionsdateien fortfahren.

Abbildung 3-18. Herunterladen der Distributionsdateien



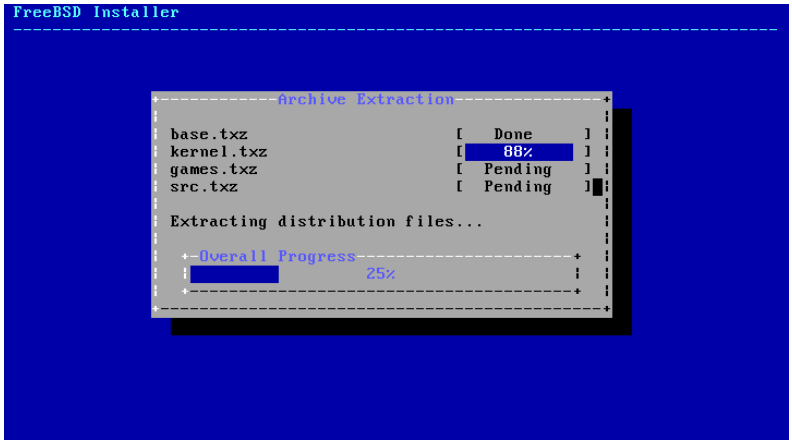
Als nächstes wird die Integrität der Distributionsdateien überprüft, um sicherzustellen, dass diese während des Ladevorgangs nicht beschädigt oder unsauber vom Installationsmedium gelesen wurden.

Abbildung 3-19. Überprüfen der Distributionsdateien



Zum Schluss werden die überprüften Distributionsdateien auf die Festplatte entpackt.

Abbildung 3-20. Entpacken der Distributionsdateien



Sobald alle benötigten Distributionsdateien entpackt wurden, wird **bsdinstall** direkt mit den Arbeiten nach der Installation fortsetzen (siehe Abschnitt 3.9).

3.9. Arbeiten nach der Installation

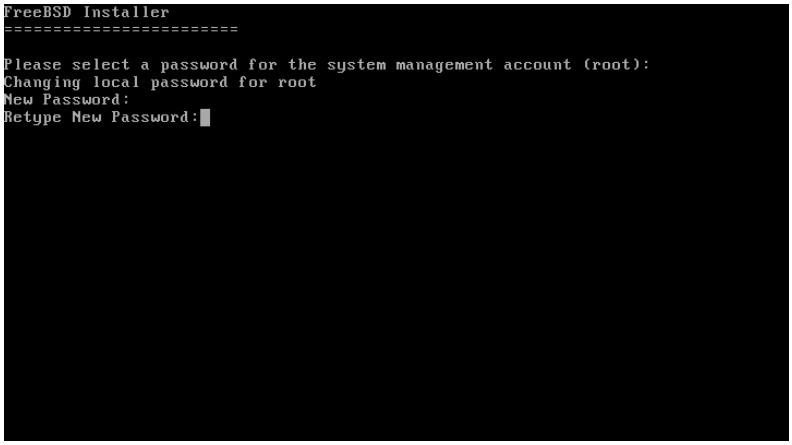
Die Konfiguration von verschiedenen Optionen folgt auf eine erfolgreiche FreeBSD-Installation. Eine solche Option kann durch das erneute betreten der Konfigurationsoptionen aus dem letzten Menü vor dem Neustart in das gerade installierte FreeBSD-System angepasst werden.

3.9.1. Setzen des `root`-Passworts

Das `root`-Password muss gesetzt werden. Wichtig ist dabei zu wissen, dass die eingegebenen Zeichen nicht auf dem Bildschirm angezeigt werden. Nachdem das Passwort eingegeben wurde, muss es zur Bestätigung erneut eingetippt

werden. Damit werden auch Tippfehler verhindert.

Abbildung 3-21. Das `root`-Passwort setzen



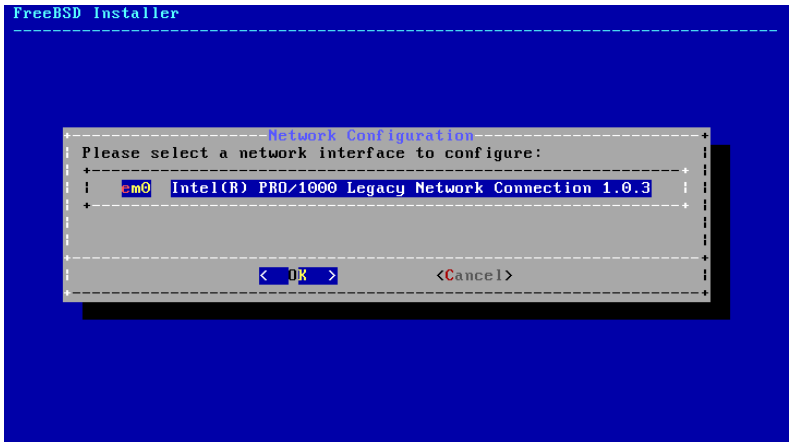
Nachdem das Passwort erfolgreich gesetzt wurde, wird die Installation nun fortgesetzt.

3.9.2. Die Netzwerkschnittstelle konfigurieren

> **Anmerkung:** Die Netzwerkkonfiguration wird übersprungen, falls dies bereits als Teil der *bootonly* durchgeführt worden ist.

Eine Liste aller gefundenen Netzwerkschnittstellen, die auf diesem Computer gefunden wurden, wird als nächstes angezeigt. Wählen Sie davon eine aus, um diese zu konfigurieren.

Abbildung 3-22. Eine zu konfigurierende Netzwerkschnittstelle auswählen



3.9.2.1. Eine drahtlose Netzwerkverbindung einrichten

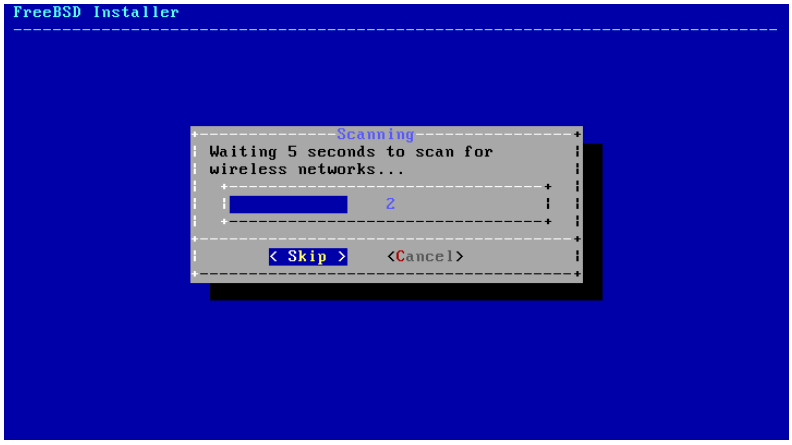
Sollte eine drahtlose Netzwerkverbindung ausgewählt worden sein, müssen WLAN-Identifikation und Sicherheitsparameter nun eingegeben werden, um die Verbindung mit dem Netzwerk herzustellen.

Drahtlose Netzwerke werden durch einen Service Set Identifier oder auch SSID genannt, identifiziert. Der SSID ist ein kurzer, eindeutiger Namen, der für jedes Netzwerk vergeben wird.

Die meisten drahtlosen Netzwerke verschlüsseln die übertragenen Daten, um die Information darin vor unautorisiertem Zugriff zu schützen. Die Verwendung von WPA2-Verschlüsselung wird empfohlen. Ältere Verschlüsselungstypen, wie WEP, bieten nur sehr wenig Sicherheit.

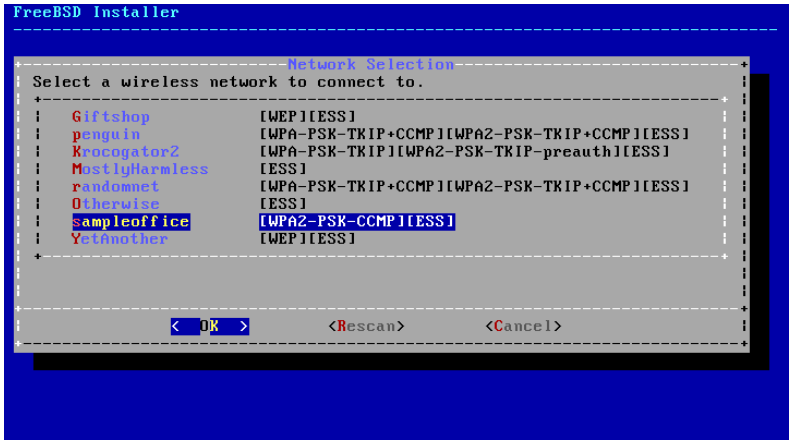
Der erste Schritt des Verbindungsaufbaus ist das drahtlose Netzwerk nach drahtlosen Zugriffspunkten (access points) zu scannen.

Abbildung 3-23. Nach drahtlosen Access Points scannen



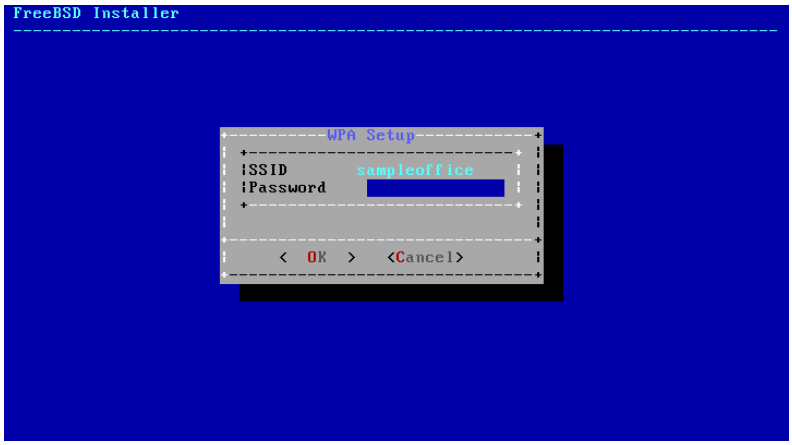
SSIDs, die während des scannens gefunden wurden, werden aufgelistet, gefolgt von einer Beschreibung der Verschlüsselungsarten, die für dieses Netzwerk verfügbar sind. Falls die gewünschte SSID nicht in der Liste auftaucht, wählen Sie [Rescan], um erneut einen Scanvorgang durchzuführen. Falls dann das gewünschte Netzwerk immer noch nicht erscheint, überprüfen Sie Ihre Antenne auf Verbindungsprobleme oder versuchen Sie, näher an den Access point zu gelangen. Scannen Sie erneut nach jeder vorgenommenen Änderung.

Abbildung 3-24. Ein drahtloses Netzwerk auswählen



Die Verschlüsselungsinformationen, um sich mit dem Netzwerk zu verbinden, werden nach der Auswahl des Netzwerks eingegeben. Mit WPA2 wird nur ein Passwort (auch bekannt als Pre-Shared Key oder PSK) benötigt. Zeichen, die in die Eingabebox getippt werden, erscheinen aus Sicherheitsgründen als Sternchen.

Abbildung 3-25. Verbindungsaufbau mit WPA2



Die Netzwerkkonfiguration wird fortgesetzt, nachdem das drahtlose Netzwerk und die Verbindungsinformationen eingegeben wurden.

3.9.2.2. Konfiguration des IPv4-Netzwerks

Wählen Sie, ob Sie ein IPv4-Netzwerk verwenden möchten. Dies ist der am häufigsten vorkommende Typ einer Netzwerkverbindung.

Abbildung 3-26. Auswahl von IPv4

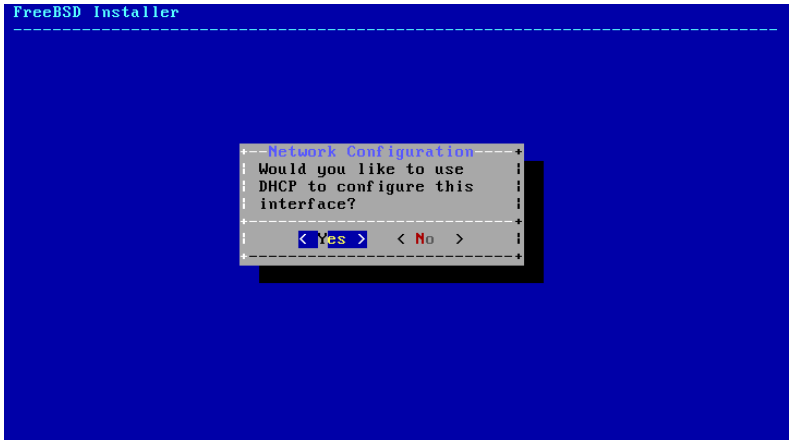
Es gibt zwei Arten, ein IPv4-Netzwerk zu konfigurieren. *DHCP* wird automatisch die Netzwerkschnittstelle richtig konfigurieren und sollte als bevorzugte Methode verwendet werden. *Statische* Konfiguration erfordert die manuelle Eingabe von Netzwerkinformationen.

> **Anmerkung:** Geben Sie keine zufällig gewählten Netzwerkinformationen ein, da dies nicht funktionieren wird. Holen Sie sich die in Abschnitt 3.3.3 gezeigten Informationen von Ihrem Netzwerkadministrator oder Serviceprovider.

3.9.2.2.1. Netzwerkkonfiguration von IPv4 mittels DHCP

Falls ein DHCP-Server zur Verfügung steht, wählen Sie [Yes], um die Netzwerkschnittstelle automatisch einrichten zu lassen.

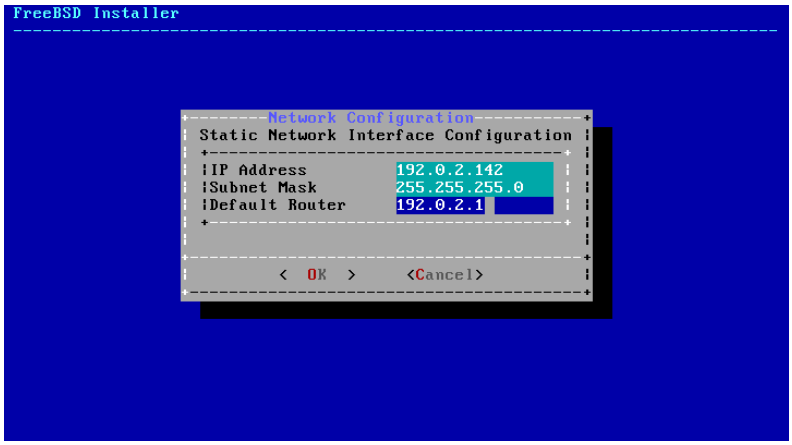
Abbildung 3-27. Auswählen der IPv4-Konfiguration über DHCP



3.9.2.2.2. Statische IPv4-Netzwerkkonfiguration

Statische Konfiguration der Netzwerkschnittstelle erfordert die die Eingabe einiger IPv4-Informationen.

Abbildung 3-28. Statische IPv4-Konfiguration



- `IP-Adresse` - Die manuell festgelegte IPv4-Adresse, welche diesem Computer zugewiesen werden soll. Diese Adresse muss eindeutig sein und darf von keinem anderen Gerät im lokalen Netzwerk bereits verwendet werden.
- `Subnetzmaske` - Die Subnetzmaske, die im lokalen Netzwerk Verwendung findet. Typischerweise ist dies `255.255.255.0`.
- `Defaultrouter` - Die IP-Adresse des Defaultrouters in diesem Netzwerk. Normalerweise ist das die Adresse des Routers oder einer anderen Netzwerkkomponente, die das lokale Netzwerk mit dem Internet verbindet. Auch bekannt als das *Default Gateway*.

3.9.2.3. Konfiguration des IPv6-Netzwerks

IPv6 ist eine neuere Methode der Netzwerkkonfiguration. Falls IPv6 verfügbar ist und verwendet werden soll, wählen Sie [Yes] aus.

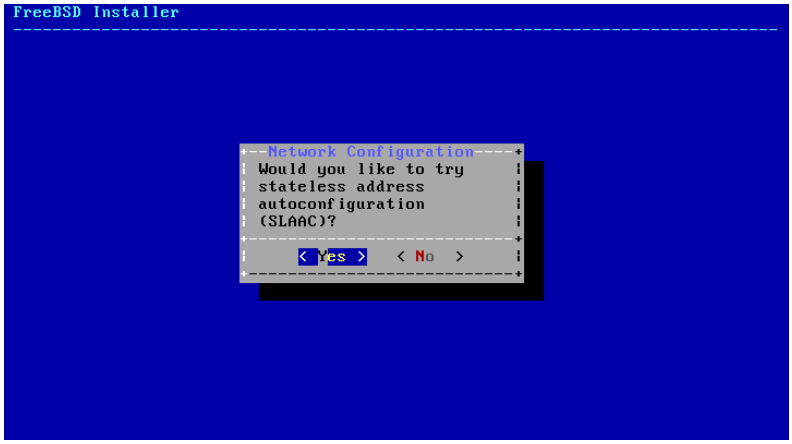
Abbildung 3-29. Auswahl von IPv6

IPv6 besitzt ebenfalls zwei Arten der Konfiguration. *SLAAC*, oder *StateLess Address AutoConfiguration*, wird die Netzwerkschnittstelle automatisch richtig konfigurieren. *Statische* Konfiguration verlangt die manuelle Eingabe von Netzwerkinformationen.

3.9.2.3.1. IPv6 Stateless Address Autoconfiguration

SLAAC erlaubt es einer IPv6-Netzwerkkomponente, die Information zur automatischen Konfiguration von einem lokalen Router abzufragen. Lesen Sie RFC4862 (http://tools.ietf.org/html/rfc4862) für weitere Informationen.

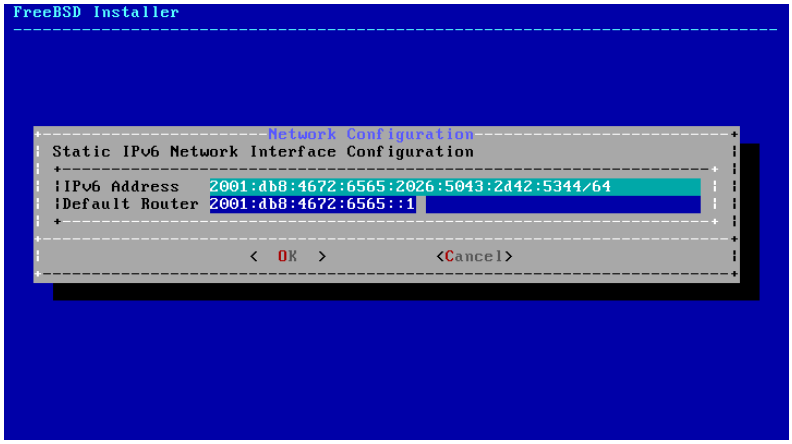
Abbildung 3-30. Auswahl der IPv6 SLAAC-Konfiguration



3.9.2.3.2. Statische IPv6-Netzwerkkonfiguration

Statische Konfiguration der Netzwerkschnittstelle benötigt die Eingabe von IPv6-Konfigurationsinformationen.

Abbildung 3-31. Statische IPv6-Konfiguration

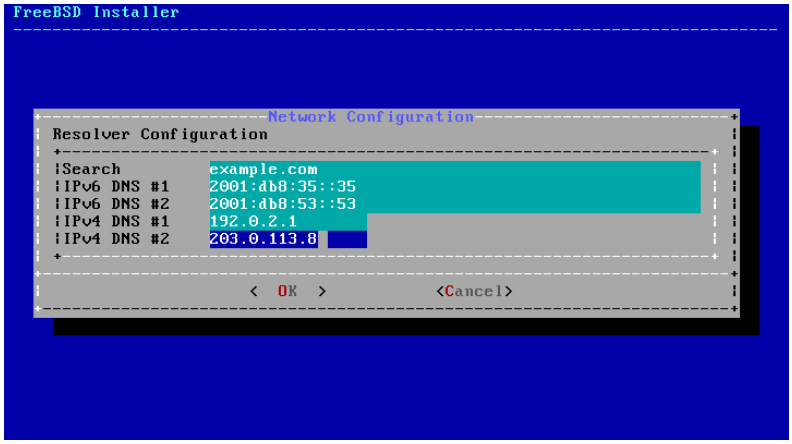


- `IPv6-Adresse` - The manuell zugewiesene IP-Adresse, welche dem Computer zugeteilt werden soll. Diese Adresse muss eindeutig sein und nicht bereits von einer anderen Netzwerkkomponente im lokalen Netzwerk verwendet werden.
- `Defaultrouter` - Die IPv6-Adresse des Defaultrouters in diesem Netzwerk. Normalerweise ist dies die Adresse des Routers oder einer anderen Netzwerkkomponente, welche das lokale Netz mit dem Internet verbindet. Auch bekannt als *Default Gateway*.

3.9.2.4. DNS-Konfiguration

Der *Domain Name System* (oder auch *DNS*) Auflöser wandelt Hostnamen von und zu Netzwerkadressen um. Falls DHCP oder SLAAC verwendet wurde, um die Netzwerkschnittstelle zu konfigurieren, ist die Konfiguration für den Auflöser möglicherweise bereits vorhanden. Andernfalls geben Sie den lokalen Netzwerkdomänennamen in das Suchfeld ein. DNS #1 und DNS #2 sind die IP-Adressen der lokalen DNS-Server. Zumindest ein DNS-Server wird benötigt.

Abbildung 3-32. DNS-Konfiguration



3.9.3. Setzen der Zeitzone

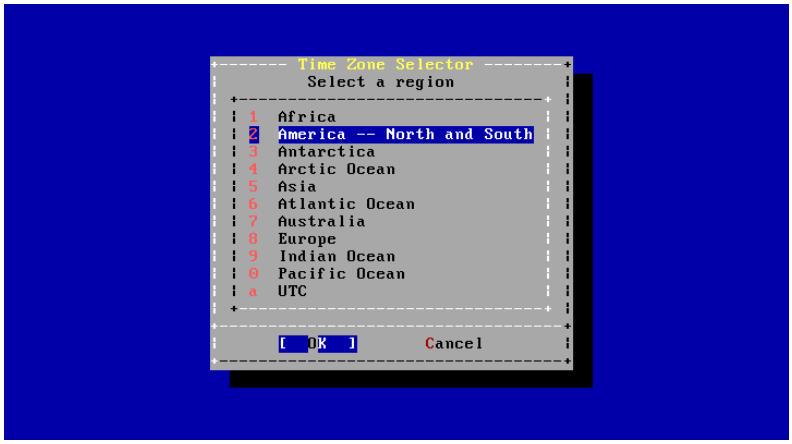
Das Setzen der Zeitzone für Ihre Maschine erlaubt es, diese auf regionale Zeitveränderungen hin anzupassen und um andere zeitzonenbezogene Funktionen richtig durchzuführen.

Das hier Beispiel gezeigte Beispiel bezieht sich auf einen Rechner in der östlichen Zeitzone der Vereinigten Staaten. Ihre Auswahl wird von Ihrer geographischen Position davon abweichen.

Abbildung 3-33. Lokale oder UTC-Zeit

Wählen Sie [Yes] oder [No], abhängig davon, wie die Rechneruhr konfiguriert ist und drücken Sie dann **Enter**. Wenn Sie nicht wissen, ob Ihr System UTC oder lokale Zeit verwendet, wählen Sie [No], um die am häufigsten verwendete lokale Zeit zu setzen.

Abbildung 3-34. Das Gebiet auswählen

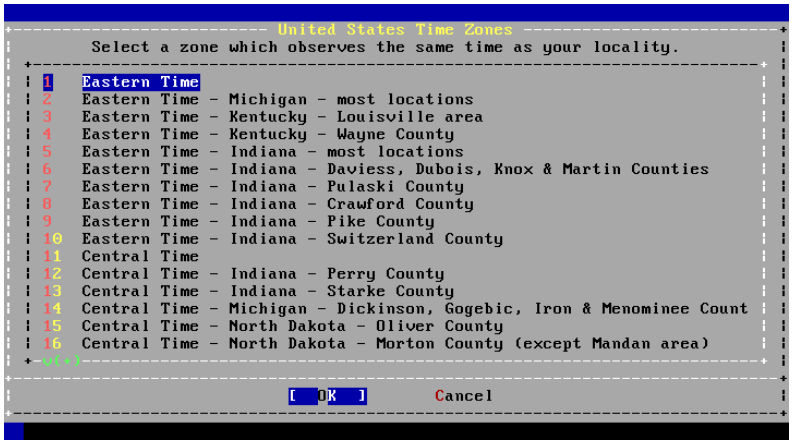


Das passende Gebiet wird durch die Pfeiltasten und das anschliessende druücken von **Enter** gewählt.

Abbildung 3-35. Das Land auswählen

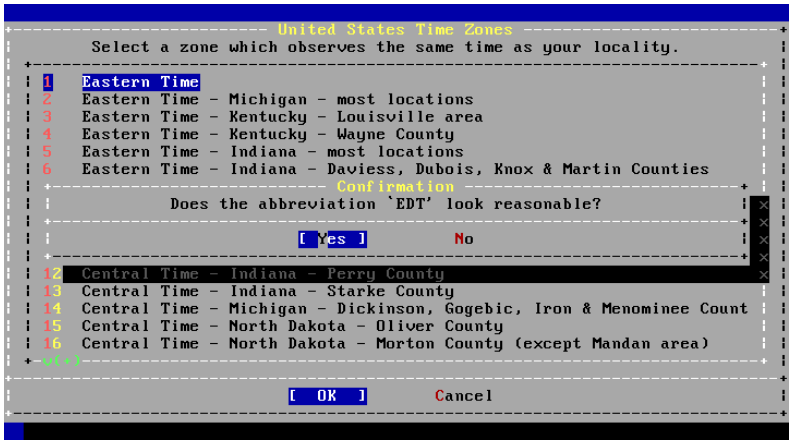
Wählen Sie das zutreffende Land mit den Pfeiltasten und durch anschliessendes drücken von **Enter** aus.

Abbildung 3-36. Wählen einer Zeitzone



Die passende Zeitzone wird durch die Pfeiltasten und anschliessendes drücken von **Enter** ausgewählt.

Abbildung 3-37. Bestätigen der Zeitzone

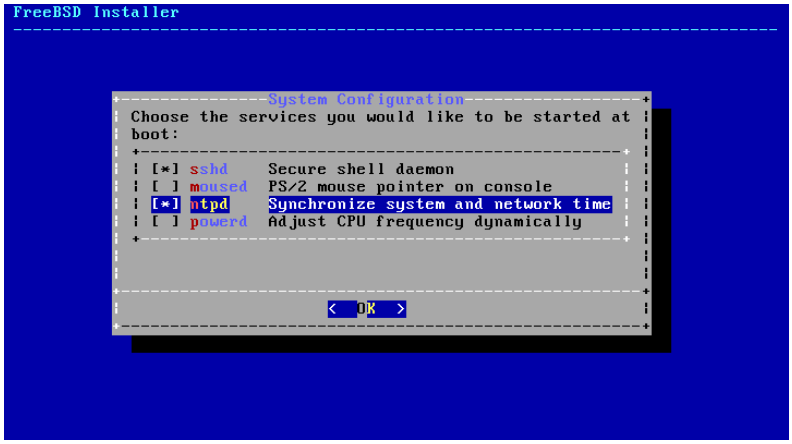


Bestätigen Sie, dass die Abkürzung für die Zeitzone richtig ist. Wenn Ihnen diese richtig erscheint, drücken Sie **Enter**, um mit dem Rest der Konfiguration nach der Installation fortzufahren.

3.9.4. Zu aktivierende Dienste auswählen

Zusätzliche Systemdienste, die zur Startzeit aktiviert werden sollen, können eingeschaltet werden. All diese Dienste sind optional.

Abbildung 3-38. Auswahl zusätzlicher Dienste



Zusätzliche Dienste

- `sshd` - Secure Shell (SSH)-Dienst für sicheren Fernzugriff.
- `moused` - Sorgt für Mausunterstützung innerhalb der Systemkonsole.
- `ntpd` - Network Time Protocol (NTP)-Dienst zur automatischen Uhrzeitsynchronisation.
- `powerd` - Systemleistungskontrollwerkzeug zur Leistungsregelung und für Stromsparfunktionen.

3.9.5. Absturzaufzeichnung aktivieren

bsdinstall wird Sie fragen, ob die Absturzaufzeichnung auf dem Zielsystem aktiviert werden soll. Die Aktivierung von Absturzaufzeichnungen kann sehr nützlich sein, um Systemfehler aufzuspüren, deswegen wird Anwendern empfohlen, diese so oft wie möglich einzusetzen. Wählen Sie [Yes], um Absturzaufzeichnungen zu aktivieren oder [No], um ohne die Aufzeichnung von Abstürzen fortzufahren.

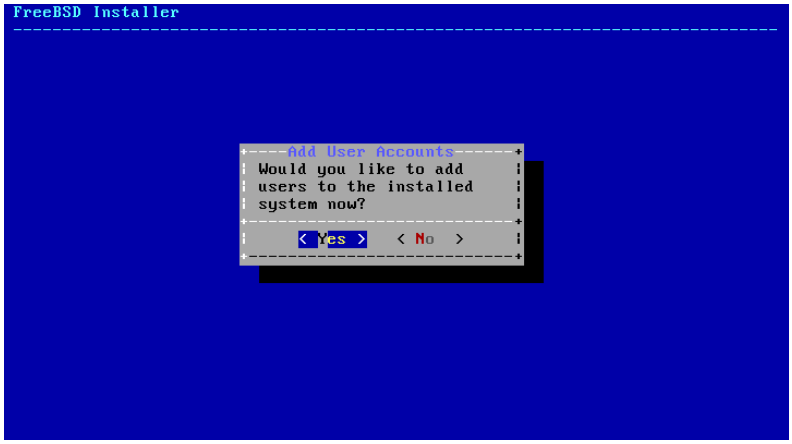
Abbildung 3-39. Aktivierung der Absturzaufzeichnung

3.9.6. Benutzer hinzufügen

Das hinzufügen von mindestens einem Benutzer während der Installaton erlaubt das Benutzen des Systems ohne als `root`-Benutzer angemeldet zu sein. Wenn man als `root` angemeldet ist, gibt es so gut wie keine Beschränkungen oder Schutz vor dem, was man tun kann. Anmeldenals normaler Benutzer ist daher sicherer und bietet mehr Schutz.

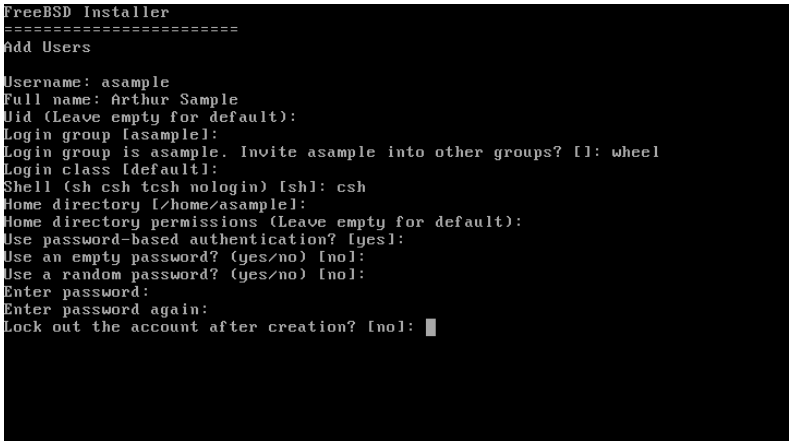
Wählen Sie [Yes], um neue Benutzer hinzuzufügen.

Abbildung 3-40. Benutzerkonten hinzufügen



Geben Sie die nötigen Informationen für den Benutzer ein, der dem System hinzugefügt werden soll.

Abbildung 3-41. Benutzerinformationen eingeben



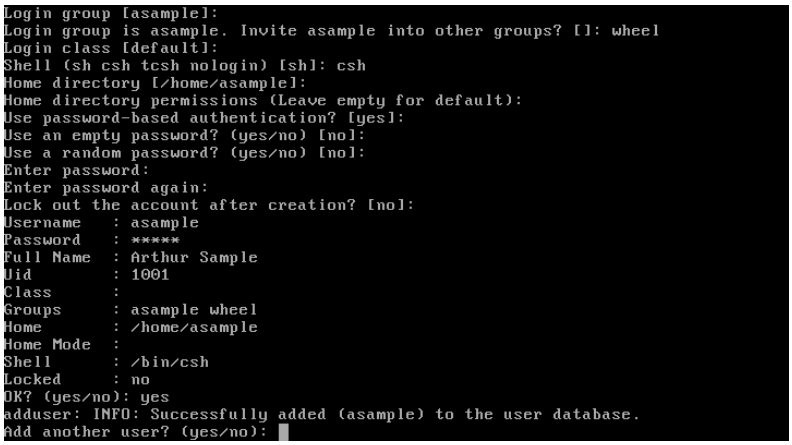
Benutzerinformationen

- `Username` - Der Name des Benutzers, den man zur Anmeldung eingeben muss. Typischerweise der erste Buchstabe des Vornamens, gefolgt vom Nachnamen.
- `Full name` - Der volle Name des Benutzers.
- `Uid` - User ID. Normalerweise wird dieses Feld leer gelassen, so dass das System einen Wert vergibt.

- `Login group` - Die Benutzergruppe. Normalerweise bleibt dieses Feld leer, um die Standardgruppe zu akzeptieren.
- `Invite` *`user`* `into other groups?` - Zusätzliche Gruppen zu denen der Benutzer als Mitglied hinzugefügt werden soll.
- `Login class` - In der Regel bleibt dieses Feld leer.
- `Shell` - Die interaktive Shell für diesen Benutzer. In diesem Beispiel wurde csh(1) ausgewält.
- `Home directory` - Das Heimatverzeichnis des Benutzers. Die Vorgabe ist für gewöhnlich richtig.
- `Home directory permissions` - Zugriffsrechte auf das Heimatverzeichnis des Benutzers. Die Vorgabe ist normalerweise die passende.
- `Use password-based authentication?` Normalerweise "yes".
- `Use an empty password?` - Normalerweise "no".
- `Use a random password?` - Normalerweise "no".
- `Enter password` - Das Passwort f¨r diesem Benutzer. Eingegebene Zeichen werden nicht am Bildschirm angezeigt.
- `Enter password again` - Das Passwort muss zur Überprüfung erneut eingegeben werden.
- `Lock out the account after creation?` - Normalerweise "no".

Nachdem alles eingegeben wurde, wird eine Zusammenfassung angezeigt und das System fragt Sie, dies so korrekt ist. Falls ein Eingabefehler gemacht wurde, geben Sie `no` ein und versuchen es erneut. Falls alles in Ordnung ist, drücken Sie `yes`, um den neuen Benutzer anzulegen.

Abbildung 3-42. Verlassen der Benutzer- und Gruppenverwaltung



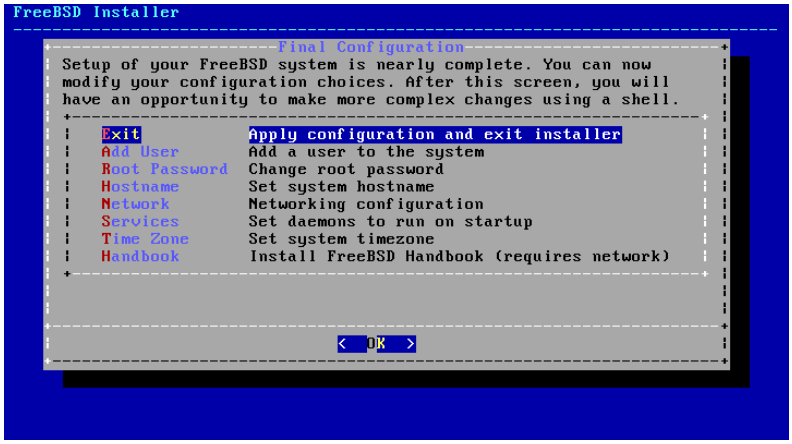
Falls es mehr Benutzer hinzuzufügen gibt, beantworten Sie die Frage "Add another user?" mit `yes`. Geben Sie `no` ein, wird das hinzufügen von Benutzern beendet und die Installation fortgesetzt.

Für weitere Informationen zum hinzufügen von Benutzern und deren Verwaltung, lesen Sie Kapitel 14.

3.9.7. Letzte Konfigurationsschritte

Nachdem alles installiert und konfiguriert wurde, bekommen Sie noch eine letzte Chance, um Einstellungen zu verändern.

Abbildung 3-43. Letzte Schritte der Konfiguration



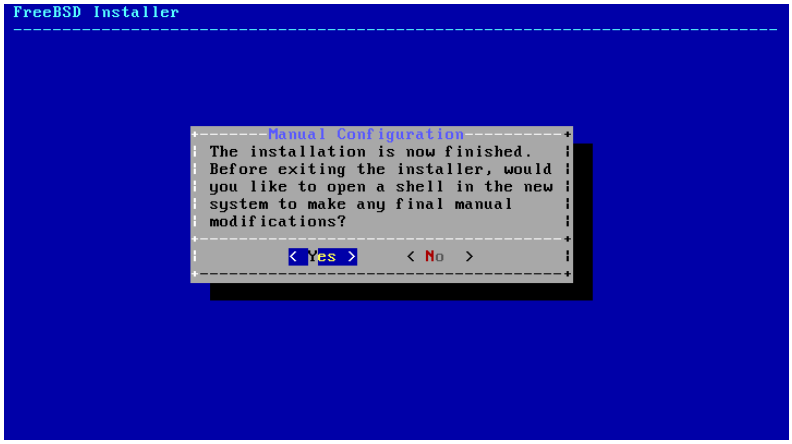
Verwenden Sie dieses Menü, um noch letzte Änderungen oder zusätzliche Konfigurationen vor dem Abschliessen der Installation zu tätigen.

Letzte Konfigurationsoptionen

- `Add User` - Beschrieben in Abschnitt 3.9.6.
- `Root Password` - Beschrieben in Abschnitt 3.9.1.
- `Hostname` - Beschrieben in Abschnitt 3.5.2.
- `Network` - Beschrieben in Abschnitt 3.9.2.
- `Services` - Beschrieben in Abschnitt 3.9.4.
- `Time Zone` - Beschrieben in Abschnitt 3.9.3.
- `Handbook` - Herunterladen und installieren des FreeBSD Handbuchs (welches Sie gerade lesen).

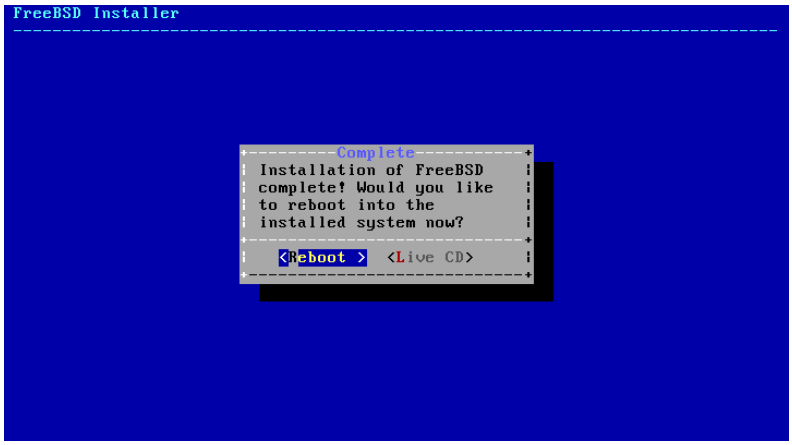
Nachdem die lezten Konfigurationsschritte beendet sind, wählen Sie Exit, um die Installation zu verlassen.

Abbildung 3-44. Manuelle Konfiguration



bsdinstall wird nach zusätzlichen Konfigurationen, die noch zu tätigen sind, fragen, bevor in das neue System gebootet wird. Wählen Sie [Yes], um in eine Shell innerhalb des neuen Systems zu wechseln oder [No], um mit dem letzten Schritt der Installation zu beginnen.

Abbildung 3-45. Die Installation vervollständigen



Wenn weitere Konfigurationen oder besondere Einstellungen benötigt werden, kann durch auswählen von [Live CD] das Installationsmedium im Live CD Modus gestartet werden.

Wenn die Installation vollständig ist, wählen Sie [Reboot], um den Computer neu zu starten und das neu installierte FreeBSD-System zu booten. Vergessen Sie nicht, die FreeBSD Installations-CD, -DVD oder den USB-Stick zu entfernen, oder der Computer wird erneut davon starten.

3.9.8. FreeBSD starten und herunterfahren

3.9.8.1. FreeBSD/i386 starten

Wenn FreeBSD startet, werden viele Informationsmeldungen ausgegeben. Die meisten davon werden aus dem

Bildschirm verschwinden, das ist normal. Nachdem das System den Startvorgang abgeschlossen hat, wird eine Anmeldeaufforderungen angezeigt. Um Nachrichten, die aus dem Bildschirm gelaufen sind, zu sehen, aktivieren Sie durch drücken von **Scroll-Lock** den *scroll-back buffer*. Die Tasten **PgUp**, **PgDn** und die Pfeiltasten dienen zur Navigation durch die Nachrichten. Durch erneutes drücken von **Scroll-Lock** wird der Bildschirm wieder entsperrt und kehrt zur normalen Anzeige zurück.

Am `login:`-Bildschirm geben Sie den Benutzernamen ein, den Sie während der Installation angelegt haben, in diesem Fall ist das `asample`. Vermeiden Sie die Anmeldung als `root`, ausser wenn es wirklich notwendig ist.

Der oben beschriebene scroll-back buffer ist in der Grösse beschränkt, somit werden vielleicht nicht alle Nachrichten sichtbar sein. Nach dem Anmelden können die meisten davon aus der Kommandozeile aus durch eingabe von `dmesg | less` betrachtet werden. Durch drücken von **q** kehren Sie wieder zur Kommandozeile zurück.

Typische Startmeldungen (Versionsinformationen wurden hier weggelassen):

```
Copyright (c) 1992-2011 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
        The Regents of the University of California. All rights reserved.
FreeBSD is a registered trademark of The FreeBSD Foundation.

    root@farrell.cse.buffalo.edu:/usr/obj/usr/src/sys/GENERIC amd64
CPU: Intel(R) Core(TM)2 Duo CPU     E8400  @ 3.00GHz (3007.77-MHz K8-class CPU)
  Origin = "GenuineIntel"  Id = 0x10676  Family = 6  Model = 17  Stepping = 6
  Features=0x783fbff<FPU,VME,DE,PSE,TSC,MSR,PAE,MCE,CX8,APIC,SEP,MTRR,PGE,MCA,CMOV,PAT,PSE36,MMX,
  Features2=0x209<SSE3,MON,SSSE3>
  AMD Features=0x20100800<SYSCALL,NX,LM>
  AMD Features2=0x1<LAHF>
real memory  = 536805376 (511 MB)
avail memory = 491819008 (469 MB)
Event timer "LAPIC" quality 400
ACPI APIC Table: <VBOX   VBOXAPIC>
ioapic0: Changing APIC ID to 1
ioapic0 <Version 1.1> irqs 0-23 on motherboard
kbd1 at kbdmux0
acpi0: <VBOX VBOXXSDT> on motherboard
acpi0: Power Button (fixed)
acpi0: Sleep Button (fixed)
Timecounter "ACPI-fast" frequency 3579545 Hz quality 900
acpi_timer0: <32-bit timer at 3.579545MHz> port 0x4008-0x400b on acpi0
cpu0: <ACPI CPU> on acpi0
pcib0: <ACPI Host-PCI bridge> port 0xcf8-0xcff on acpi0
pci0: <ACPI PCI bus> on pcib0
isab0: <PCI-ISA bridge> at device 1.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <Intel PIIX4 UDMA33 controller> port 0x1f0-0x1f7,0x3f6,0x170-0x177,0x376,0xd000-0xd00f a
ata0: <ATA channel 0> on atapci0
ata1: <ATA channel 1> on atapci0
vgapci0: <VGA-compatible display> mem 0xe0000000-0xe0ffffff irq 18 at device 2.0 on pci0
em0: <Intel(R) PRO/1000 Legacy Network Connection 1.0.3> port 0xd010-0xd017 mem 0xf0000000-0xf001
em0: Ethernet address: 08:00:27:9f:e0:92
pci0: <base peripheral> at device 4.0 (no driver attached)
pcm0: <Intel ICH (82801AA)> port 0xd100-0xd1ff,0xd200-0xd23f irq 21 at device 5.0 on pci0
pcm0: <SigmaTel STAC9700/83/84 AC97 Codec>
ohci0: <OHCI (generic) USB controller> mem 0xf0804000-0xf0804fff irq 22 at device 6.0 on pci0
```

```
usbus0: <OHCI (generic) USB controller> on ohci0
pci0: <bridge> at device 7.0 (no driver attached)
acpi_acad0: <AC Adapter> on acpi0
atkbdc0: <Keyboard controller (i8042)> port 0x60,0x64 irq 1 on acpi0
atkbd0: <AT Keyboard> irq 1 on atkbdc0
kbd0 at atkbd0
atkbd0: [GIANT-LOCKED]
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: model IntelliMouse Explorer, device ID 4
attimer0: <AT timer> port 0x40-0x43,0x50-0x53 on acpi0
Timecounter "i8254" frequency 1193182 Hz quality 0
Event timer "i8254" frequency 1193182 Hz quality 100
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
atrtc0: <AT realtime clock> at port 0x70 irq 8 on isa0
Event timer "RTC" frequency 32768 Hz quality 0
ppc0: cannot reserve I/O port range
Timecounters tick every 10.000 msec
pcm0: measured ac97 link rate at 485193 Hz
em0: link state changed to UP
usbus0: 12Mbps Full Speed USB v1.0
ugen0.1: <Apple> at usbus0
uhub0: <Apple OHCI root HUB, class 9/0, rev 1.00/1.00, addr 1> on usbus0
cd0 at ata1 bus 0 scbus1 target 0 lun 0
cd0: <VBOX CD-ROM 1.0> Removable CD-ROM SCSI-0 device
cd0: 33.300MB/s transfers (UDMA2, ATAPI 12bytes, PIO 65534bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not present
ada0 at ata0 bus 0 scbus0 target 0 lun 0
ada0: <VBOX HARDDISK 1.0> ATA-6 device
ada0: 33.300MB/s transfers (UDMA2, PIO 65536bytes)
ada0: 12546MB (25694208 512 byte sectors: 16H 63S/T 16383C)
ada0: Previously was known as ad0
Timecounter "TSC" frequency 3007772192 Hz quality 800
Root mount waiting for: usbus0
uhub0: 8 ports with 8 removable, self powered
Trying to mount root from ufs:/dev/ada0p2 [rw]...
Setting hostuuid: 1848d7bf-e6a4-4ed4-b782-bd3f1685d551.
Setting hostid: 0xa03479b2.
Entropy harvesting: interrupts ethernet point_to_point kickstart.
Starting file system checks:
/dev/ada0p2: FILE SYSTEM CLEAN; SKIPPING CHECKS
/dev/ada0p2: clean, 2620402 free (714 frags, 327461 blocks, 0.0% fragmentation)
Mounting local file systems:.
vboxguest0 port 0xd020-0xd03f mem 0xf0400000-0xf07fffff,0xf0800000-0xf0803fff irq 20 at device 4.0
vboxguest: loaded successfully
Setting hostname: machine3.example.com.
Starting Network: lo0 em0.
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
```

```
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=9b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM>
        ether 08:00:27:9f:e0:92
        nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
        media: Ethernet autoselect (1000baseT <full-duplex>)
        status: active
Starting devd.
Starting Network: usbus0.
DHCPREQUEST on em0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 192.168.1.142 -- renewal in 43200 seconds.
add net ::ffff:0.0.0.0: gateway ::1
add net ::0.0.0.0: gateway ::1
add net fe80::: gateway ::1
add net ff02::: gateway ::1
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/local/lib
32-bit compatibility ldconfig path: /usr/lib32
Creating and/or trimming log files.
Starting syslogd.
No core dumps found.
Clearing /tmp (X related).
Updating motd:.
Configuring syscons: blanktime.
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|       o         |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|       ..     . .|
|      o  .   . + |
|     . ..   . E .|
|    . .  o o . . |
```

```
|     +  S = .    |
|    +  . = o     |
|     +  . * .    |
|    . .  o .     |
|      .o. .      |
+-----------------+
Starting sshd.
Starting cron.
Starting background file system checks in 60 seconds.

Thu Oct  6 19:15:31 MDT 2011

FreeBSD/amd64 (machine3.example.com) (ttyv0)

login:
```

Das Generieren der RSA- und DSA-Schlüssel kann auf langsameren Rechnern einige Zeit benötigen. Dies geschieht nur während der Startphase einer neuen Installation und auch nur, wenn **sshd** zum automatischen Start gesetzt ist. Die nachfolgenden Startvorgänge werden schneller sein.

FreeBSD installiert standardmässig keine graphische Umgebung, jedoch stehen viele zur Verfügung. Lesen Sie Kapitel 6 für weitere Informationen.

3.9.9. FreeBSD herunterfahren

Das korrekte herunterfahren eines FreeBSD-Computers hilft, beugt dem Datenverlust vor und schützt sogar die Hardware vor Schäden. Schalten Sie nicht einfach den Strom ab. Wenn der Benutzer ein Mitglied der `wheel`-Gruppe ist, können Sie zum Superuser durch die Eingabe von `su` und der anschliessenden Eingabe des Passworts von `root` werden. Andernfalls melden Sie sich mit `root` an und verwenden den Befehl `shutdown -p now`. Das System wird jetzt sauber heruntergefahren und den Rechner ausschalten.

Die **Ctrl**+**Alt**+**Del** Kombination kann verwendet werden, um das System neu zu starten, jedoch wird dies nicht während des normalen Betriebs empfohlen.

3.10. Fehlerbehebung

Der folgende Abschnitt behandelt einfache Fehlerbehebungen für die Installation, wie beispielsweise häufig auftretende Fehler, die von Anwendern berichtet wurden.

3.10.1. Was man tun sollte, wenn etwas schiefgeht

Wegen verschiedener Limitierungen der PC-Architektur ist es unmöglich dass die Geräteerkennung 100% verlässlich funktioniert. Jedoch gibt es ein paar Dinge, die man tun kann, wenn es fehlschlägt.

Überprüfen Sie das Dokument Hardware Notes (http://www.FreeBSD.org/releases/index.html) nach Ihrer Version von FreeBSD, um sicher zu stellen, dass Ihre Hardware auch unterstützt wird.

Wenn Ihre Hardware unterstützt wird und Sie immer noch Abstürze oder andere Probleme erleben, müssen Sie einen eigenen Kernel bauen. Das wird Ihnen erlauben, Unterstützung für Geräte, die im `GENERIC`-Kernel nicht vorhanden

sind, hinzuzufügen. Der Kernel auf den Bootmedien ist mit der Annahme konfiguriert, dass die Hardwaregeräte sich in Ihren Fabrikeinstellungen in Bezug auf IRQs, I/O-Adressen und DMA-Kanälen befinden. Wenn Ihre Hardware neu konfiguriert wurde, werden Sie möglicherweise die Konfiguration des Kernels bearbeiten und diesen neu erstellen müssen, um FreeBSD mitzuteilen, wo es gewisse Dinge zu finden hat.

Es ist auch möglich, dass ein fehlerhaft erkanntes Gerät die Erkennung eines vorhandenen, späteren Geräts ebenfalls fehlschlagen lässt. In diesem Fall sollte die Erkennung des fehlerhaften Gerätetreibers deaktiviert werden.

> **Anmerkung:** Manche Installationsprobleme können Aktualisierung der Firmware auf verschiedenen Hardwarekomponenten verhindert oder verringert werden, meistens am Mainboard. Mit Mainboard-Firmware ist für gewöhnlich das BIOS gemeint. Die meisten Mainboard- und Computerhersteller haben eine Webseite mit Aktualisierungen und Informationen zur Durchführung.
>
> Hersteller raten meist von einer Aktualisierung des Mainboard-BIOS ab, ausser es gibt einen guten Grund dafür, wie beispielsweise eine kritische Aktualisierung. Der Aktualisierungsvorgang *kann* schiefgehen, was das BIOS unvollständig macht und den Computer nicht mehr starten lässt.

3.10.2. Fragen und Antworten zur Fehlerbehebung

1. Mein System hängt während die Geräteerkennung beim Starten durchgeführt wird oder verhält sich merkwürdig während der Installation.

FreeBSD macht starken Gebrauch vom ACPI-Dienst des Systems auf den i386-, amd64-, and ia64-Platformen, um den System bei der Konfiguration während des Startvorgangs zu helfen. Leider existieren immer noch Fehler im ACPI-Treiber, in den Mainboards und der BIOS-Firmware. ACPI kann durch setzen der Einstellung `hint.acpi.0.disabled` im dritten Teil des Bootloaders deaktiviert werden:

```
set hint.acpi.0.disabled="1"
```

Dies wird nach jedem Neustart des Systems wieder zurückgesetzt, also ist es notwendig, die Zeile `hint.acpi.0.disabled="1"` zu der Datei `/boot/loader.conf` hinzuzufügen. Weitere Informationen über den Bootloader lassen sich in Abschnitt 13.1 nachlesen.

Kapitel 4. Grundlagen des UNIX Betriebssystems

Umgeschrieben von Chris Shumway. Übersetzt von Uwe Pierau.

4.1. Übersicht

Das folgende Kapitel umfasst die grundlegenden Kommandos und Funktionsweisen des Betriebssystems FreeBSD. Viel von dem folgenden Material gilt auch für jedes andere UNIX-artige System. Falls Sie mit dem Material schon vertraut sind, können Sie dieses Kapitel überlesen. Wenn FreeBSD neu für Sie ist, dann sollten Sie dieses Kapitel auf jeden Fall aufmerksam lesen.

Dieser Abschnitt behandelt die folgenden Themen:

- virtuelle Konsolen,
- Zugriffsrechte unter UNIX sowie Datei-Flags unter FreeBSD,
- Zugriffskontrolllisten für Dateisysteme,
- die Verzeichnisstruktur von FreeBSD,
- Organisation von Dateisystemen unter FreeBSD,
- Ein- und Abhängen von Dateisystemen,
- Prozesse, Dämonen und Signale,
- Shells und die Login-Umgebung,
- Texteditoren,
- Geräte und Gerätedateien,
- Binärformate unter FreeBSD und
- wie Sie in den Manualpages nach weiteren Informationen suchen können.

4.2. Virtuelle Konsolen und Terminals

Sie können FreeBSD mit einem Terminal benutzen, der nur Text darstellen kann. Wenn Sie FreeBSD auf diese Weise benutzen, stehen Ihnen alle Möglichkeiten eines UNIX Betriebssystems zur Verfügung. Dieser Abschnitt beschreibt was Terminals und Konsolen sind und wie sie unter FreeBSD eingesetzt werden.

4.2.1. Die Konsole

Wenn Ihr FreeBSD-System ohne eine graphische Benutzeroberfläche startet, wird am Ende des Systemstarts, nachdem die Startskripten gelaufen sind, ein Anmeldeprompt ausgegeben. Die letzten Startmeldungen sollten ähnlich wie die Folgenden aussehen:

```
Additional ABI support:.
Local package initialization:.
```

```
Additional TCP options:.

Fri Sep 20 13:01:06 EEST 2002

FreeBSD/i386 (pc3.example.org) (ttyv0)

login:
```

Beachten Sie die letzten beiden Zeilen der Ausgabe, die vorletzte lautet:

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

Diese Zeile enthält einige Informationen über das gerade gestartete System. Die Ausgabe stammt von der FreeBSD-Konsole einer Maschine mit einem Intel oder Intel-kompatiblen Prozessor der x86-Architektur[1]. Der Name des Systems (jedes UNIX System besitzt einen Namen) ist `pc3.example.org` und die Ausgabe stammt von der Systemkonsole, dem Terminal `ttyv0`.

Das Ende der Ausgabe ist immer die Aufforderung zur Eingabe eines Benutzernamens:

```
login:
```

Der Anmeldevorgang wird im nächsten Abschnitt erläutert.

4.2.2. Der Anmeldevorgang

FreeBSD ist ein Mehrbenutzersystem, das Multitasking unterstützt. Das heißt mehrere Benutzer können gleichzeitig viele Programme auf einem System laufen lassen.

Jedes Mehrbenutzersystem muss die Benutzer voneinander unterscheiden können. Bei FreeBSD und allen anderen UNIX-artigen Betriebssystemen wird dies dadurch erreicht, dass sich die Benutzer anmelden müssen, bevor sie Programme laufen lassen können. Jeder Benutzer besitzt einen eindeutigen Namen (den Account) und ein dazugehörendes Passwort, die beide bei der Anmeldung abgefragt werden.

Nachdem FreeBSD gestartet ist und die Startskripten[2], gelaufen sind, erscheint eine Aufforderung zur Eingabe des Benutzernamens:

```
login:
```

Wenn Ihr Benutzername beispielsweise `john` ist, geben Sie jetzt `john` gefolgt von **Enter** ein. Sie sollten dann eine Aufforderung zur Eingabe des Passworts erhalten:

```
login: john
Password:
```

Geben Sie jetzt das Passwort von `john` gefolgt von **Enter** ein. Das Passwort wird aus Sicherheitsgründen nicht auf dem Bildschirm angezeigt.

Wenn Sie das richtige Passwort eingegeben haben, sind Sie am System angemeldet und können nun alle verfügbaren Kommandos absetzen.

Anmgemeldet sind Sie, wenn Sie die Tagesmeldungen (*message of today*) gefolgt von einer Eingabeaufforderung (dem Zeichen `#`, `$` oder `%`) gesehen haben.

4.2.3. Virtuelle Konsolen

Da FreeBSD mehrere Programme gleichzeitig laufen lassen kann, ist eine einzige Konsole, an der Kommandos abgesetzt werden können, zu wenig. Abhilfe schaffen virtuelle Konsolen, die mehrere Konsolen zur Verfügung stellen.

Die Anzahl der virtuellen Konsolen unter FreeBSD können Sie einstellen. Zwischen den einzelnen Konsolen können Sie mit speziellen Tastenkombinationen wechseln. Jede Konsole verfügt über einen eigenen Ausgabekanal und FreeBSD ordnet die Tastatureingaben und Monitorausgaben der richtigen Konsole zu, wenn Sie zwischen den Konsolen wechseln.

Zum Umschalten der Konsolen stellt FreeBSD spezielle Tastenkombinationen bereit[3]. Benutzen Sie **Alt-F1**, **Alt-F2** bis **Alt-F8**, um zwischen den verschiedenen Konsolen umzuschalten.

Wenn Sie zu einer anderen Konsole wechseln, sichert FreeBSD den Bildschirminhalt und gibt den Bildschirminhalt der neuen Konsole aus. Dies erzeugt die Illusion mehrerer Bildschirme und Tastaturen, an denen Sie Kommandos absetzen können. Wenn eine Konsole nicht sichtbar ist, weil Sie auf eine andere Konsole gewechselt haben, laufen die dort abgesetzten Kommandos weiter.

4.2.4. `/etc/ttys`

In der Voreinstellung stehen unter FreeBSD acht virtuelle Konsolen zur Verfügung, deren Anzahl Sie leicht erhöhen oder verringern können. Die Anzahl und Art der Konsolen wird in `/etc/ttys` eingestellt.

Jede Zeile in `/etc/ttys`, die nicht mit # anfängt, konfiguriert einen Terminal oder eine virtuelle Konsole. In der Voreinstellung werden in dieser Datei neun virtuelle Konsolen definiert, von denen acht aktiviert sind. Die Konsolen sind in den Zeilen, die mit `ttyv` beginnen, definiert:

```
# name  getty                           type    status          comments
#
ttyv0   "/usr/libexec/getty Pc"         cons25  on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv2   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv3   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv4   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv5   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv6   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv7   "/usr/libexec/getty Pc"         cons25  on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

Die Hilfeseite ttys(5) enthält eine ausführliche Beschreibung der Spalten dieser Datei und der Optionen, die Sie zum Konfigurieren der virtuellen Konsolen benutzen können.

4.2.5. Die Konsole im Single-User-Modus

Eine eingehende Beschreibung des Single-User-Modus finden Sie in Abschnitt 13.6.2. Im Single-User-Modus steht Ihnen nur *eine* Konsole zur Verfügung. Die Definition dieser Konsole befindet sich ebenfalls in `/etc/ttys`. Suchen Sie nach einer Zeile, die mit `console` beginnt:

```
# name  getty                           type    status          comments
#
```

```
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off secure
```

> **Anmerkung:** In der Zeile, die mit `console` beginnt, können Sie `secure` durch `insecure` ersetzen. Wenn Sie danach in den Single-User-Modus booten, verlangt das System ebenfalls die Eingabe des `root`-Passworts.
>
> *Setzen Sie `insecure` nicht leichtfertig ein.* Wenn Sie das Passwort von `root` vergessen, wird es schwierig, in den Single-User-Modus zu gelangen, wenn Sie den FreeBSD-Boot-Prozess nicht genau verstehen.

4.2.6. Den Videomodus der Konsole anpassen

Der Standard-Videomodus der FreeBSD-Konsole kann auf jeden Modus eingestellt werden, der von Ihrer Grafikkarte und Ihrem Monitor unterstützt wird (beispielsweise 1024x768 oder 1280x1024). Wollen Sie eine andere Einstellung verwenden, müssen Sie Ihren Kernel neu kompilieren, nachdem Sie die zwei folgenden Zeilen in Ihre Kernelkonfigurationsdatei aufgenommen haben:

```
OPTIONS VESA
options SC_PIXEL_MODE
```

Nachdem Sie den Kernel mit diesen zwei Optionen neu kompiliert haben, bestimmen Sie die möglichen Videomodi mit dem Werkzeug vidcontrol(1). Um beispielsweise einer Liste aller unterstützten Modi zu erhalten, verwenden Sie den folgenden Befehl:

```
# vidcontrol -i mode
```

Als Ergebnis erhalten Sie eine Liste aller Videomodi, die von Ihrer Hardware unterstützt werden. Sie wählen einen neuen Modus aus, indem Sie den entsprechenden Wert (wiederum als Benutzer `root`) an vidcontrol(1) übergeben:

```
# vidcontrol MODE_279
```

Um diese Einstellung dauerhaft zu speichern, müssen Sie die folgende Zeile in die Datei `/etc/rc.conf` aufnehmen:

```
allscreens_flags="MODE_279"
```

4.3. Zugriffsrechte

FreeBSD, das ein direkter Abkömmling von BSD UNIX ist, stützt sich auf mehrere Grundkonzepte von UNIX Systemen. Das erste und ausgeprägteste: FreeBSD ist ein Mehrbenutzer-Betriebssystem. Das System ermöglicht, dass mehrere Benutzer gleichzeitig an völlig verschiedenen und unabhängigen Aufgaben arbeiten können. Es ist verantwortlich für eine gerechte Auf- und Zuteilung von Nachfragen nach Hardware- und Peripheriegeräten, Speicher und CPU-Zeit unter den Benutzern.

Da das System mehrere Benutzer unterstützt, hat alles, was das System verwaltet, einen Satz von Rechten, die bestimmen, wer die jeweilige Ressource lesen, schreiben oder ausführen darf. Diese Zugriffsrechte stehen in drei Achtergruppen, die in drei Teile unterteilt sind: einen für den Besitzer der Datei, einen für die Gruppe, zu der die Datei gehört und einen für alle anderen. Die numerische Darstellung sieht wie folgt aus:

Wert	Zugriffsrechte	Auflistung im Verzeichnis
0	Kein Lesen, Kein Schreiben, Kein Ausführen	`---`
1	Kein Lesen, Kein Schreiben, Ausführen	`--x`
2	Kein Lesen, Schreiben, Kein Ausführen	`-w-`
3	Kein Lesen, Schreiben, Ausführen	`-wx`
4	Lesen, Kein Schreiben, Kein Ausführen	`r--`
5	Lesen, Kein Schreiben, Ausführen	`r-x`
6	Lesen, Schreiben, Kein Ausführen	`rw-`
7	Lesen, Schreiben, Ausführen	`rwx`

Sie können `-l` auf der Kommandozeile von ls(1) angeben, um eine ausführliche Verzeichnisauflistung zu sehen, die in einer Spalte die Zugriffsrechte für den Besitzer, die Gruppe und alle anderen enthält. Die Ausgabe von `ls -l` könnte wie folgt aussehen:

```
% ls -l
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
...
```

Die erste Spalte der Ausgabe enthält die Zugriffsrechte:

```
-rw-r--r--
```

Das erste Zeichen von links ist ein Symbol, welches angibt, ob es sich um eine normale Datei, ein Verzeichnis, ein zeichenorientiertes Gerät, ein Socket oder irgendeine andere Pseudo-Datei handelt. In diesem Beispiel zeigt `-` eine normale Datei an. Die nächsten drei Zeichen, dargestellt als `rw-`, ergeben die Rechte für den Datei-Besitzer. Die drei Zeichen danach `r--` die Rechte der Gruppe, zu der die Datei gehört. Die letzten drei Zeichen, `r--`, geben die Rechte für den Rest der Welt an. Ein Minus bedeutet, dass das Recht nicht gegeben ist. In diesem Fall sind die Zugriffsrechte also: der Eigentümer kann die Datei lesen und schreiben, die Gruppe kann lesen und alle anderen können auch nur lesen. Entsprechend obiger Tabelle wären die Zugriffsrechte für diese Datei `644`, worin jede Ziffer die drei Teile der Zugriffsrechte dieser Datei verkörpert.

Das ist alles schön und gut, aber wie kontrolliert das System die Rechte von Hardware-Geräten? FreeBSD behandelt die meisten Hardware-Geräte als Dateien, welche Programme öffnen, lesen und mit Daten beschreiben können wie alle anderen Dateien auch. Diese Spezial-Dateien sind im Verzeichnis `/dev` gespeichert.

Verzeichnisse werden ebenfalls wie Dateien behandelt. Sie haben Lese-, Schreib- und Ausführ-Rechte. Das Ausführungs-Bit hat eine etwas andere Bedeutung für ein Verzeichnis als für eine Datei. Die Ausführbarkeit eines Verzeichnisses bedeutet, dass in das Verzeichnis zum Beispiel mit `cd` gewechselt werden kann. Das bedeutet auch, dass in dem Verzeichnis auf Dateien, deren Namen bekannt sind, zugegriffen werden kann, vorausgesetzt die Zugriffsrechte der Dateien lassen dies zu.

Das Leserecht auf einem Verzeichnis erlaubt es, sich den Inhalt des Verzeichnisses anzeigen zu lassen. Um eine Datei mit bekanntem Namen in einem Verzeichnis zu löschen, müssen auf dem Verzeichnis Schreib- *und*

Ausführ-Rechte gesetzt sein.

Es gibt noch mehr Rechte, aber die werden vor allem in speziellen Umständen benutzt, wie zum Beispiel bei SetUID-Binaries und Verzeichnissen mit gesetztem Sticky-Bit. Mehr über Zugriffsrechte von Dateien und wie sie gesetzt werden, finden Sie in chmod(1).

4.3.1. Symbolische Zugriffsrechte

Beigesteuert von Tom Rhodes.

Die Zugriffsrechte lassen sich auch über Symbole anstelle von oktalen Werten festlegen. Symbolische Zugriffsrechte werden in der Reihenfolge *`Wer`*, *`Aktion`* und *`Berechtigung`* angegeben. Die folgenden Symbole stehen zur Auswahl:

Option	Symbol	Bedeutung
`Wer`	u	Benutzer (*user*)
`Wer`	g	Gruppe (*group*)
`Wer`	o	Andere (*other*)
`Wer`	a	Alle
`Aktion`	+	Berechtigungen hinzufügen
`Aktion`	-	Berechtigungen entziehen
`Aktion`	=	Berechtigungen explizit setzen
`Berechtigung`	r	lesen (*read*)
`Berechtigung`	w	schreiben (*write*)
`Berechtigung`	x	ausführen (*execute*)
`Berechtigung`	t	Sticky-Bit
`Berechtigung`	s	Set-UID oder Set-GID

Symbolische Zugriffsrechte werden wie die numerischen mit dem Kommando chmod(1) vergeben. Wenn Sie beispielsweise allen anderen Benutzern den Zugriff auf die Datei *`FILE`* verbieten wollen, benutzen Sie den nachstehenden Befehl:

```
% chmod go= FILE
```

Wenn Sie mehr als eine Änderung der Rechte einer Datei vornehmen wollen, können Sie eine durch Kommata getrennte Liste der Rechte angeben. Das folgende Beispiel entzieht der Gruppe und der Welt (den anderen) die Schreibberechtigung auf die Datei *`FILE`* und fügt dann für alle Ausführungsrechte hinzu:

```
% chmod go-w,a+x FILE
```

4.3.2. FreeBSD Datei-Flags

Beigetragen von Tom Rhodes.

Zusätzlich zu den vorhin diskutierten Zugriffsrechten unterstützt FreeBSD auch die sogenannten “Datei-Flags”. Diese erhöhen die Sicherheit Ihres Systems, indem sie eine verbesserte Kontrolle von Dateien erlauben. Verzeichnisse werden allerdings nicht unterstützt.

Diese verbesserte Sicherheit führt dazu, dass manche Dateien nicht einmal von `root` gelöscht oder bearbeitet werden können.

Datei-Flags können über chflags(1) gesetzt oder gelöscht werden. Um beispielsweise die Datei `file1` mit dem "unlöschbar"-Flag zu sichern, geben Sie folgenden Befehl ein:

```
# chflags sunlink file1
```

Um dieses Flag wieder zu löschen, geben Sie den Befehl erneut ein. Allerdings setzen Sie ein "no" vor `sunlink`:

```
# chflags nosunlink file1
```

Um die Flags dieser Datei anzuzeigen, verwenden Sie ls(1) zusammen mit der Option `-lo`:

```
# ls -lo file1
```

Dadurch erhalten Sie eine Ausgabe ähnlich der folgenden:

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

Viele Flags können nur von `root` gesetzt oder gelöscht werden. Andere wiederum können auch vom Eigentümer der Datei gesetzt werden. Weitere Informationen zu Datei-Flags finden sich in den Manualpages chflags(1) und chflags(2).

4.3.3. Die Berechtigungen setuid, setgid, und sticky

Beigetragen von Tom Rhodes.

Anders als die Berechtigungen, die bereits angesprochen wurden, existieren drei weitere Einstellungen, über die alle Administratoren Bescheid wissen sollten. Dies sind die Berechtigungen `setuid`, `setgid` und `sticky`.

Diese Einstellungen sind wichtig für manche UNIX-Operationen, da sie Funktionalitäten zur Verfügung stellen, die normalerweise nicht an gewöhnliche Anwender vergeben wird. Um diese zu verstehen, muss der Unterschied zwischen der realen und der effektiven Benutzer-ID erwähnt werden.

Die reale Benutzer-ID ist die UID, welche den Prozess besitzt oder gestartet hat. Die effektive UID ist diejenige, als die der Prozess läuft. Beispielsweise wird passwd(1) mit der realen ID des Benutzers ausgeführt, der sein Passwort ändert. Um jedoch die Passwortdatenbank zu bearbeiten, wird es effektiv als `root`-Benutzer ausgeführt. Das ermöglicht es normalen Benutzern, ihr Passwort zu ändern, ohne einen `Permission Denied`-Fehler angezeigt zu bekommen.

> **Anmerkung:** Die `nosuid` mount(8)-Option wird dafür sorgen, dass diese Anwendungen stillschweigend scheitern. Genauer gesagt, sie werden nicht ausgeführt und der Anwender wird darüber auch nicht informiert. Auf diese Option kann man sich nicht vollständig verlassen, da ein `nosuid`-Wrapper in der Lage wäre, dies zu umgehen, wie in der mount(8) Manualpage zu lesen ist.

Die setuid-Berechtigung kann durch das Voranstellen bei einer Berechtigungsgruppe mit der Nummer Vier (4) gesetzt werden, wie im folgenden Beispiel gezeigt wird:

```
# chmod 4755 suidexample.sh
```

Die Berechtigungen auf der *suidexample.sh*-Datei sollten jetzt wie folgt aussehen:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidexample.sh
```

In dem Beispiel sollte auffallen, dass ein `s` jetzt Teil der Berechtigungen des Dateibesitzers geworden ist, welches das Ausführen-Bit ersetzt. Dies ermöglicht es Werkzeugen mit erhöhten Berechtigungen zu laufen, wie z.B. `passwd`.

Um dies in Echtzeit zu beobachten, öffnen Sie zwei Terminals. Starten Sie auf einem den `passwd`-Prozess als normaler Benutzer. Während es auf die Passworteingabe wartet, überprüfen Sie die Prozesstabelle und sehen Sie sich die Informationen des `passwd`-Kommandos an.

Im Terminal A:

```
Changing local password for trhodes
Old Password:
```

Im Terminal B:

```
# ps aux | grep passwd

trhodes  5232  0.0  0.2  3420  1608   0  R+    2:10AM   0:00.00 grep passwd
        root     5211  0.0  0.2  3620  1724   2  I+    2:09AM   0:00.01 passwd
```

Wie oben erwähnt, wird `passwd` von einem normalen Benutzer ausgeführt, benutzt aber die effektive UID von `root`.

Die `setgid`-Berechtigung führt die gleiche Aktion wie die `setuid`-Berechtigung durch, allerdings verändert sie die Gruppenberechtigungen. Wenn eine Anwendung oder ein Werkzeug mit dieser Berechtigung ausgeführt wird, erhält es die Berechtigungen basierend auf der Gruppe, welche die Datei besitzt und nicht die des Benutzers, der den Prozess gestartet hat.

Um die `setgid`-Berechtigung auf einer Datei zu setzen, geben Sie dem `chmod`-Befehl eine führende Zwei (2) mit, wie im folgenden gezeigt:

```
# chmod 2755 sgidexample.sh
```

Die neue Einstellung kann wie zuvor betrachtet werden. Beachten Sie, dass das `s` sich jetzt in dem Feld befindet, das für die Berechtigungen der Gruppe bestimmt ist:

```
-rwxr-sr-x   1 trhodes  trhodes    44 Aug 31 01:49 sgidexample.sh
```

> **Anmerkung:** Obwohl es sich bei dem in diesen Beispielen gezeigten Shellskript um eine ausführbare Datei handelt, wird es nicht mit einer anderen EUID oder effektiven Benutzer-ID ausgeführt. Das ist so, weil Shellskripte keinen Zugriff auf setuid(2)-Systemaufrufe erhalten.

Diese beiden ersten angesprochenen Spezialberechtigungen (die `setuid` und `setgid` Berechtigungs-Bits) können die Systemsicherheit verringern, da sie erhöhte Rechte ermöglichen. Es gibt noch ein drittes Berechtigungs-Bit, das die Sicherheit eines Systems erhöhen kann: das `sticky bit`.

Das `sticky bit` erlaubt, wenn es auf ein Verzeichnis angewendet wird, nur dem Besitzer der Datei diese Dateien auch zu löschen. Dieses Recht ist nützlich, um die Löschung von Dateien in öffentlichen Verzeichnissen durch Benutzer, denen diese Dateien nicht gehören, zu verhindern, wie z.B. in `/tmp`. Um diese Berechtigung anzuwenden, stellen Sie der Berechtigung eine Eins (1) voran, beispielsweise so:

```
# chmod 1777 /tmp
```

Den Effekt können Sie sich ansehen, indem Sie das Kommando `ls` ausführen:

```
# ls -al / | grep tmp

drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

Das `sticky bit` kann anhand des `t` ganz am Ende der Berechtigungen abgelesen werden.

4.4. Verzeichnis-Strukturen

Die FreeBSD-Verzeichnishierarchie ist die Grundlage, um ein umfassendes Verständnis des Systems zu erlangen. Das wichtigste Konzept, das Sie verstehen sollten, ist das Root-Verzeichnis "/". Dieses Verzeichnis ist das erste, das während des Bootens eingehangen wird. Es enthält das notwendige Basissystem, um das System in den Mehrbenutzerbetrieb zu bringen. Das Root-Verzeichnis enthält auch die Mountpunkte für Dateisysteme, die beim Wechsel in den Multiuser-Modus eingehängt werden.

Ein Mountpunkt ist ein Verzeichnis, in das zusätzliche Dateisysteme (in der Regel unterhalb des Wurzelverzeichnisses) eingehängt werden können. Dieser Vorgang wird in Abschnitt 4.5 ausführlich beschrieben. Standard-Mountpunkte sind `/usr`, `/var`, `/tmp`, `/mnt` sowie `/cdrom`. Auf diese Verzeichnisse verweisen üblicherweise Einträge in der Datei `/etc/fstab`. `/etc/fstab` ist eine Tabelle mit verschiedenen Dateisystemen und Mountpunkten als Referenz des Systems. Die meisten der Dateisysteme in `/etc/fstab` werden beim Booten automatisch durch das Skript rc(8) gemountet, wenn die zugehörigen Einträge nicht mit der Option `noauto` versehen sind. Weitere Informationen zu diesem Thema finden Sie im Abschnitt 4.6.1.

Eine vollständige Beschreibung der Dateisystem-Hierarchie finden Sie in hier(7). Als Beispiel sei eine kurze Übersicht über die am häufigsten verwendeten Verzeichnisse gegeben:

Verzeichnis	Beschreibung
`/`	Wurzelverzeichnis des Dateisystems.
`/bin/`	Grundlegende Werkzeuge für den Single-User-Modus sowie den Mehrbenutzerbetrieb.
`/boot/`	Programme und Konfigurationsdateien, die während des Bootens benutzt werden.
`/boot/defaults/`	Vorgaben für die Boot-Konfiguration, siehe loader.conf(5).
`/dev/`	Gerätedateien, siehe intro(4).
`/etc/`	Konfigurationsdateien und Skripten des Systems.
`/etc/defaults/`	Vorgaben für die System Konfigurationsdateien, siehe rc(8).
`/etc/mail/`	Konfigurationsdateien von MTAs wie sendmail(8).
`/etc/namedb/`	Konfigurationsdateien von `named`, siehe named(8).
`/etc/periodic/`	Täglich, wöchentlich oder monatlich ablaufende Skripte, die von cron(8) gestartet werden. Siehe periodic(8).
`/etc/ppp/`	Konfigurationsdateien von `ppp`, siehe ppp(8).

Verzeichnis	Beschreibung
/mnt/	Ein leeres Verzeichnis, das von Systemadministratoren häufig als temporärer Mountpunkt genutzt wird.
/proc/	Prozess Dateisystem, siehe procfs(5) und mount_procfs(8).
/rescue/	Statisch gelinkte Programme zur Wiederherstellung des Systems, lesen Sie dazu auch rescue(8).
/root/	Home Verzeichnis von root.
/sbin/	Systemprogramme und administrative Werkzeuge, die grundlegend für den Single-User-Modus und den Mehrbenutzerbetrieb sind.
/tmp/	Temporäre Dateien, die für gewöhnlich bei einem Neustart des Systems verloren gehen. Häufig wird ein speicherbasiertes Dateisystem unter /tmp eingehängt. Dieser Vorgang kann automatisiert werden, wenn Sie die tmpmfs-bezogenen Variablen von rc.conf(5) verwenden. Alternativ können Sie auch einen entsprechenden Eintrag in /etc/fstab aufnehmen. Weitere Informationen finden Sie in mdmfs(8).
/usr/	Der Großteil der Benutzerprogramme und Anwendungen.
/usr/bin/	Gebräuchliche Werkzeuge, Programmierhilfen und Anwendungen.
/usr/include/	Standard C include-Dateien.
/usr/lib/	Bibliotheken.
/usr/libdata/	Daten verschiedener Werkzeuge.
/usr/libexec/	System-Dämonen und System-Werkzeuge, die von anderen Programmen ausgeführt werden.
/usr/local/	Lokale Programme, Bibliotheken usw. Die Ports-Sammlung benutzt dieses Verzeichnis als Zielverzeichnis für zu installierende Anwendungen. Innerhalb von /usr/local sollte das von hier(7) beschriebene Layout für /usr benutzt werden. Das man Verzeichnis wird direkt unter /usr/local anstelle unter /usr/local/share angelegt. Die Dokumentation der Ports findet sich in share/doc/*port*.
/usr/obj/	Von der Architektur abhängiger Verzeichnisbaum, der durch das Bauen von /usr/src entsteht.
/usr/ports/	Die FreeBSD-Ports-Sammlung (optional).
/usr/sbin/	System-Dämonen und System-Werkzeuge, die von Benutzern ausgeführt werden.
/usr/share/	Von der Architektur unabhängige Dateien.
/usr/src/	Quelldateien von BSD und/oder lokalen Ergänzungen.
/usr/X11R6/	Optionale X11R6-Programme und Bibliotheken.

Verzeichnis	Beschreibung
`/var/`	Wird für mehrere Zwecke genutzt und enthält Logdateien, temporäre Daten und Spooldateien. Manchmal wird ein speicherbasiertes Dateisystem unter `/var` eingehängt. Dieser Vorgang kann automatisiert werden, wenn Sie die varmfs-bezogenen Variablen von rc.conf(5) verwenden. Alternativ können Sie auch einen entsprechenden Eintrag in `/etc/fstab` aufnehmen. Weitere Informationen finden Sie in mdmfs(8).
`/var/log/`	Verschiedene Logdateien des Systems.
`/var/mail/`	Postfächer der Benutzer.
`/var/spool/`	Verschiedene Spool-Verzeichnisse der Drucker- und Mailsysteme.
`/var/tmp/`	Temporäre Dateien. Dateien in diesem Verzeichnis bleiben in der Regel auch bei einem Neustart des Systems erhalten, es sei denn, bei `/var` handelt es sich um ein speicherbasiertes Dateisystem.
`/var/yp/`	NIS maps.

4.5. Festplatten, Slices und Partitionen

FreeBSD identifiziert Dateien anhand eines Dateinamens. In Dateinamen wird zwischen Groß- und Kleinschreibung unterschieden: `readme.txt` und `README.TXT` bezeichnen daher zwei verschiedene Dateien. FreeBSD benutzt keine Dateiendungen wie `.txt`, um den Typ der Datei (ein Programm, ein Dokument oder andere Daten) zu bestimmen.

Dateien werden in Verzeichnissen gespeichert. In einem Verzeichnis können sich keine oder hunderte Dateien befinden. Ein Verzeichnis kann auch andere Verzeichnisse enthalten und so eine Hierarchie von Verzeichnissen aufbauen, die Ihnen die Ablage von Daten erleichtert.

In Dateinamen werden Verzeichnisse durch einen Schrägstrich (`/`, *Slash*) getrennt. Wenn das Verzeichnis `foo` ein Verzeichnis `bar` enthält, in dem sich die Datei `readme.txt` befindet, lautet der vollständige Name der Datei (oder der *Pfad* zur Datei) `foo/bar/readme.txt`.

Verzeichnisse und Dateien werden in einem Dateisystem gespeichert. Jedes Dateisystem besitzt ein *Wurzelverzeichnis* (*Root-Directory*), das weitere Verzeichnisse enthalten kann.

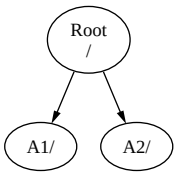
Dieses Konzept kennen Sie vielleicht von anderen Betriebssystemen, aber es gibt einige Unterschiede: In MS-DOS werden Datei- und Verzeichnisnamen mit dem Zeichen `\` getrennt, Mac OS® benutzt dazu das Zeichen `:`.

FreeBSD kennt keine Laufwerksbuchstaben und in Pfaden werden keine Bezeichnungen für Laufwerke benutzt. Die Pfadangabe `c:/foo/bar/readme.txt` gibt es in FreeBSD nicht.

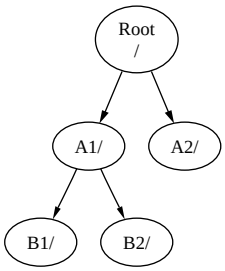
Stattdessen wird ein Dateisystem als Wurzeldateisystem (*root file system*) ausgewählt. Das Wurzelverzeichnis dieses Dateisystems wird `/` genannt. Jedes andere Dateisystem wird unter dem Wurzeldateisystem *eingehangen* (*mount*). Daher scheint jedes Verzeichnis, unabhängig von der Anzahl der Platten, auf derselben Platte zu liegen.

Betrachten wir drei Dateisysteme `A`, `B` und `C`. Jedes Dateisystem besitzt ein eigenes Wurzelverzeichnis, das zwei andere Verzeichnisse enthält: `A1`, `A2`, `B1`, `B2`, `C1` und `C2`.

Das Wurzeldateisystem soll `A` sein. Das Kommando `ls` zeigt darin die beiden Verzeichnisse `A1` und `A2` an. Der Verzeichnisbaum sieht wie folgt aus:

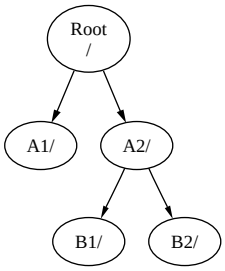


Ein Dateisystem wird in einem Verzeichnis eines anderen Dateisystems eingehangen. Wir hängen nun das Dateisystem `B` in das Verzeichnis `A1` ein. Das Wurzelverzeichnis von `B` ersetzt nun das Verzeichnis `A1` und die Verzeichnisse des Dateisystems `B` werden sichtbar:



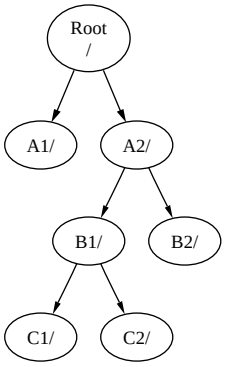
Jede Datei in den Verzeichnissen `B1` oder `B2` kann über den Pfad `/A1/B1` oder `/A1/B2` erreicht werden. Dateien aus dem Verzeichnis `/A1` sind jetzt verborgen. Wenn das Dateisystem `B` wieder *abgehangen* wird (*umount*), erscheinen die verborgenen Dateien wieder.

Wenn das Dateisystem `B` unter dem Verzeichnis `A2` eingehangen würde, sähe der Verzeichnisbaum so aus:

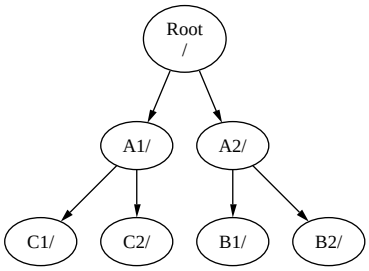


Die Dateien des Dateisystems `B` wären unter den Pfaden `/A2/B1` und `/A2/B2` erreichbar.

Dateisysteme können übereinander eingehangen werden. Der folgende Baum entsteht, wenn im letzten Beispiel das Dateisystem `C` in das Verzeichnis `B1` des Dateisystems `B` eingehangen wird:



`C` könnte auch im Verzeichnis `A1` eingehangen werden:



Der MS-DOS-Befehl `join` kann Ähnliches bewirken.

Normalerweise müssen Sie sich nicht mit Dateisystemen beschäftigen. Während der Installation werden die Dateisysteme und die Stellen, in der sie eingehangen werden, festgelegt. Dateisysteme müssen Sie erst wieder anlegen, wenn Sie eine neue Platte hinzufügen.

Sie können sogar mit nur einem großen Dateisystem auskommen. Dies hat mehrere Nachteile und einen Vorteil.

Vorteile mehrerer Dateisysteme

- Die Dateisysteme können mit unterschiedlichen Optionen (*mount options*) eingehangen werden. Bei sorgfältiger Planung können Sie beispielsweise das Wurzeldateisystem nur lesbar einhängen. Damit schützen Sie sich vor dem unabsichtlichen Löschen oder Editieren kritischer Dateien. Von Benutzern beschreibbare Dateisysteme wie `/home` können Sie mit der Option *nosuid* einhängen, wenn sie von anderen Dateisystemen getrennt sind. Die *SUID*- und *GUID*-Bits verlieren auf solchen Dateisystemen ihre Wirkung und die Sicherheit des Systems kann dadurch erhöht werden.
- Die Lage von Dateien im Dateisystem wird, abhängig vom Gebrauch des Dateisystems, automatisch von FreeBSD optimiert. Ein Dateisystem mit vielen kleinen Dateien, die häufig geschrieben werden, wird anders behandelt als ein Dateisystem mit wenigen großen Dateien. Mit nur einem Dateisystem ist diese Optimierung unmöglich.
- In der Regel übersteht ein FreeBSD-Dateisystem auch einen Stromausfall. Allerdings kann ein Stromausfall zu einem kritischen Zeitpunkt das Dateisystem beschädigen. Wenn die Daten über mehrere Dateisysteme verteilt sind, lässt sich das System mit hoher Wahrscheinlichkeit noch starten. Dies erleichtert das Zurückspielen von Datensicherungen.

Vorteil eines einzelnen Dateisystems

- Die Größe von Dateisystemen liegt fest. Es kann passieren, dass Sie eine Partition vergrößern müssen. Dies ist nicht leicht: Sie müssen die Daten sichern, das Dateisystem vergrößert anlegen und die gesicherten Daten zurückspielen.

  Wichtig: FreeBSD kennt den Befehl growfs(8), mit dem man Dateisysteme im laufenden Betrieb vergrößern kann.

Dateisysteme befinden sich in Partitionen (damit sind nicht die normalen MS-DOS-Partitionen gemeint). Jede Partition wird mit einem Buchstaben von a bis h bezeichnet und kann nur ein Dateisystem enthalten. Dateisysteme können daher über ihren Mount-Point, den Punkt an dem sie eingehangen sind, oder den Buchstaben der Partition, in der sie liegen, identifiziert werden.

FreeBSD benutzt einen Teil der Platte für den *Swap-Bereich*, der dem Rechner *virtuellen Speicher* zur Verfügung stellt. Dadurch kann der Rechner Anwendungen mehr Speicher zur Verfügung stellen als tatsächlich eingebaut ist. Wenn der Speicher knapp wird, kann FreeBSD nicht benutzte Daten in den Swap-Bereich auslagern. Die ausgelagerten Daten können später wieder in den Speicher geholt werden (dafür werden dann andere Daten ausgelagert).

Für einige Partitionen gelten besondere Konventionen:

Partition	Konvention
a	Enthält normalerweise das Wurzeldateisystem
b	Enthält normalerweise den Swap-Bereich
c	Ist normalerweise genauso groß wie die Slice in der die Partition liegt. Werkzeuge, die auf der kompletten Slice arbeiten, wie ein Bad-Block-Scanner, können so die c-Partition benutzen. Für gewöhnlich legen Sie in dieser Partition kein Dateisystem an.
d	Früher hatte die d-Partition eine besondere Bedeutung. Heute ist dies nicht mehr der Fall und die Partition d kann wie jede andere Partition auch verwendet werden.

Jede Partition, die ein Dateisystem enthält, wird in einer *Slice* angelegt. Slice ist der Begriff, den FreeBSD für MS-DOS-Partitionen verwendet. Slices werden von eins bis vier durchnummeriert.

Die Slice-Nummern werden mit vorgestelltem s hinter den Gerätenamen gestellt: “da0*s1*” ist die erste Slice auf dem ersten SCSI-Laufwerk. Auf einer Festplatte gibt es höchstens vier Slices. In einer Slice des passenden Typs kann es weitere logische Slices geben. Diese erweiterten Slices werden ab fünf durchnummeriert: “ad0s*5*” ist die erste erweiterte Slice auf einer IDE-Platte. Diese Geräte werden von Dateisystemen benutzt, die sich in einer kompletten Slice befinden müssen.

Slices, “dangerously dedicated”-Festplatten und andere Platten enthalten Partitionen, die mit Buchstaben von a bis h bezeichnet werden. Der Buchstabe wird an den Gerätenamen gehangen: “da0*a*” ist die a-Partition des ersten da-Laufwerks. Dieses Laufwerk ist “dangerously dedicated”. “ad1s3*e*” ist die fünfte Partition in der dritten Slice der zweiten IDE-Platte.

Schließlich wird noch jede Festplatte des Systems eindeutig bezeichnet. Der Name einer Festplatte beginnt mit einem Code, der den Typ der Platte bezeichnet. Es folgt eine Nummer, die angibt, um welche Festplatte es sich handelt. Anders als bei Slices werden Festplatten von Null beginnend durchnummeriert. Gängige Festplatten-Namen sind in Tabelle 4-1 zusammengestellt.

Wenn Sie eine Partition angeben, erwartet FreeBSD, dass Sie auch die Slice und die Platte angeben, in denen sich die Partition befindet. Wenn Sie eine Slice angeben, müssen Sie auch die Platte der Slice angeben. Setzen Sie den Namen aus dem Plattennamen gefolgt von einem `s`, der Slice-Nummer und dem Buchstaben der Partition zusammen. Einige Beispiele finden Sie in Beispiel 4-1.

Der Aufbau einer Festplatte wird in Beispiel 4-2 dargestellt.

Um FreeBSD zu installieren, müssen Sie zuerst Slices auf den Festplatten anlegen. Innerhalb der Slices, die Sie für FreeBSD verwenden wollen, müssen Sie dann Partitionen anlegen. In den Partitionen wiederum werden die Dateisysteme (oder der Auslagerungsbereich) angelegt. Für Dateisysteme müssen Sie schließlich noch festlegen, wo diese eingehangen werden (Mount-Point).

Tabelle 4-1. Laufwerk-Codes

Code	Bedeutung
`ad`	ATAPI (IDE) Festplatte
`da`	SCSI-Festplatte
`acd`	ATAPI (IDE) CD-ROM
`cd`	SCSI-CD-ROM
`fd`	Disketten-Laufwerk

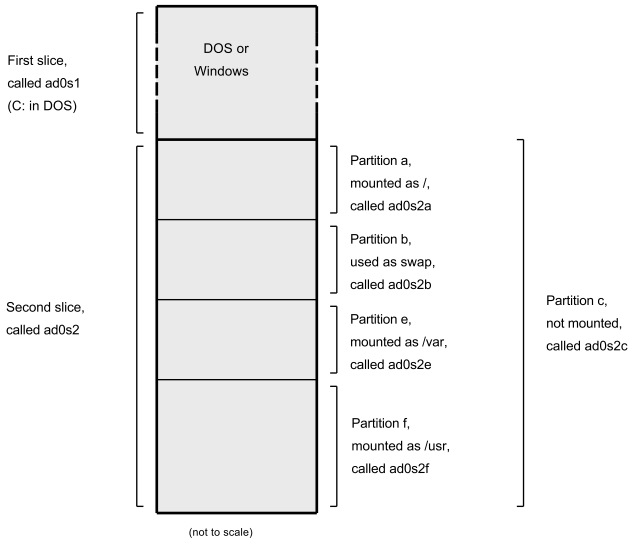
Beispiel 4-1. Namen von Platten, Slices und Partitionen

Name	Bedeutung
`ad0s1a`	Die erste Partition (`a`) in der ersten Slice (`s1`) der ersten IDE-Festplatte (`ad0`).
`da1s2e`	Die fünfte Partition (`e`) der zweiten Slice (`s2`) auf der zweiten SCSI-Festplatte (`da1`).

Beispiel 4-2. Aufteilung einer Festplatte

Das folgende Diagramm zeigt die Sicht von FreeBSD auf die erste IDE-Festplatte eines Rechners. Die Platte soll 4 GB groß sein und zwei Slices (MS-DOS-Partitionen) mit je 2 GB besitzen. Die erste Slice enthält ein MS-DOS-Laufwerk (`C:`), die zweite Slice wird von FreeBSD benutzt. Im Beispiel verwendet die FreeBSD-Installationen drei Datenpartitionen und einen Auslagerungsbereich.

Jede der drei Partitionen enthält ein Dateisystem. Das Wurzeldateisystem ist die `a`-Partition. In der `e`-Partition befindet sich der `/var`-Verzeichnisbaum und in der `f`-Partition befindet sich der Verzeichnisbaum unterhalb von `/usr`.



4.6. Anhängen und Abhängen von Dateisystemen

Ein Dateisystem wird am besten als ein Baum mit der Wurzel `/` veranschaulicht. `/dev`, `/usr`, und die anderen Verzeichnisse im Rootverzeichnis sind Zweige, die wiederum eigene Zweige wie `/usr/local` haben können.

Es gibt verschiedene Gründe, bestimmte dieser Verzeichnisse auf eigenen Dateisystemen anzulegen. `/var` enthält `log/`, `spool/` sowie verschiedene andere temporäre Dateien und kann sich daher schnell füllen. Es empfiehlt sich, `/var` von `/` zu trennen, da es schlecht ist, wenn das Root-Dateisystem voll läuft.

Ein weiterer Grund bestimmte Verzeichnisbäume auf andere Dateisysteme zu legen, ist gegeben, wenn sich die Verzeichnisbäume auf gesonderten physikalischen oder virtuellen Platten, wie Network File System oder CD-ROM-Laufwerken, befinden.

4.6.1. Die `fstab` Datei

Während des Boot-Prozesses werden in `/etc/fstab` aufgeführte Verzeichnisse, sofern sie nicht mit der Option `noauto` versehen sind, automatisch angehangen.

Die Zeilen in `/etc/fstab` haben das folgende Format:

```
device  /mount-point    fstype  options dumpfreq        passno
```

`device`

Ein existierender Gerätename wie in Abschnitt 19.2 beschrieben.

`mount-point`

Ein existierendes Verzeichnis, an das das Dateisystem angehangen wird.

`fstype`

Der Typ des Dateisystems, der an mount(8) weitergegeben wird. FreeBSDs Standarddateisystem ist `ufs`.

`options`

Entweder `rw` für beschreibbare Dateisysteme oder `ro` für schreibgeschützte Dateisysteme, gefolgt von weiteren benötigten Optionen. Eine häufig verwendete Option ist `noauto` für Dateisysteme, die während der normalen Bootsequenz nicht angehangen werden sollen. Weitere Optionen finden sich in mount(8).

`dumpfreq`

Gibt die Anzahl der Tage an, nachdem das Dateisystem gesichert werden soll. Fehlt der Wert, wird `0` angenommen.

`passno`

Bestimmt die Reihenfolge, in der die Dateisysteme überprüft werden sollen. Für Dateisysteme, die übersprungen werden sollen, ist `passno` auf null zu setzen. Für das Root-Dateisystem, das vor allen anderen überprüft werden muss, sollte der Wert von `passno` eins betragen. Allen anderen Dateisystemen sollten Werte größer eins zugewiesen werden. Wenn mehrere Dateisysteme den gleichen Wert besitzen, wird fsck(8) versuchen, diese parallel zu überprüfen.

4.6.2. Das `mount` Kommando

mount(8) hängt schließlich Dateisysteme an.

In der grundlegenden Form wird es wie folgt benutzt:

```
# mount device mountpoint
```

Viele Optionen werden in mount(8) beschrieben, die am häufigsten verwendeten sind:

Optionen von `mount`

`-a`

Hängt alle Dateisysteme aus `/etc/fstab` an. Davon ausgenommen sind Dateisysteme, die mit “noauto” markiert sind, die mit der Option `-t` ausgeschlossen wurden und Dateisysteme, die schon angehangen sind.

`-d`

Führt alles bis auf den `mount`-Systemaufruf aus. Nützlich ist diese Option in Verbindung mit `-v`. Damit wird angezeigt, was mount(8) tatsächlich versuchen würde, um das Dateisystem anzuhängen.

`-f`

Erzwingt das Anhängen eines unsauberen Dateisystems oder erzwingt die Rücknahme des Schreibzugriffs, wenn der Status des Dateisystems von beschreibbar auf schreibgeschützt geändert wird.

`-r`

Hängt das Dateisystem schreibgeschützt ein. Das kann auch durch Angabe von `ro` als Argument der Option `-o` erreicht werden.

`-t` *`fstype`*

Hängt das Dateisystem mit dem angegebenen Typ an, oder hängt nur Dateisysteme mit dem angegebenen Typ an, wenn auch `-a` angegeben wurde.

Die Voreinstellung für den Typ des Dateisystems ist “ufs”.

`-u`

Aktualisiert die Mountoptionen des Dateisystems.

`-v`

Geschwätzig sein.

`-w`

Hängt das Dateisystem beschreibbar an.

`-o` erwartet eine durch Kommata separierte Liste von Optionen, unter anderem die folgenden:

noexec

Verbietet das Ausführen von binären Dateien auf dem Dateisystem. Dies ist eine nützliche Sicherheitsfunktion.

nosuid

SetUID und SetGID Bits werden auf dem Dateisystem nicht beachtet. Dies ist eine nützliche Sicherheitsfunktion.

4.6.3. Das `umount` Kommando

umount(8) akzeptiert als Parameter entweder einen Mountpoint, einen Gerätenamen, oder die Optionen `-a` oder `-A`.

Jede Form akzeptiert `-f`, um das Abhängen zu erzwingen, und `-v`, um etwas geschwätziger zu sein. Seien Sie bitte vorsichtig mit `-f`: Ihr Computer kann abstürzen oder es können Daten auf dem Dateisystem beschädigt werden, wenn Sie das Abhängen erzwingen.

`-a` und `-A` werden benutzt um alle Dateisysteme, deren Typ durch `-t` modifiziert werden kann, abzuhängen. `-A` hängt das Rootdateisystem nicht ab.

4.7. Prozesse

Da FreeBSD ein Multitasking-Betriebssystem ist, sieht es so aus, als ob mehrere Prozesse zur gleichen Zeit laufen. Jedes Programm, das zu irgendeiner Zeit läuft, wird *Prozess* genannt. Jedes Kommando startet mindestens einen Prozess. Einige Systemprozesse laufen ständig und stellen die Funktion des Systems sicher.

Jeder Prozess wird durch eine eindeutige Nummer identifiziert, die *Prozess-ID* oder *PID* genannt wird. Prozesse haben ebenso wie Dateien einen Besitzer und eine Gruppe, die festlegen, welche Dateien und Geräte der Prozess benutzen kann. Dabei finden die vorher beschriebenen Zugriffsrechte Anwendung. Die meisten Prozesse haben auch einen Elternprozess, der sie gestartet hat. Wenn Sie in der Shell Kommandos eingeben, dann ist die Shell ein Prozess und jedes Kommando, das Sie starten, ist auch ein Prozess. Jeder Prozess, den Sie auf diese Weise starten, besitzt den

Shell-Prozess als Elternprozess. Die Ausnahme hiervon ist ein spezieller Prozess, der init(8) heißt. `init` ist immer der erste Prozess und hat somit die PID 1. `init` wird vom Kernel beim Booten von FreeBSD gestartet.

Die Kommandos ps(1) und top(1) sind besonders nützlich, um sich die Prozesse auf einem System anzusehen. `ps` zeigt eine statische Liste der laufenden Prozesse und kann deren PID, Speicherverbrauch und die Kommandozeile, mit der sie gestartet wurden und vieles mehr anzeigen. `top` zeigt alle laufenden Prozesse an und aktualisiert die Anzeige, so dass Sie Ihrem Computer bei der Arbeit zuschauen können.

Normal zeigt Ihnen `ps` nur die laufenden Prozesse, die Ihnen gehören. Zum Beispiel:

```
% ps
  PID  TT  STAT      TIME COMMAND
  298  p0  Ss     0:01.10 tcsh
 7078  p0  S      2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393  p0  I      0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630  p0  S      2:50.89 /usr/local/lib/netscape-linux/navigator-linux-4.77.bi
48730  p0  IW     0:00.00 (dns helper) (navigator-linux-)
72210  p0  R+     0:00.00 ps
  390  p1  Is     0:01.14 tcsh
 7059  p2  Is+    1:36.18 /usr/local/bin/mutt -y
 6688  p3  IWs    0:00.00 tcsh
10735  p4  IWs    0:00.00 tcsh
20256  p5  IWs    0:00.00 tcsh
  262  v0  IWs    0:00.00 -tcsh (tcsh)
  270  v0  IW+    0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280  v0  IW+    0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284  v0  IW     0:00.00 /bin/sh /home/nik/.xinitrc
  285  v0  S      0:38.45 /usr/X11R6/bin/sawfish
```

Wie Sie sehen, gibt ps(1) mehrere Spalten aus. In der `PID` Spalte findet sich die vorher besprochene Prozess-ID. PIDs werden von 1 beginnend bis 99999 zugewiesen und fangen wieder von vorne an, wenn die Grenze überschritten wird. Ist eine PID bereits vergeben, wird diese allerdings nicht erneut vergeben. Die Spalte `TT` zeigt den Terminal, auf dem das Programm läuft. `STAT` zeigt den Status des Programms an und kann für die Zwecke dieser Diskussion ebenso wie `TT` ignoriert werden. `TIME` gibt die Zeit an, die das Programm auf der CPU gelaufen ist – dies ist nicht unbedingt die Zeit, die seit dem Start des Programms vergangen ist, da die meisten Programme hauptsächlich auf bestimmte Dinge warten, bevor sie wirklich CPU-Zeit verbrauchen. Unter der Spalte `COMMAND` finden Sie schließlich die Kommandozeile, mit der das Programm gestartet wurde.

ps(1) besitzt viele Optionen, um die angezeigten Informationen zu beeinflussen. Eine nützliche Kombination ist `auxww`. Mit `a` werden Information über alle laufenden Prozesse und nicht nur Ihrer eigenen angezeigt. Der Name des Besitzers des Prozesses, sowie Informationen über den Speicherverbrauch werden mit `u` angezeigt. `x` zeigt auch Dämonen-Prozesse an, und `ww` veranlasst ps(1) die komplette Kommandozeile für jeden Befehl anzuzeigen, anstatt sie abzuschneiden, wenn sie zu lang für die Bildschirmausgabe wird.

Die Ausgabe von top(1) sieht ähnlich aus:

```
% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33  22:39:10
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt, 79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse
```

```
  PID USERNAME PRI NICE  SIZE    RES STATE    TIME   WCPU    CPU COMMAND
72257 nik       28   0  1960K  1044K RUN      0:00 14.86%  1.42% top
 7078 nik        2   0 15280K 10960K select   2:54  0.88%  0.88% xemacs-21.1.14
  281 nik        2   0 18636K  7112K select   5:36  0.73%  0.73% XF86_SVGA
  296 nik        2   0  3240K  1644K select   0:12  0.05%  0.05% xterm
48630 nik        2   0 29816K  9148K select   3:18  0.00%  0.00% navigator-linu
  175 root       2   0   924K   252K select   1:41  0.00%  0.00% syslogd
 7059 nik        2   0  7260K  4644K poll     1:38  0.00%  0.00% mutt
...
```

Die Ausgabe ist in zwei Abschnitte geteilt. In den ersten fünf Kopfzeilen finden sich die zuletzt zugeteilte PID, die Systemauslastung (engl. *load average*), die Systemlaufzeit (die Zeit seit dem letzten Reboot) und die momentane Zeit. Die weiteren Zahlen im Kopf beschreiben wie viele Prozesse momentan laufen (im Beispiel 47), wie viel Speicher und Swap verbraucht wurde und wie viel Zeit das System in den verschiedenen CPU-Modi verbringt.

Darunter befinden sich einige Spalten mit ähnlichen Informationen wie in der Ausgabe von ps(1). Wie im vorigen Beispiel können Sie die PID, den Besitzer, die verbrauchte CPU-Zeit und das Kommando erkennen. top(1) zeigt auch den Speicherverbrauch des Prozesses an, der in zwei Spalten aufgeteilt ist. Die erste Spalte gibt den gesamten Speicherverbrauch des Prozesses an, in der zweiten Spalte wird der aktuelle Verbrauch angegeben. **Netscape®** hat im gezeigten Beispiel insgesamt 30 MB Speicher verbraucht. Momentan benutzt es allerdings nur 9 MB.

Die Anzeige wird von top(1) automatisch alle zwei Sekunden aktualisiert. Der Zeitraum kann mit `-s` eingestellt werden.

4.8. Dämonen, Signale und Stoppen von Prozessen

Wenn Sie einen Editor starten, können Sie ihn leicht bedienen und Dateien laden. Sie können das, weil der Editor dafür Vorsorge getroffen hat und auf einem *Terminal* läuft. Manche Programme erwarten keine Eingaben von einem Benutzer und lösen sich bei erster Gelegenheit von ihrem Terminal. Ein Web-Server zum Beispiel verbringt den ganzen Tag damit, auf Anfragen zu antworten und erwartet keine Eingaben von Ihnen. Programme, die E-Mail von einem Ort zu einem anderen Ort transportieren sind ein weiteres Beispiel für diesen Typ von Anwendungen.

Wir nennen diese Programme *Dämonen*. Dämonen stammen aus der griechischen Mythologie und waren weder gut noch böse. Sie waren kleine dienstbare Geister, die meistens nützliche Sachen für die Menschheit vollbrachten. Ähnlich wie heutzutage Web-Server und Mail-Server nützliche Dienste verrichten. Seit langer Zeit ist daher das BSD Maskottchen dieser fröhlich aussehende Dämon mit Turnschuhen und Dreizack.

Programme, die als Dämon laufen, werden entsprechend einer Konvention mit einem “d” am Ende benannt. **BIND** steht beispielsweise für Berkeley Internet Name Domain, das tatsächlich laufende Programm heißt aber `named`. Der Apache Webserver wird `httpd` genannt, der Druckerspool-Dämon heißt `lpd` usw. Dies ist allerdings eine Konvention und keine unumstößliche Regel: Der Dämon der Anwendung **sendmail** heißt `sendmail` und nicht `maild`, wie Sie vielleicht gedacht hatten.

Manchmal müssen Sie mit einem Dämon kommunizieren. Dazu verwenden Sie *Signale*. Sie können mit einem Dämonen oder jedem anderen laufenden Prozess kommunizieren, indem Sie diesem ein Signal schicken. Sie können verschiedene Signale verschicken – manche haben eine festgelegte Bedeutung, andere werden von der Anwendung interpretiert. Die Dokumentation zur fraglichen Anwendung wird erklären, wie die Anwendung Signale interpretiert. Sie können nur Signale zu Prozessen senden, die Ihnen gehören. Normale Benutzer haben nicht die Berechtigung, Prozessen anderer Benutzer mit kill(1) oder kill(2) Signale zu schicken. Der Benutzer `root` darf jedem Prozess Signale schicken.

In manchen Fällen wird FreeBSD Signale senden. Wenn eine Anwendung schlecht geschrieben ist und auf Speicher zugreift, auf den sie nicht zugreifen soll, so sendet FreeBSD dem Prozess das *Segmentation Violation* Signal (`SIGSEGV`). Wenn eine Anwendung den alarm(3) Systemaufruf benutzt hat, um nach einiger Zeit benachrichtigt zu werden, bekommt sie das Alarm Signal (`SIGALRM`) gesendet.

Zwei Signale können benutzt werden, um Prozesse zu stoppen: `SIGTERM` und `SIGKILL`. Mit `SIGTERM` fordern Sie den Prozess höflich zum Beenden auf. Der Prozess kann das Signal abfangen und merken, dass er sich beenden soll. Er hat dann Gelegenheit Logdateien zu schließen und die Aktion, die er vor der Aufforderung sich zu beenden durchführte, abzuschließen. Er kann sogar `SIGTERM` ignorieren, wenn er eine Aktion durchführt, die nicht unterbrochen werden darf.

`SIGKILL` kann von keinem Prozess ignoriert werden. Das Signal lässt sich mit "Mich interessiert nicht, was du gerade machst, hör sofort auf damit!" umschreiben. Wenn Sie einem Prozess `SIGKILL` schicken, dann wird FreeBSD diesen sofort beenden[4].

Andere Signale, die Sie vielleicht verschicken wollen, sind `SIGHUP`, `SIGUSR1` und `SIGUSR2`. Diese Signale sind für allgemeine Zwecke vorgesehen und verschiedene Anwendungen werden unterschiedlich auf diese Signale reagieren.

Nehmen wir an, Sie haben die Konfiguration Ihres Webservers verändert und möchten dies dem Server mitteilen. Sie könnten den Server natürlich stoppen und `httpd` wieder starten. Die Folge wäre eine kurze Zeit, in der der Server nicht erreichbar ist. Die meisten Dämonen lesen Ihre Konfigurationsdatei beim Empfang eines `SIGHUP` neu ein. Da es keinen Standard gibt, der vorschreibt, wie auf diese Signale zu reagieren ist, lesen Sie bitte die Dokumentation zu dem in Frage kommenden Dämon.

Mit kill(1) können Sie, wie unten gezeigt, Signale verschicken.

Verschicken von Signalen

Das folgende Beispiel zeigt, wie Sie inetd(8) ein Signal schicken. Die Konfigurationsdatei von `inetd` ist `/etc/inetd.conf`. Diese Konfigurationsdatei liest `inetd` ein, wenn er ein `SIGHUP` empfängt.

1. Suchen Sie die Prozess-ID des Prozesses, dem Sie ein Signal schicken wollen. Benutzen Sie dazu ps(1) und grep(1). Mit grep(1) können Sie in einer Ausgabe nach einem String suchen. Da inetd(8) unter dem Benutzer `root` läuft und Sie das Kommando als normaler Benutzer absetzen, müssen Sie ps(1) mit `ax` aufrufen:

   ```
   % ps -ax | grep inetd
     198  ??  IWs    0:00.00 inetd -wW
   ```

   Die Prozess-ID von inetd(8) ist 198. In einigen Fällen werden Sie auch das `grep inetd` Kommando in der Ausgabe sehen. Dies hat damit zu tun, wie ps(1) die Liste der laufenden Prozesse untersucht.

2. Senden Sie das Signal mit kill(1). Da inetd(8) unter dem Benutzer `root` läuft, müssen Sie zuerst mit su(1) `root` werden:

   ```
   % su
   Password:
   # /bin/kill -s HUP 198
   ```

   kill(1) wird, wie andere Kommandos von UNIX Systemen auch, keine Ausgabe erzeugen, wenn das Kommando erfolgreich war. Wenn Sie versuchen, einem Prozess, der nicht Ihnen gehört, ein Signal zu senden, dann werden Sie die Meldung `kill: PID: Operation not permitted` sehen. Wenn Sie sich bei der Eingabe der PID vertippen, werden Sie das Signal dem falschen Prozess schicken, was schlecht sein kann. Wenn Sie Glück haben, existiert der Prozess nicht und Sie werden mit der Ausgabe `kill: PID: No such process` belohnt.

> **Warum soll ich `/bin/kill` benutzen?:** Viele Shells stellen `kill` als internes Kommando zur Verfügung, das heißt die Shell sendet das Signal direkt, anstatt `/bin/kill` zu starten. Das kann nützlich sein, aber die unterschiedlichen Shells benutzen eine verschiedene Syntax, um die Namen der Signale anzugeben. Anstatt jede Syntax zu lernen, kann es einfacher sein, `/bin/kill ...` direkt aufzurufen.

Andere Signale senden Sie auf die gleiche Weise, ersetzen Sie nur `TERM` oder `KILL` entsprechend.

> **Wichtig:** Es kann gravierende Auswirkungen haben, wenn Sie zufällig Prozesse beenden. Insbesondere init(8) mit der Prozess-ID ist ein Spezialfall. Mit `/bin/kill -s KILL 1` können Sie Ihr System schnell herunterfahren. Überprüfen Sie die Argumente von kill(1) *immer* zweimal *bevor* Sie **Return** drücken.

4.9. Shells

Von der tagtäglichen Arbeit mit FreeBSD wird eine Menge mit der Kommandozeilen Schnittstelle der Shell erledigt. Die Hauptaufgabe einer Shell besteht darin, Kommandos der Eingabe anzunehmen und diese auszuführen. Viele Shells haben außerdem eingebaute Funktionen, die die tägliche Arbeit erleichtern, beispielsweise eine Dateiverwaltung, die Vervollständigung von Dateinamen (Globbing), einen Kommandozeileneditor, sowie Makros und Umgebungsvariablen. FreeBSD enthält die Shells `sh` (die Bourne Shell) und `tcsh` (die verbesserte C-Shell) im Basissystem. Viele andere Shells, wie `zsh` oder `bash`, befinden sich in der Ports-Sammlung.

Welche Shell soll ich benutzen? Das ist wirklich eine Geschmacksfrage. Sind Sie ein C-Programmierer, finden Sie vielleicht eine C-artige Shell wie die `tcsh` angenehmer. Kommen Sie von Linux oder ist Ihnen der Umgang mit UNIX Systemen neu, so könnten Sie die `bash` probieren. Der Punkt ist, dass jede Shell ihre speziellen Eigenschaften hat, die mit Ihrer bevorzugten Arbeitsumgebung harmonieren können oder nicht. Sie müssen sich eine Shell aussuchen.

Ein verbreitetes Merkmal in Shells ist die Dateinamen-Vervollständigung. Sie müssen nur einige Buchstaben eines Kommandos oder eines Dateinamen eingeben und die Shell vervollständigt den Rest automatisch durch drücken der **Tab**-Taste. Hier ist ein Beispiel. Angenommen, Sie haben zwei Dateien `foobar` und `foo.bar`. Die Datei `foo.bar` möchten Sie löschen. Nun würden Sie an der Tastatur eingeben: `rm fo[Tab]. [Tab]`.

Die Shell würde dann `rm foo[BEEP].bar` ausgeben.

[BEEP] meint den Rechner-Piepser. Diesen gibt die Shell aus, um anzuzeigen, dass es den Dateinamen nicht vervollständigen konnte, da es mehrere Möglichkeiten gibt. Beide Dateien `foobar` und `foo.bar` beginnen mit `fo`, so konnte nur bis `foo` ergänzt werden. Nachdem Sie `.` eingaben und dann die **Tab**-Taste drückten, konnte die Shell den Rest für Sie ausfüllen.

Ein weiteres Merkmal der Shell ist der Gebrauch von Umgebungsvariablen. Dies sind veränderbare Schlüsselpaare im Umgebungsraum der Shell, die jedes von der Shell aufgerufene Programm lesen kann. Daher enthält der Umgebungsraum viele Konfigurationsdaten für Programme. Die folgende Liste zeigt verbreitete Umgebungsvariablen und was sie bedeuten:

Variable	Beschreibung
USER	Name des angemeldeten Benutzers.

Variable	Beschreibung
PATH	Liste mit Verzeichnissen (getrennt durch Doppelpunkt) zum Suchen nach Programmen.
DISPLAY	Der Name des X11-Bildschirms, auf dem Ausgaben erfolgen sollen.
SHELL	Die aktuelle Shell.
TERM	Name des Terminaltyps des Benutzers. Benutzt, um die Fähigkeiten des Terminals zu bestimmen.
TERMCAP	Datenbankeintrag der Terminal Escape Codes, benötigt um verschieden Terminalfunktionen auszuführen.
OSTYPE	Typ des Betriebsystems, beispielsweise FreeBSD.
MACHTYPE	Die CPU Architektur auf dem das System läuft.
EDITOR	Vom Benutzer bevorzugter Text-Editor.
PAGER	Vom Benutzer bevorzugter Text-Betrachter.
MANPATH	Liste mit Verzeichnissen (getrennt durch Doppelpunkt) zum Suchen nach Manualpages.

Das Setzen von Umgebungsvariablen funktioniert von Shell zu Shell unterschiedlich. Zum Beispiel benutzt man in C-artigen Shells wie der `tcsh` dazu `setenv`. Unter Bourne-Shells wie `sh` oder `bash` benutzen Sie zum Setzen von Umgebungsvariablen `export`. Um beispielsweise die Variable `EDITOR` mit `csh` oder `tcsh` auf `/usr/local/bin/emacs` zu setzen, setzen Sie das folgende Kommando ab:

```
% setenv EDITOR /usr/local/bin/emacs
```

Unter Bourne-Shells:

```
% export EDITOR="/usr/local/bin/emacs"
```

Sie können die meisten Shells Umgebungsvariablen expandieren lassen, in dem Sie in der Kommandozeile ein `$` davor eingeben. Zum Beispiel gibt `echo $TERM` aus, worauf `$TERM` gesetzt ist, weil die Shell `$TERM` expandiert und das Ergebnis an `echo` gibt.

Shells behandeln viele Spezialzeichen, so genannte Metazeichen, als besondere Darstellungen für Daten. Das allgemeinste ist das Zeichen `*`, das eine beliebige Anzahl Zeichen in einem Dateinamen repräsentiert. Diese Metazeichen können zum Vervollständigen von Dateinamen (Globbing) benutzt werden. Beispielsweise liefert das Kommando `echo *` nahezu das gleiche wie die Eingabe von `ls`, da die Shell alle Dateinamen die mit `*` übereinstimmen, an `echo` weitergibt.

Um zu verhindern, dass die Shell diese Sonderzeichen interpretiert, kann man sie schützen, indem man ihnen einen Backslash (`\`) voranstellt. `echo $TERM` gibt aus, auf was auch immer Ihr Terminal gesetzt ist. `echo \$TERM` gibt `$TERM` genauso aus, wie es hier steht.

4.9.1. Ändern der Shell

Der einfachste Weg Ihre Shell zu ändern, ist das Kommando `chsh` zu benutzen. `chsh` platziert Sie im Editor, welcher durch Ihre Umgebungsvariable `EDITOR` gesetzt ist, im `vi` wenn die Variable nicht gesetzt ist. Ändern Sie die Zeile mit “Shell:” entsprechend Ihren Wünschen.

Sie können auch `chsh` mit der Option `-s` aufrufen, dann wird Ihre Shell gesetzt, ohne dass Sie in einen Editor gelangen. Um Ihre Shell zum Beispiel auf die `bash` zu ändern, geben Sie das folgende Kommando ein:

```
% chsh -s /usr/local/bin/bash
```

> **Anmerkung:** Die von Ihnen gewünschte Shell *muss* in `/etc/shells` aufgeführt sein. Haben Sie eine Shell aus der Ports-Sammlung installiert, sollte das schon automatisch erledigt werden. Installierten Sie die Shell von Hand, so müssen Sie sie dort eintragen.
>
> Haben Sie beispielsweise die `bash` nach `/usr/local/bin` installiert, geben Sie Folgendes ein:
>
> ```
> # echo "/usr/local/bin/bash" >> /etc/shells
> ```
>
> Danach können Sie `chsh` aufrufen.

4.10. Text-Editoren

Eine großer Teil der Konfiguration wird bei FreeBSD durch das Editieren von Textdateien erledigt. Deshalb ist es eine gute Idee, mit einem Texteditor vertraut zu werden. FreeBSD hat ein paar davon im Basissystem und sehr viel mehr in der Ports-Sammlung.

Der am leichtesten und einfachsten zu erlernende Editor nennt sich **ee**, was für *easy editor* steht. Um **ee** zu starten, gibt man in der Kommandozeile `ee filename` ein, wobei *`filename`* den Namen der zu editierenden Datei darstellt. Um zum Beispiel `/etc/rc.conf` zu editieren, tippen Sie `ee /etc/rc.conf` ein. Einmal im Editor, finden Sie alle Editor-Funktionen oben im Display aufgelistet. Das Einschaltungszeichen `^` steht für die **Ctrl** (oder **Strg**) Taste, mit `^e` ist also die Tastenkombination **Ctrl**+**e** gemeint. Um **ee** zu verlassen, drücken Sie **Esc** und wählen dann `leave editor` aus. Der Editor fragt nach, ob Sie speichern möchten, wenn die Datei verändert wurde.

FreeBSD verfügt über leistungsfähigere Editoren wie **vi** als Teil des Basissystems, andere Editoren wie **emacs** oder **vim** sind Teil der Ports-Sammlung. Diese Editoren bieten höhere Funktionalität und Leistungsfähigkeit, jedoch auf Kosten einer etwas schwierigeren Erlernbarkeit. Wenn Sie viele Textdateien editieren, sparen Sie auf lange Sicht mehr Zeit durch das Erlernen von Editoren wie **vim** oder **emacs** ein.

Viele Anwendungen, die Dateien verändern oder Texteingabe erwarten, werden automatisch einen Texteditor öffnen. Um den Standardeditor zu ändern, setzen Sie die Umgebungsvariable `EDITOR`. Um mehr darüber zu erfahren, lesen Sie den Abschnitt Shells.

4.11. Geräte und Gerätedateien

Der Begriff Gerät wird meist in Verbindung mit Hardware wie Laufwerken, Druckern, Grafikkarten oder Tastaturen gebraucht. Der Großteil der Meldungen, die beim Booten von FreeBSD angezeigt werden, beziehen sich auf gefundene Geräte. Sie können sich die Bootmeldungen später in `/var/run/dmesg.boot` ansehen.

Gerätenamen, die Sie wahrscheinlich in den Bootmeldungen sehen werden, sind zum Beispiel `acd0`, das erste IDE CD-ROM oder `kbd0`, die Tastatur.

Auf die meisten Geräte wird unter UNIX Systemen über spezielle Gerätedateien im `/dev` Verzeichnis zugegriffen.

4.11.1. Anlegen von Gerätedateien

Wenn sie ein neues Gerät zu Ihrem System hinzufügen, oder die Unterstützung für zusätzliche Geräte kompilieren, müssen ein oder mehrere Gerätedateien erstellt werden.

4.11.1.1. DEVFS (Gerätedateisystem)

Das Gerätedateisystem DEVFS ermöglicht durch den Namensraum des Dateisystems Zugriff auf den Namensraum der Geräte im Kernel. Damit müssen Gerätedateien nicht mehr extra angelegt werden, sondern werden von DEVFS verwaltet.

Weitere Informationen finden Sie in devfs(5).

4.12. Binärformate

Um zu verstehen, warum FreeBSD das Format elf(5) benutzt, müssen Sie zunächst etwas über die drei gegenwärtig "dominanten" ausführbaren Formate für UNIX Systeme wissen:

- a.out(5)

  Das älteste und "klassische" Objektformat von UNIX Systemen. Es benutzt einen kurzen, kompakten Header mit einer magischen Nummer am Anfang, die oft benutzt wird, um das Format zu charakterisieren (weitere Details finden Sie unter a.out(5)). Es enthält drei geladene Segmente: .text, .data und .bss, sowie eine Symboltabelle und eine Stringtabelle.

- COFF

  Das Objektformat von SVR3. Der Header enthält nun eine "Sectiontable". Man kann also mit mehr als nur den Sections .text, .data und .bss arbeiten.

- elf(5)

  Der Nachfolger von COFF. Kennzeichnend sind mehrere Sections und mögliche 32-Bit- oder 64-Bit-Werte. Ein wesentlicher Nachteil: ELF wurde auch unter der Annahme entworfen, dass es nur eine ABI (Application Binary Interface) pro Systemarchitektur geben wird. Tatsächlich ist diese Annahme falsch – nicht einmal für die kommerzielle SYSV-Welt (in der es mindestens drei ABIs gibt: SVR4, Solaris, SCO) trifft sie zu.

  FreeBSD versucht, dieses Problem zu umgehen, indem ein Werkzeug bereitgestellt wird, um ausführbare Dateien im ELF-Format mit Informationen über die ABI zu versehen, zu der sie passen. Weitere Informationen finden Sie in der Manualpage brandelf(1).

FreeBSD kommt aus dem "klassischen" Lager und verwendete traditionell das Format a.out(5), eine Technik, die bereits über viele BSD-Releases hinweg eingesetzt und geprüft worden ist. Obwohl es bereits seit einiger Zeit möglich war, auf einem FreeBSD-System auch Binaries (und Kernel) im ELF-Format zu erstellen und auszuführen, widersetzte FreeBSD sich anfangs dem "Druck", auf ELF als Standardformat umzusteigen. Warum? Nun, als das Linux-Lager die schmerzhafte Umstellung auf ELF durchführte, ging es nicht so sehr darum, dem ausführbaren Format `a.out` zu entkommen, als dem unflexiblen, auf Sprungtabellen basierten Mechanismus für "Shared-Libraries" der die Konstruktion von Shared-Libraries für Hersteller und Entwickler gleichermaßen sehr kompliziert machte. Da die verfügbaren ELF-Werkzeuge eine Lösung für das Problem mit den Shared-Libraries anboten und ohnehin generell als "ein Schritt vorwärts" angesehen wurden, wurde der Aufwand für die Umstellung als notwendig akzeptiert und die Umstellung wurde durchgeführt. Unter FreeBSD ist der Mechanismus von

Shared-Libraries enger an den Stil des Shared-Library-Mechanismus von Suns SunOS™ angelehnt und von daher sehr einfach zu verwenden.

Ja, aber warum gibt es so viele unterschiedliche Formate?

In alter, grauer Vorzeit gab es simple Hardware. Diese simple Hardware unterstützte ein einfaches, kleines System. `a.out` war absolut passend für die Aufgabe, Binaries auf diesem simplen System (eine PDP-11) darzustellen. Als UNIX von diesem simplen System portiert wurde, wurde auch das `a.out`-Format beibehalten, weil es für die frühen Portierungen auf Architekturen wie den Motorola 68000 und VAX ausreichte.

Dann dachte sich ein schlauer Hardware-Ingenieur, dass, wenn er Software zwingen könnte, einige Tricks anzustellen, es ihm möglich wäre, ein paar Gatter im Design zu sparen, und seinen CPU-Kern schneller zu machen. Obgleich es dazu gebracht wurde, mit dieser neuen Art von Hardware (heute als RISC bekannt) zu arbeiten, war `a.out` für diese Hardware schlecht geeignet. Deshalb wurden viele neue Formate entwickelt, um eine bessere Leistung auf dieser Hardware zu erreichen, als mit dem begrenzten, simplen `a.out`-Format. Dinge wie COFF, ECOFF und einige andere obskure wurden erdacht und ihre Grenzen untersucht, bevor die Dinge sich in Richtung ELF entwickelten.

Hinzu kam, dass die Größe von Programmen gewaltig wurde und Festplatten sowie physikalischer Speicher immer noch relativ klein waren. Also wurde das Konzept von Shared-Libraries geboren. Das VM-System wurde auch immer fortgeschrittener. Obwohl bei jedem dieser Fortschritte das `a.out`-Format benutzt worden ist, wurde sein Nutzen mit jedem neuen Merkmal mehr und mehr gedehnt. Zusätzlich wollte man Dinge dynamisch zur Ausführungszeit laden, oder Teile ihres Programms nach der Initialisierung wegwerfen, um Hauptspeicher oder Swap-Speicher zu sparen. Programmiersprachen wurden immer fortschrittlicher und man wollte, dass Code automatisch vor der main-Funktion aufgerufen wird. Das `a.out`-Format wurde oft überarbeitet, um alle diese Dinge zu ermöglichen und sie funktionierten auch für einige Zeit. `a.out` konnte diese Probleme nicht ohne ein ständiges Ansteigen eines Overheads im Code und in der Komplexität handhaben. Obwohl ELF viele dieser Probleme löste, wäre es sehr aufwändig, ein System umzustellen, das im Grunde genommen funktionierte. Also musste ELF warten, bis es aufwändiger war, bei `a.out` zu bleiben, als zu ELF überzugehen.

Im Laufe der Zeit haben sich die Erstellungswerkzeuge, von denen FreeBSD seine Erstellungswerkzeuge abgeleitet hat (speziell der Assembler und der Loader), in zwei parallele Zweige entwickelt. Im FreeBSD-Zweig wurden Shared-Libraries hinzugefügt und einige Fehler behoben. Das GNU-Team, das diese Programme ursprünglich geschrieben hat, hat sie umgeschrieben und eine simplere Unterstützung zur Erstellung von Cross-Compilern durch beliebiges Einschalten verschiedener Formate usw. hinzugefügt. Viele Leute wollten Cross-Compiler für FreeBSD erstellen, aber sie hatten kein Glück, denn FreeBSD's ältere Sourcen für **as** und **ld** waren hierzu nicht geeignet. Die neuen GNU-Werkzeuge (**binutils**) unterstützen Cross-Compilierung, ELF, Shared-Libraries, C++-Erweiterungen und mehr. Weiterhin geben viele Hersteller ELF-Binaries heraus und es ist gut, wenn FreeBSD sie ausführen kann.

ELF ist ausdrucksfähiger als `a.out` und gestattet eine bessere Erweiterbarkeit des Basissystems. Die ELF-Werkzeuge werden besser gewartet und bieten Unterstützung von Cross-Compilierung, was für viele Leute wichtig ist. ELF mag etwas langsamer sein, als `a.out`, aber zu versuchen, das zu messen, könnte schwierig werden. Es gibt unzählige Details, in denen sich die beiden Formate unterscheiden, wie sie Pages abbilden, Initialisierungscode handhaben usw. Keins davon ist sehr wichtig, aber es sind Unterschiede. Irgendwann wird die Unterstützung für Programme im `a.out`-Format aus dem `GENERIC`-Kernel entfernt werden. Wenn es dann keinen oder kaum noch Bedarf für die Unterstützung dieses Formates gibt, werden die entsprechenden Routinen ganz entfernt werden.

4.13. Weitere Informationen

4.13.1. Manualpages

Die umfassendste Dokumentation rund um FreeBSD gibt es in Form von Manualpages. Annähernd jedes Programm im System bringt eine kurze Referenzdokumentation mit, die die grundsätzliche Funktion und verschiedene Parameter erklärt. Diese Dokumentationen kann man mit dem `man` Kommando benutzen. Die Benutzung des `man` Kommandos ist einfach:

```
% man Kommando
```

`Kommando` ist der Name des Kommandos, über das Sie etwas erfahren wollen. Um beispielsweise mehr über das Kommando `ls` zu lernen, geben Sie ein:

```
% man ls
```

Die Online-Dokumentation ist in nummerierte Sektionen unterteilt:

1. Benutzerkommandos.
2. Systemaufrufe und Fehlernummern.
3. Funktionen der C Bibliothek.
4. Gerätetreiber.
5. Dateiformate.
6. Spiele und andere Unterhaltung.
7. Verschiedene Informationen.
8. Systemverwaltung und -Kommandos.
9. Kernel Entwickler.

In einigen Fällen kann dasselbe Thema in mehreren Sektionen auftauchen. Es gibt zum Beispiel ein `chmod` Benutzerkommando und einen `chmod()` Systemaufruf. In diesem Fall können Sie dem `man` Kommando sagen, aus welcher Sektion Sie die Information erhalten möchten, indem Sie die Sektion mit angeben:

```
% man 1 chmod
```

Dies wird Ihnen die Manualpage für das Benutzerkommando `chmod` zeigen. Verweise auf eine Sektion der Manualpages werden traditionell in Klammern gesetzt. So bezieht sich chmod(1) auf das Benutzerkommando `chmod` und mit chmod(2) ist der Systemaufruf gemeint.

Das ist nett, wenn Sie den Namen eines Kommandos wissen, und lediglich wissen wollen, wie es zu benutzen ist. Aber was tun Sie, wenn Sie Sich nicht an den Namen des Kommandos erinnern können? Sie können mit `man` nach Schlüsselbegriffen in den Kommandobeschreibungen zu suchen, indem Sie den Parameter `-k` benutzen:

```
% man -k mail
```

Mit diesem Kommando bekommen Sie eine Liste der Kommandos, deren Beschreibung das Schlüsselwort “mail” enthält. Diese Funktionalität erhalten Sie auch, wenn Sie das Kommando `apropos` benutzen.

Nun, Sie schauen Sich alle die geheimnisvollen Kommandos in `/usr/bin` an, haben aber nicht den blassesten Schimmer, wozu die meisten davon gut sind? Dann rufen Sie doch das folgende Kommando auf:

```
% cd /usr/bin
% man -f *
```

Dasselbe erreichen Sie durch Eingabe von:

```
% cd /usr/bin
% whatis *
```

4.13.2. GNU Info Dateien

FreeBSD enthält viele Anwendungen und Utilities der Free Software Foundation (FSF). Zusätzlich zu den Manualpages bringen diese Programme ausführlichere Hypertext-Dokumente (`info` genannt) mit, welche man sich mit dem Kommando `info` ansehen kann. Wenn Sie **emacs** installiert haben, können Sie auch dessen info-Modus benutzen.

Um das Kommando info(1) zu benutzen, geben Sie ein:

```
% info
```

Eine kurze Einführung gibt es mit `h`; eine Befehlsreferenz erhalten Sie durch Eingabe von: `?`.

Fußnoten

1. Genau das ist mit `i386` gemeint. Auch wenn Ihr System keine Intel 386 CPU besitzt, wird `i386` ausgegeben. Es wird immer die Architektur und nicht der Typ des Prozessors ausgegeben.
2. Startskripten sind Programme, die FreeBSD automatisch bei jedem Startvorgang ausführt. Der Zweck der Skripte besteht darin, das System zu konfigurieren und nützliche Dienste im Hintergrund zu starten.
3. Eine recht technische und genaue Beschreibung der FreeBSD-Konsole und der Tastatur-Treiber finden Sie in den Hilfeseiten syscons(4), atkbd(4), vidcontrol(1) und kbdcontrol(1). Lesen Sie diese Seiten, wenn Sie an den Einzelheiten interessiert sind.
4. Das stimmt nicht ganz: Es gibt Fälle, in denen ein Prozess nicht unterbrochen werden kann. Wenn der Prozesss zum Beispiel eine Datei von einem anderen Rechner auf dem Netzwerk liest und dieser Rechner aus irgendwelchen Gründen nicht erreichbar ist (ausgeschaltet, oder ein Netzwerkfehler), dann ist der Prozess nicht zu unterbrechen. Wenn der Prozess den Lesezugriff nach einem Timeout von typischerweise zwei Minuten aufgibt, dann wir er beendet.

Kapitel 5. Installieren von Anwendungen: Pakete und Ports

Übersetzt von Uwe Pierau.

5.1. Übersicht

FreeBSD enthält sehr viele Systemwerkzeuge, die Teil des Basissystems sind. Allerdings sind Sie früher oder später auf Software Dritter angewiesen, damit Sie bestimmte Arbeiten durchführen können. Um diese Software zu installieren, stellt FreeBSD zwei sich ergänzende Methoden zur Verfügung: Die Ports-Sammlung (zur Installation aus dem Quellcode) sowie Pakete (auch als *Packages* bezeichnet, zur Installation von vorkompilierten binären Softwarepaketen). Sie können beide Methoden benutzen, um Ihre Lieblingsanwendungen von lokalen Medien oder über das Netzwerk zu installieren.

Dieses Kapitel behandelt die folgenden Themen:

- Die Installation binärer Softwarepakete.
- Der Bau Software Dritter aus dem Quellcode mithilfe der Ports-Sammlung.
- Wie zuvor installierte Pakete oder Ports entfernt werden.
- Wie Sie die Voreinstellungen der Ports-Sammlung überschreiben.
- Die Suche nach geeigneter Software.
- Wie Sie Ihre Anwendungen aktualisieren.

5.2. Installation von Software

Wenn Sie schon einmal ein UNIX System benutzt haben, werden Sie wissen, dass zusätzliche Software meist wie folgt installiert wird:

1. Download der Software, die als Quelltext oder im Binärformat vorliegen kann.
2. Auspacken der Software, die typischerweise ein mit compress(1), gzip(1) oder bzip2(1) komprimiertes Tar-Archiv enthält.
3. Durchsuchen der Dokumentation, die sich meist in Dateien wie `INSTALL`, `README` oder mehreren Dateien im Verzeichnis `doc/` befindet, nach Anweisungen, wie die Software zu installieren ist.
4. Kompilieren der Software wenn sie als Quelltext vorliegt. Dazu müssen Sie vielleicht das `Makefile` anpassen, oder `configure` laufen lassen, oder andere Arbeiten durchführen.
5. Testen und installieren der Software.

Das beschreibt aber nur den optimalen Fall. Wenn Sie Software installieren, die nicht speziell für FreeBSD geschrieben wurde, müssen Sie vielleicht sogar den Quelltext anpassen, damit die Software funktioniert.

Wenn Sie unbedingt wollen, können Sie mit FreeBSD Software nach der “althergebrachten” Methode installieren. Mit Paketen oder Ports bietet Ihnen FreeBSD allerdings zwei Methoden an, die Ihnen sehr viel Zeit sparen können. Zurzeit werden über 24,000 Anwendungen Dritter über diese Methoden zur Verfügung gestellt.

Das FreeBSD-Paket einer Anwendung besteht aus einer einzigen Datei, die Sie sich herunterladen müssen. Das Paket enthält schon übersetzte Kommandos der Anwendung, sowie zusätzliche Konfigurationsdateien oder Dokumentation. Zur Handhabung der Pakete stellt FreeBSD-Kommandos wie pkg_add(1), pkg_delete(1) oder pkg_info(1) zur Verfügung. Mit diesem System können neue Anwendungen mit einem Kommando, `pkg_add`, installiert werden.

Der FreeBSD-Port einer Anwendung ist eine Sammlung von Dateien, die das Kompilieren der Quelltexte einer Anwendung automatisieren.

Die Dateien eines Ports führen für Sie alle oben aufgeführten Schritte zum Installieren einer Anwendung durch. Mit einigen wenigen Kommandos wird der Quellcode der Anwendung automatisch heruntergeladen, ausgepackt, gepatcht, übersetzt und installiert.

Tatsächlich kann das Portsystem auch dazu benutzt werden, Pakete zu generieren, die Sie mit den gleich beschriebenen Kommandos, wie `pkg_add`, manipulieren können.

Pakete und Ports beachten Abhängigkeiten zwischen Anwendungen. Angenommen, Sie wollen eine Anwendung installieren, die von einer Bibliothek abhängt und die Anwendung wie die Bibliothek sind als Paket oder Port für FreeBSD verfügbar. Wenn Sie `pkg_add` oder das Portsystem benutzen, um die Anwendung zu installieren, werden Sie bemerken, dass die Bibliothek zuerst installiert wird, wenn sie nicht schon vorher installiert war.

Sie werden sich fragen, warum FreeBSD-Pakete und -Ports unterstützt, wo doch beide Methoden fast gleiches leisten. Beide Methoden haben ihre Stärken und welche Sie einsetzen, hängt letztlich von Ihren Vorlieben ab.

Vorteile von Paketen

- Das komprimierte Paket einer Anwendung ist normalerweise kleiner als das komprimierte Archiv der Quelltexte.
- Pakete müssen nicht mehr kompiliert werden. Dies ist ein Vorteil, wenn Sie große Pakete, wie **Mozilla**, **KDE** oder **GNOME** auf langsamen Maschinen installieren.
- Wenn Sie Pakete verwenden, brauchen Sie nicht zu verstehen, wie Sie Software unter FreeBSD kompilieren.

Vorteile von Ports

- Da die Pakete auf möglichst vielen System laufen sollen, werden Optionen beim Übersetzen zurückhaltend gesetzt. Wenn Sie eine Anwendung über die Ports installieren, können Sie die Angabe der Optionen optimieren. Zum Beispiel können Sie spezifischen Code für Pentium 4 oder Athlon Prozessoren erzeugen.
- Die Eigenschaften einiger Anwendungen werden über Optionen zum Zeitpunkt des Übersetzens festgelegt. **Apache** kann zum Beispiel über viele eingebaute Optionen konfiguriert werden. Wenn Sie das Portsystem benutzen, können Sie die Vorgaben für die Optionen überschreiben.

  Für einige Fälle existieren verschiedene Pakete einer Anwendung, die beim Übersetzen unterschiedlich konfiguriert wurden. Für **Ghostscript** gibt es ein `ghostscript`-Paket und ein `ghostscript-nox11`-Paket, die sich durch die X11 Unterstützung unterscheiden. Diese grobe Unterscheidung ist mit dem Paketsystem möglich, wird aber schnell unhandlich, wenn eine Anwendung mehr als ein oder zwei Optionen zum Zeitpunkt des Übersetzens besitzt.
- Die Lizenzbestimmungen mancher Software verbietet ein Verbreiten in binärer Form. Diese Software muss als Quelltext ausgeliefert werden.
- Einige Leute trauen binären Distributionen nicht. Wenn Sie den Quelltext besitzen, können Sie sich diesen (zumindest theoretisch) durchlesen und nach möglichen Problemen durchsuchen.

- Wenn Sie eigene Anpassungen besitzen, benötigen Sie den Quelltext, um diese anzuwenden.
- Manch einer besitzt gerne den Quelltext, um ihn zu lesen, wenn es einmal langweilig ist, ihn zu hacken, oder sich einfach ein paar Sachen abzugucken (natürlich nur, wenn es die Lizenzbestimmungen erlauben).

Wenn Sie über aktualisierte Ports informiert sein wollen, lesen Sie bitte die Mailinglisten FreeBSD ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports) und FreeBSD ports bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs).

> **Warnung:** Bevor Sie eine Anwendung installieren, sollten Sie auf der Seite http://vuxml.FreeBSD.org/ über mögliche Sicherheitsprobleme mit der Anwendung informieren.
>
> Die Anwendung `ports-mgmt/portaudit` prüft automatisch alle installierten Anwendungen auf bekannte Sicherheitslöcher. Vor dem Bau eines Ports findet ebenfalls eine Prüfung statt. Installierte Pakete prüfen Sie mit dem Kommando `portaudit -F -a`.

Der Rest dieses Kapitels beschreibt, wie Sie Software Dritter mit Paketen oder Ports auf einem FreeBSD-System installieren und verwalten.

5.3. Suchen einer Anwendung

Bevor Sie eine Anwendung installieren, müssen Sie deren Art und Namen kennen.

Die Anzahl der nach FreeBSD portierten Anwendungen steigt ständig. Zum Glück gibt es einige Wege, die richtige zu finden.

- Eine aktuelle Liste verfügbarer Anwendungen, die sich auch durchsuchen lässt, finden Sie unter http://www.FreeBSD.org/ports/ (http://www.FreeBSD.org/ports/index.html). Die Anwendungen sind in Kategorien unterteilt und Sie können sich alle Anwendungen einer Kategorie anzeigen lassen. Wenn Sie den Namen der Anwendung kennen, können Sie natürlich auch direkt nach dem Namen suchen.
- FreshPorts, das von Dan Langille gepflegt wird, erreichen Sie unter http://www.FreshPorts.org/. FreshPorts verfolgt Änderungen an Anwendungen aus den Ports. Mit FreshPorts können Sie ein oder mehrere Ports beobachten und sich eine E-Mail schicken lassen, wenn ein Port aktualisiert wird.
- Wenn Sie den Namen einer Anwendung nicht kennen, versuchen Sie eine Webseite wie FreshMeat (http://www.freshmeat.net/), um eine passende Anwendung zu finden. Schauen Sie dann auf der FreeBSD-Webseite nach, ob die Anwendung schon portiert wurde.
- Wenn Sie den Portnamen kennen und nur nach der Kategorie suchen wollen, verwenden Sie das Kommando whereis(1). Geben Sie einfach `whereis` *`Datei`* ein. *`Datei`* ist der Name des Programms, das Sie suchen:

  ```
  # whereis lsof
  lsof: /usr/ports/sysutils/lsof
  ```

  Damit haben wir herausgefunden, dass sich `lsof`, ein Systemwerkzeug, im Verzeichnis `/usr/ports/sysutils/lsof` befindet.

- Auch mit einem einfachen echo(1)-Befehl können Sie herausfinden, wo Sie einen bestimmten Port finden. Dazu ein Beispiel:

  ```
  # echo /usr/ports/*/*lsof*
            /usr/ports/sysutils/lsof
  ```

  Beachten Sie aber, dass dieser Befehl auch alle Dateien im Verzeichnis `/usr/ports/distfiles` findet, auf die der angegebene Suchbegriff passt.

- Ein weiterer Weg, einen bestimmten Port zu finden, ist es, die eingebaute Suchfunktion der Ports-Sammlung zu benutzen. Dazu muss Ihr Arbeitsverzeichnis `/usr/ports` sein. In diesem Verzeichnis rufen Sie `make search name=`*`Anwendungsname`* auf, wobei *`Anwendungsname`* der Name der gesuchten Anwendung ist. Wenn Sie zum Beispiel nach `lsof` suchen:

  ```
  # cd /usr/ports
  # make search name=lsof
  Port:   lsof-4.56.4
  Path:   /usr/ports/sysutils/lsof
  Info:   Lists information about open files (similar to fstat(1))
  Maint:  obrien@FreeBSD.org
  Index:  sysutils
  B-deps:
  R-deps:
  ```

  Der Teil der Ausgabe der Sie interessiert ist die Zeile, die mit “Path:” beginnt, weil sie Ihnen sagt, wo der Port zu finden ist. Die anderen Informationen werden zum Installieren des Ports nicht direkt benötigt, Sie brauchen sich darum jetzt nicht weiter zu kümmern.

  Erweiterte Suchen führen Sie mit dem Kommando `make search key=`*`Text`* aus. Damit werden Portnamen, Kommentare, Beschreibungen und Abhängigkeiten nach *`Text`* durchsucht. Dies kann sehr nützlich sein, wenn Sie den Namen des Programms, nach dem Sie suchen, nicht kennen.

  In beiden Fällen wird Groß- und Kleinschreibung bei der Suche ignoriert. Die Suche nach “LSOF” wird dieselben Ergebnisse wie die Suche nach “lsof” liefern.

5.4. Benutzen des Paketsystems

Beigesteuert von Chern Lee.

Es gibt viele unterschiedliche Werkzeuge um Pakete in FreeBSD zu verwalten:

- Auf einem laufenden System kann `sysinstall` benutzt werden, um Pakete zu installieren, zu löschen und verfügbare und installierte anzuzeigen. Weitere Informationen finden Sie unter Abschnitt 2.10.11.
- Die Paketverwaltungswerkzeuge der Kommandozeile sind die Themen von diesem Kapitel.

5.4.1. Installieren eines Pakets

Mit pkg_add(1) können Sie ein FreeBSD-Paket von einer lokalen Datei oder über das Netzwerk installieren.

Beispiel 5-1. Download vor Installation eines Pakets

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-     This machine is in Vienna, VA, USA, hosted by Verio.
230-         Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375 bytes).
100% |***********************************************| 92375       00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

Wenn Sie die Pakete nicht lokal vorliegen haben (zum Beispiel auf den FreeBSD-CD-ROMs), ist es wahrscheinlich einfacher den Schalter `-r` von pkg_add(1) zu verwenden. Das Werkzeug bestimmt dann automatisch das nötige Objektformat und die richtige Version des Pakets, lädt dieses dann von einem FTP-Server und installiert das Paket.

```
# pkg_add -r lsof
```

Das obige Beispiel würde ohne weitere Interaktion das richtige Paket herunterladen und installieren. Pakete werden vom FreeBSD-Hauptserver heruntergeladen. Wenn Sie anderen Server verwenden möchten, geben Sie den Server in der Umgebungsvariablen `PACKAGESITE` an. Die Dateien werden mit fetch(3), das Umgebungsvariablen wie `FTP_PASSIVE_MODE`, `FTP_PROXY` und `FTP_PASSWORD` berücksichtigt, heruntergeladen. Wenn Sie durch eine Firewall geschützt werden, müssen Sie vielleicht eine oder mehrere dieser Umgebungsvariablen setzen oder einen FTP oder HTTP Proxy verwenden. Eine Liste der unterstützten Umgebungsvariablen finden Sie in fetch(3). Beachten Sie, dass im obigen Beispiel `lsof` anstelle von `lsof-4.56.4` verwendet wird. Wenn Sie pkg_add(1) zum Herunterladen eines Pakets verwenden, darf die Versionsnummer des Pakets nicht angegeben werden, da automatisch die neuste Version der Anwendung geholt wird.

> **Anmerkung:** Unter FreeBSD-CURRENT oder FreeBSD-STABLE holt pkg_add(1) die neuste Version einer Anwendung, unter einer Release holt pkg_add(1) die Version der Anwendung, die im Release enthalten ist. Sie können dies ändern, indem Sie die Umgebungsvariable `PACKAGESITE` überschreiben. Wenn Sie bespielsweise FreeBSD 8.1-RELEASE installiert haben, versucht pkg_add(1) in der Voreinstellung die Pakete von `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8.1-release/Latest/` zu laden. Wollen Sie pkg_add(1) dazu zwingen, nur FreeBSD 8-STABLE-Pakete herunterzuladen, setzen Sie die Umgebungsvariable `PACKAGESITE` auf `ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-8-stable/Latest/`.

Pakete werden im `.tgz`- und `.tbz`-Format ausgeliefert. Sie finden Sie unter ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/ oder auf der FreeBSD-CD-ROM-Distribution. Jede CD der FreeBSD Distribution (oder des PowerPaks) enthält Pakete im Verzeichnis `/packages`. Die Struktur des Paketbaums entspricht dem `/usr/ports` Baum. Jede Kategorie besitzt ein eigenes Verzeichnis und alle Pakete befinden sich im Verzeichnis `All`.

Die Verzeichnisstruktur des Paketbaums ist ein Abbild der Ports, da beide Systeme eng zusammenarbeiten.

5.4.2. Verwalten von Paketen

pkg_info(1) zeigt alle installierten Pakete und deren Beschreibung an.

```
# pkg_info
cvsup-16.1          A general network file distribution system optimized for CV
docbook-1.2         Meta-port for the different versions of the DocBook DTD
...
```

pkg_version(1) vergleicht die Version installierter Pakete mit der Version aus der Ports-Sammlung.

```
# pkg_version
cvsup                       =
docbook                     =
...
```

Die Symbole in der zweiten Spalte zeigen das Alter des Pakets im Vergleich zu der lokalen Version aus der Ports-Sammlung an.

Symbol	Bedeutung
=	Die Version des installierten Paketes stimmt mit der Version aus der lokalen Ports-Sammlung überein.
<	Die installierte Version ist älter als die der verfügbaren Version aus der Ports-Sammlung.
>	Die installierte Version ist neuer als die aus der Ports-Sammlung (Eventuell ist die lokale Ports-Sammlung veraltet).
?	Das installierte Paket konnte in der Ports-Sammlung nicht gefunden werden. Das kann dadurch hervorgerufen werden, dass ein installierter Port aus der Ports-Sammlung entfernt wurde oder einen neuen Namen erhalten hat.
*	In der Ports-Sammlung befinden sich mehrere Versionen der Anwendung.
!	Das installierte Paket ist zwar im Index enthalten, aus irgendeinem Grund war `pkg_version` aber dennoch nicht in der Lage, die Versionsnummer des installierten Pakets mit der Versionsnummer des entsprechenden Eintrags im Index zu vergleichen.

5.4.3. Entfernen eines Pakets

Um ein zuvor installiertes Paket zu entfernen, benutzen Sie das Werkzeug pkg_delete(1).

```
# pkg_delete xchat-1.7.1
```

Beachten Sie, dass pkg_delete(1) die vollständige Bezeichnung des Pakets benötigt (also Paketname *und* Versionsnummer). Die Eingabe von `xchat` (anstelle von `xchat-1.7.1`) ist daher nicht ausreichend. Zwar können Sie die Versionsnummer eines installierten Pakets mit pkg_version(1) herausfinden, es ist aber auch möglich, ein Paket zu deinstallieren, ohne die exakte Versionsnummer zu kennen, wenn Sie Wildcards einsetzen:

```
# pkg_delete xchat\*
```

In diesem Beispiel werden alle Pakete gelöscht, deren Name mit `xchat` beginnt.

5.4.4. Verschiedenes

Informationen über alle installierte Pakete werden in `/var/db/pkg` abgelegt. Das Verzeichnis enthält Dateien, in denen sich die Beschreibungen der Pakete und Listen von Dateien, die zu einem Paket gehören, befinden.

5.5. Benutzen der Ports-Sammlung

Die folgenden Abschnitte stellen die grundlegenden Anweisungen vor, um Anwendungen aus der Ports-Sammlung auf Ihren Rechner zu installieren oder zu löschen. ports(7) enthält eine Auflistung aller verfügbaren `make`-Targets und Umgebungsvariablen.

5.5.1. Installation der Ports-Sammlung

Bevor Sie einen Port installieren können, müssen Sie zuerst die Ports-Sammlung installieren, die aus Makefiles, Patches und Beschreibungen besteht. Die Ports-Sammlung wird für gewöhnlich unter `/usr/ports` installiert.

Bei der FreeBSD-Installation hatten Sie in **sysinstall** die Möglichkeit, die Ports-Sammlung zu installieren. Wenn Sie die Sammlung damals nicht installiert haben, können Sie das mit den folgenden Anweisungen nachholen:

Installieren mit CVSup

Dies ist eine schnelle Methode, um die Ports-Sammlung zu installieren und zu aktualisieren. **CVSup** wird im Abschnitt Benutzen von CVSup des Handbuchs beschrieben.

> **Anmerkung:** Die im Basissystem enthaltene Variante des **CVSup**-Protokolls heißt **csup**.

Achten Sie darauf, dass das Verzeichnis `/usr/ports` leer ist, bevor Sie **csup** das erste Mal ausführen! Haben Sie die Ports-Sammlung zuvor schon aus einer anderen Quelle installiert, wird **csup** bereits aus dem Repository entfernte Patches nicht aus der lokalen Kopie der Ports-Sammlung löschen.

1. Rufen Sie `csup` auf:

   ```
   # csup -L 2 -h cvsup.FreeBSD.org /usr/share/examples/cvsup/ports-supfile
   ```

Ersetzen Sie *cvsup.FreeBSD.org* durch einen **CVSup**-Server in Ihrer Nähe. Eine vollständige Liste der **CVSup**-Spiegel finden Sie im Abschnitt CVSup-Server des Handbuchs.

> **Anmerkung:** Sie sollten sich eine an Ihre Bedürfnisse angepasste `ports-supfile` erstellen, um so beispielsweise zu vermeiden, dass Sie bei jedem Aufruf von **CVSup** wieder die Parameterliste übergeben müssen.

1. Dazu kopieren Sie zuerst als `root` die Datei `/usr/share/examples/cvsup/ports-supfile` nach `/root` oder in Ihr Heimatverzeichnis.
2. Danach müssen Sie die Datei `ports-supfile` anpassen.
3. Dazu ersetzen Sie *cvsup.FreeBSD.org* durch einen **CVSup**-Server in Ihrer Nähe. Eine vollständige Liste der **CVSup**-Spiegel finden Sie im Abschnitt CVSup-Server des Handbuchs.
4. Nun können Sie `csup` mit folgender Syntax starten:

   ```
   # csup -L 2 /root/ports-supfile
   ```

2. Mit csup(1) können Sie später auch die Ports-Sammlung aktualisieren. Die installierten Ports werden mit diesem Kommando allerdings nicht aktualisiert.

Installieren mit Portsnap

Bei **Portsnap** handelt es sich um ein alternatives System zur Distribution der Ports-Sammlung. Eine detaillierte Beschreibung von **Portsnap** finden Sie im Abschnitt
Portsnap: Ein Werkzeug zur Aktualisierung der Ports-Sammlung des Handbuchs.

1. Laden Sie einen komprimierten Snapshot der Ports-Sammlung in das Verzeichnis `/var/db/portsnap` herunter. Danach können Sie die Internetverbindung trennen, wenn Sie dies wünschen.

   ```
   # portsnap fetch
   ```

2. Wenn Sie **Portsnap** das erste Mal verwenden, müssen Sie den Snapshot nach `/usr/ports` extrahieren:

   ```
   # portsnap extract
   ```

   Ist die Ports-Sammlung bereits installiert, und Sie wollen diese nur aktualisieren, führen Sie stattdessen folgenden Befehl aus:

   ```
   # portsnap update
   ```

Installieren mit sysinstall

Nicht zuletzt ist es auch möglich, die Ports-Sammlung über **sysinstall** zu installieren. Beachten Sie dabei aber, dass bei dieser Methode nicht die aktuellste Version der Ports-Sammlung, sondern die Version, die zum Zeitpunkt der Veröffentlichung der installierten FreeBSD-Version aktuell war, installiert wird. Haben Sie Zugriff auf das Internet, so sollten Sie daher stets eine der weiter oben beschriebenen Methoden verwenden, um die Ports-Sammlung zu installieren.

1. Führen Sie als `root` `sysinstall` aus:

   ```
   # sysinstall
   ```

2. Wählen Sie den Punkt Configure aus und drücken Sie **Enter**.

3. Wählen Sie dann Distributions aus und drücken Sie **Enter**.
4. In diesem Menü wählen Sie ports aus und drücken die **Leertaste**.
5. Danach wählen Sie `Exit` aus und drücken **Enter**.
6. Legen Sie nun ein geeignetes Installationsmedium, wie CD-ROM oder FTP, fest.
7. Wählen Sie nun `Exit` aus und drücken **Enter**.
8. Verlassen Sie **sysinstall** mit **X**.

5.5.2. Ports installieren

Was ist mit einem “Gerüst” im Zusammenhang mit der Ports-Sammlung gemeint? In aller Kürze: ein Gerüst eines Ports ist ein minimaler Satz von Dateien, mit denen das FreeBSD-System eine Anwendung sauber übersetzen und installieren kann. Ein jeder Port beinhaltet:

- Eine Datei `Makefile`. Das `Makefile` enthält verschiedene Anweisungen, die spezifizieren, wie eine Anwendung kompiliert wird und wo sie auf Ihrem System installiert werden sollte.
- Eine Datei `distinfo`. Diese enthält Informationen, welche Dateien heruntergeladen werden müssen sowie deren MD5-Prüfsummen (die Sie mit sha256(1) überprüfen können, um sicher zu gehen, dass diese Dateien während des Herunterladens nicht beschädigt wurden).
- Ein `files` Verzeichnis. Hierin liegen Patches, welche das Übersetzen und Installieren der Anwendung ermöglichen. Patches sind im Wesentlichen kleine Dateien, die Änderungen an speziellen Dateien spezifizieren. Sie liegen als reiner Text vor und sagen ungefähr: “Lösche Zeile 10” oder “Ändere Zeile 26 zu ...”. Patches sind auch bekannt unter dem Namen “diffs”, weil Sie mit dem Programm diff(1) erstellt werden.

  Dieses Verzeichnis kann auch noch andere Dateien enthalten, welche zum Bauen des Ports benutzt werden.
- Eine Datei `pkg-descr`. Eine ausführlichere, oft mehrzeilige Beschreibung der Anwendung.
- Eine Datei `pkg-plist`. Das ist eine Liste aller Dateien, die durch diesen Port installiert werden. Außerdem sind hier Informationen enthalten, die zum Entfernen des Ports benötigt werden.

Einige Ports besitzen noch andere Dateien, wie `pkg-message`, die vom Portsystem benutzt werden, um spezielle Situationen zu handhaben. Wenn Sie mehr über diese Dateien oder das Port-System erfahren sollen, lesen Sie bitte das FreeBSD Porter’s Handbook (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/porters-handbook/index.html).

Ein Port enthält lediglich Anweisungen, wie der Quelltext zu bauen ist, nicht aber den eigentlichen Quelltext. Den Quelltext erhalten Sie von einer CD-ROM oder aus dem Internet. Quelltexte werden in einem Format nach Wahl des jeweiligen Software-Autors ausgeliefert. Häufig ist dies ein gezipptes Tar-Archiv, aber es kann auch mit einem anderen Tool komprimiert oder gar nicht komprimiert sein. Der Quelltext, in welcher Form er auch immer vorliegen mag, wird “Distfile” genannt. Die zwei Methoden, mit denen ein Port installiert wird, werden unten besprochen.

> **Anmerkung:** Zum Installieren von Ports müssen Sie als Benutzer `root` angemeldet sein.

Warnung: Stellen Sie sicher, dass die Ports-Sammlung aktuell ist, bevor Sie einen Port installieren. Informieren Sie sich vorher zusätzlich unter http://vuxml.FreeBSD.org/ über mögliche Sicherheitsprobleme des zu installierenden Ports.

Vor der Installation kann **portaudit** eine neue Anwendung automatisch auf Sicherheitslöcher prüfen. Das Werkzeug befindet sich in der Ports-Sammlung (`ports-mgmt/portaudit`). Vor der Installation einer neuen Anwendung sollten Sie mit `portaudit -F` die Sicherheitsdatenbank aktualisieren. Die täglich laufende Sicherheitsprüfung des Systems aktualisiert die Datenbank und prüft installierte Anwendungen auf vorhandene Sicherheitslöcher. Weiteres erfahren Sie in den Hilfeseiten portaudit(1) und periodic(8).

Die Ports-Sammlung geht davon, dass Ihr System über eine funktionierende Internetverbindung verfügt. Ist dies nicht der Fall, müssen Sie eine Kopie des zu installierenden Distfiles manuell nach `/usr/ports/distfiles` kopieren.

Dazu wechseln Sie als erstes in das Verzeichnis des Ports, den Sie installieren wollen:

```
# cd /usr/ports/sysutils/lsof
```

Im Verzeichnis `lsof` kann man das Gerüst erkennen. Der nächste Schritt ist das Übersetzen (auch Bauen genannt) des Ports durch die Eingabe des Befehls `make`:

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.57
...
[Ausgabe des Auspackens weggelassen]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
===>  Patching for lsof-4.57
===>  Applying FreeBSD patches for lsof-4.57
===>  Configuring for lsof-4.57
...
[configure-Ausgabe weggelassen]
...
===>  Building for lsof-4.57
...
[Ausgabe der Übersetzung weggelassen]
...
#
```

Ist die Übersetzungsprozedur beendet, landen Sie wiederum in der Kommandozeile und können das Programm im nächsten Schritt installieren. Dazu verwenden Sie den Befehl `make install`:

```
# make install
===>  Installing for lsof-4.57
...
[Ausgabe der Installation weggelassen]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.57
===>   Registering installation for lsof-4.57
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
```

```
      increased privileges.
#
```

Nachdem die Installation abgeschlossen ist, können Sie die gerade installierte Anwendung starten. Da `lsof` eine Anwendung ist, die mit erhöhten Rechten läuft, wird eine Sicherheitswarnung angezeigt. Sie sollten alle Warnungen während des Baus und der Installation eines Ports beachten.

Es ist eine gute Idee, das Unterverzeichnis `work` nach erfolgter Installation wieder zu löschen. Einerseits gewinnen Sie dadurch Speicherplatz, andererseits könnte es sonst zu Problemen bei der Aktualisierung des Ports auf eine neuere Version kommen.

```
# make clean
===>  Cleaning for lsof-4.57
#
```

> **Anmerkung:** Sie können zwei Schritte sparen, wenn Sie gleich `make install clean` anstelle von `make`, `make install` und `make clean` eingeben.

> **Anmerkung:** Um die Suche nach Kommandos zu beschleunigen, speichern einige Shells eine Liste der verfügbaren Kommandos in den durch die Umgebungsvariable `PATH` gegebenen Verzeichnissen. Nach der Installation eines Ports müssen Sie in einer solchen Shell vielleicht das Kommando `rehash` absetzen, um die neu installierten Kommandos benutzen zu können. Das Kommando `rehash` gibt es in Shells wie der `tcsh`. Unter Shells wie der `sh` benutzen Sie das Kommando `hash -r`. Weiteres entnehmen Sie bitte der Dokumentation Ihrer Shell.

Einige von Dritten angebotenen DVD-ROM-Produkte wie das FreeBSD Toolkit von der FreeBSD Mall (http://www.freebsdmall.com/) enthalten auch Distfiles (komprimierte Quellcodepakete). Diese lassen sich über die Ports-Sammlung installieren. Dazu hängen Sie die DVD-ROM unter `/cdrom` in den Verzeichnisbaum ein. Wenn Sie einen anderen Mountpunkt verwenden, sollten Sie die make-Variable `CD_MOUNTPTS` setzen, damit die auf der DVD-ROM enthaltenen Distfiles automatisch verwendet werden.

> **Anmerkung:** Beachten Sie bitte, dass die Lizenzen einiger Ports die Einbeziehung auf der CD-ROM verbieten. Das kann verschiedene Gründe haben. Beispielsweise eine Registrierung vor dem Herunterladen erforderlich oder die Weiterverteilung ist verboten. Wenn Sie einen Port installieren wollen, der nicht auf der CD-ROM enthalten ist, müssen Sie online sein.

Die Ports-Sammlung benutzt zum Herunterladen von Dateien fetch(3), das Umgebungsvariablen wie `FTP_PASSIVE_MODE`, `FTP_PROXY` und `FTP_PASSWORD` berücksichtigt. Wenn Sie durch eine Firewall geschützt werden, müssen Sie vielleicht eine oder mehrere dieser Umgebungsvariablen setzen, oder einen FTP oder HTTP Proxy verwenden. Eine Liste der unterstützten Umgebungsvariablen finden Sie in fetch(3).

Benutzer ohne eine ständige Internet-Verbindung werden das Kommando `make fetch` zu schätzen wissen. Das Kommando lädt alle benötigten Dateien eines Ports herunter. Sie können das Kommando im Verzeichnis `/usr/ports` laufen lassen. In diesem Fall werden *alle* Dateien heruntergeladen. Es ist auch möglich, `make fetch` nur in einem Teil des Baums, wie `/usr/ports/net`, aufzurufen. Die Dateien von allen abhängigen Ports werden

mit diesem Kommando allerdings nicht heruntergeladen. Wenn Sie diese Dateien ebenfalls herunterladen wollen, ersetzen Sie im Kommando `fetch` durch `fetch-recursive`.

> **Anmerkung:** Abhängig davon, in welchem Verzeichnis Sie `make` aufrufen, können Sie analog zu `make fetch` die Ports einer Kategorie oder alle Ports bauen. Beachten Sie allerdings, dass manche Ports nicht zusammen installiert werden können. Weiterhin gibt es Fälle, in denen zwei Ports unterschiedliche Inhalte in derselben Datei speichern wollen.

Manchmal ist es erforderlich, die benötigten Dateien von einem anderen Ort als den im Port vorgesehenen herunterzuladen. Der Ort wird durch die Variable `MASTER_SITES` vorgegeben, die Sie wie folgt überschreiben können:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

Im Beispiel wurde `MASTER_SITES` mit dem Wert `ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/` überschrieben.

> **Anmerkung:** Einige Ports besitzen Optionen, mit denen Sie zusätzliche Funktionen oder Sicherheitsoptionen einstellen können (oder manchmal auch müssen). Zusätzliche Optionen können beispielsweise für `www/firefox`, `security/gpgme` und `mail/sylpheed-claws` angegeben werden. Wenn ein Port über zusätzliche Optionen verfügt, werden diese beim Bau des Ports auf der Konsole ausgegeben.

5.5.2.1. Vorgabe-Verzeichnisse ändern

Manchmal ist es nützlich (oder erforderlich), in anderen Verzeichnissen zu arbeiten. Die Verzeichnisse können Sie mit den Variablen `WRKDIRPREFIX` und `PREFIX` einstellen. Die Variable `WRKDIRPREFIX` gibt das Bauverzeichnis an:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

Dieses Kommando baut den Port in `/usr/home/example/ports` und installiert ihn unter `/usr/local`.

Die Variable `PREFIX` legt das Installations-Verzeichnis fest:

```
# make PREFIX=/usr/home/example/local install
```

In diesem Beispiel wird der Port unter `/usr/ports` gebaut und nach `/usr/home/example/local` installiert.

Sie können beide Variablen auch zusammen benutzen:

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

Die Kommandozeile ist zu lang, um sie hier komplett wiederzugeben, aber Sie sollten die zugrunde liegende Idee erkennen.

5.5.2.2. Probleme mit `imake`

Einige Ports, welche imake(1) (Teil des X-Window-Systems) benutzen, funktionieren nicht gut mit `PREFIX` und bestehen darauf, unter `/usr/X11R6` installiert zu werden. In ähnlicher Weise verhalten sich einige Perl-Ports, die `PREFIX` ignorieren und sich in den Perl-Verzeichnisbaum installieren. Zu erreichen, dass solche Ports `PREFIX` beachten, ist schwierig oder sogar unmöglich.

5.5.2.3. Ports rekonfigurieren

Beim Bau einiger Ports erhalten Sie ein ncurses-basiertes Menü, über dessen Optionen Sie den Bau dieser Ports beeinflussen können. Es gibt diverse Möglichkeiten, dieses Menü nach dem Bau eines Ports erneut aufzurufen, um beispielsweise Optionen zu entfernen, hinzuzufügen oder anzupassen. Sie können beispielsweise in das Verzeichnis des Ports wechseln und dort den Befehl `make config` eingeben, wodurch das Menü mit den ursprünglichen gewählten Optionen erneut aufgerufen wird. Eine andere Möglichkeit bietet der Befehl `make showconfig`, mit dem Sie eine Liste aller Konfigurationsoptionen dieses Ports aufrufen. Eine weitere Alternative bietet der Befehl `make rmconfig`, der die von Ihnen ursprünglich gewählten Optionen zurücksetzt und es Ihnen dadurch ermöglicht, die Konfiguration erneut zu beginnen. Die eben erwähnten Optionen (und viele andere) werden ausführlich in der Manualpage ports(7) beschrieben.

5.5.3. Entfernen installierter Ports

Da Sie nun wissen, wie man einen Port installiert, wollen Sie sicher auch wissen, wie man ein über einen Port installiertes Programm wieder deinstallieren kann. Ports werden analog zu Paketen mit pkg_delete(1) deinstalliert (Lesen Sie sich den Abschnitt Benutzen des Paketsystems des Handbuchs durch, wenn Sie weitere Informationen benötigen.). Für das vorhin installierte Programm `lsof` würden Sie dazu wie folgt vorgehen:

```
# pkg_delete lsof-4.57
```

5.5.4. Ports aktualisieren

Als erstes sollten sie sich alle installierten Ports anzeigen lassen, von denen eine aktuellere Version in der Ports-Sammlung existiert. Dazu verwenden Sie den Befehl pkg_version(1):

```
# pkg_version -v
```

5.5.4.1. `/usr/ports/UPDATING`

Nachdem Sie die Ports-Sammlung auf den neusten Stand gebracht haben, lesen Sie bitte zuerst die Datei `/usr/ports/UPDATING`, bevor Sie einen Port aktualisieren. In dieser Datei werden Probleme und zusätzlich durchzuführende Schritte bei der Aktualisierung einzelner Ports beschrieben. Dazu gehören solche Dinge wie geänderte Dateiformate, verschobene Konfigurationsdateien, aber auch Inkompatibilitäten zu einer Vorgängerversion.

Sollte `UPDATING` etwas hier Gesagtem widersprechen, so gilt das in `UPDATING` Gesagte.

5.5.4.2. Ports mit Portupgrade aktualisieren

portupgrade wurde entwickelt, um die Aktualisierung von Ports zu vereinfachen. Sie können **portupgrade** über den Port `ports-mgmt/portupgrade` wie jeden anderen Port mit `make install clean` installieren:

```
# cd /usr/ports/ports-mgmt/portupgrade/
# make install clean
```

Durchsuchen Sie regelmäßig (am besten vor jeder Aktualisierung) die Liste der installierten Ports mit `pkgdb -F` und beheben Sie alle gefundenen Probleme.

Wenn Sie `portupgrade -a` eingeben, beginnt **portupgrade** automatisch mit der Aktualisierung aller veralteter Ports Ihres Systems. Verwenden Sie den Schalter `-i`, wenn Sie individuell entscheiden wollen, ob ein Port aktualisiert werden soll:

```
# portupgrade -ai
```

Wenn Sie nur eine einzelne Anwendung anstelle aller Anwendungen aktualisieren wollen, verwenden Sie das Kommando `portupgrade` *`pkgname`*. Geben Sie den Schalter `-R` an, wenn **portupgrade** zuvor alle Ports aktualisieren soll, die von dem gegebenen Paket abhängen.

Der Schalter `-P` verwendet zur Installation Pakete anstelle von Ports. Mit dieser Option durchsucht **portupgrade** die in der Umgebungsvariablen `PKG_PATH` aufgeführten Verzeichnisse nach Paketen. Sind lokal keine Pakete vorhanden, versucht **portupgrade** die Pakete über das Netz herunterzuladen. Gibt es die Pakete weder lokal noch auf entfernten Rechnern, werden die Ports verwendet. Um dies zu verhindern, benutzen Sie die Option `-PP`.

```
# portupgrade -PP gnome2
```

Wenn Sie nur die Quelldateien des Ports (oder die Pakete mit `-P`) herunterladen möchten, ohne die Anwendung zu bauen oder zu installieren, geben Sie die Option `-F` an. Weitere Möglichkeiten lesen Sie bitte in der Hilfeseite portupgrade(1) nach.

5.5.4.3. Ports mit Portmanager aktualisieren

Portmanager ist ein weiteres Werkzeug, das die Aktualisierung installierter Ports erleichtert. Es kann über den Port `ports-mgmt/portmanager` installiert werden:

```
# cd /usr/ports/ports-mgmt/portmanager
# make install clean
```

Alle installierten Ports können danach durch folgende Eingabe aktualisiert werden:

```
# portmanager -u
```

Wenn Sie zusätzlich die Optionen `-ui` an **Portmanager** übergeben, werden Sie bei jedem Schritt um eine Bestätigung gefragt. **Portmanager** ist außerdem in der Lage, neue Ports auf Ihrem System zu installieren. Im Gegensatz zum bekannten `make install clean` aktualisiert es aber vor dem Bau und der Installation eines Ports alle abhängigen Ports.

```
# portmanager x11/gnome2
```

Treten bei den Abhängigkeiten des zu installierenden Ports Probleme auf, ist **Portmanager** in der Lage, alle Abhängigkeiten in der korrekten Reihenfolge neu zu bauen. Nachdem dieser Schritt abgeschlossen ist, wird der problematische Port ebenfalls neu gebaut.

```
# portmanager graphics/gimp -f
```

Weitere Informationen finden Sie in der Manualpage portmanager(1).

5.5.4.4. Ports mit Portmaster aktualisieren

Bei **Portmaster** handelt es sich um ein weiteres Werkzeug zum Aktualisieren von Ports. **Portmaster** nutzt nur Werkzeuge, die bereits im Basissystem vorhanden sind (ist also nicht von weiteren Ports abhängig). Es verwendet Informationen in `/var/db/pkg/`, um festzustellen, welche Ports aktualisiert werden sollen. Sie können dieses Program über den Port `ports-mgmt/portmaster` installieren:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

Portmaster teilt Ports in vier Kategorien ein:

- Root ports (no dependencies, not depended on)
- Trunk ports (no dependencies, are depended on)
- Branch ports (have dependencies, are depended on)
- Leaf ports (have dependencies, not depended on)

Um eine Liste aller installierter Ports anzuzeigen (und nach neueren Versionen zu suchen), verwenden Sie die Option `-L`:

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache-2.2.3
        ===>>> New version available: apache-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

Um alle derzeit installierten Ports zu aktualisieren, verwenden Sie einfach den folgenden Befehl:

```
# portmaster -a
```

> **Anmerkung:** In der Voreinstellung erzeugt **Portmaster** eine Sicherheitskopie, bevor ein installierter Port gelöscht wird. Ist die Installation der neuen Version erfolgreich, wird dieses Backup wieder gelöscht. Wollen Sie das Backup lieber manuell löschen, verwenden Sie die Option `-b` beim Aufruf von **Portmaster**. Durch die Verwendung der Option `-i` wird **Portmaster** im interaktiven Modus gestartet und fragt bei jedem zu aktualisierenden Port nach, wie Sie vorgehen wollen.

Treten während der Aktualisierung Fehler auf, können Sie die Option `-f` verwenden, um alle Ports zu aktualisieren beziehungsweise neu zu bauen:

```
# portmaster -af
```

Portmaster ist auch in der Lage, neue Ports zu installieren, wobei zuvor alle abhängigen Ports aktualisiert werden:

```
# portmaster shells/bash
```

Weiterführende Informationen finden Sie in der Manualpage portmaster(8).

5.5.5. Platzbedarf von Ports

Die Ports-Sammlung kann sehr viel Plattenplatz verschlingen. Führen Sie nach dem Bau und der Installation eines Ports `make clean` aus, um die Arbeitsverzeichnisse zu löschen. Dieser Befehl entfernt das Verzeichnis `work` des gebauten Ports. Wollen Sie die gesamte Ports-Sammlung aufräumen, verwenden Sie folgenden Befehl:

```
# portsclean -C
```

Im Laufe der Zeit werden sich zahlreiche Distfiles im Verzeichnis `distfiles` ansammeln. Sie können diese entweder händisch löschen, oder Sie verwenden den folgenden Befehl, um alle Distfiles zu löschen, die nicht länger benötigt werden:

```
# portsclean -D
```

Falls Sie nur alle Distfiles löschen wollen, die von keinem derzeit installierten Port referenziert werden:

```
# portsclean -DD
```

> **Anmerkung:** Das Werkzeug `portsclean` wird automatisch bei der Installation von **portupgrade** mit installiert.

Denken Sie daran, installierte Ports wieder zu entfernen, wenn Sie diese nicht mehr benötigen. Um diese Arbeit zu erleichtern, können Sie den Port `ports-mgmt/pkg_cutleaves` installieren.

5.6. Nach der Installation

Nach der Installation einer neuen Anwendung wollen Sie wahrscheinlich die mitgelieferte Dokumentation lesen und die Konfigurationsdateien der Anwendung anpassen. Wenn die Anwendung ein Dæmon ist, sollten Sie sicherstellen, dass die Anwendung beim Booten startet.

Die einzelnen Schritte sind natürlich von Anwendung zu Anwendung verschieden. Wenn Sie sich allerdings nach der Installation einer Anwendung die Frage “Was nun?” stellen, helfen die folgenden Hinweise vielleicht weiter.

- Finden Sie mit pkg_info(1) heraus, welche Dateien die Anwendung wo installiert hat. Wenn Sie beispielsweise gerade die Version 1.0.0 von FooPackage installiert haben, zeigt Ihnen das folgende Kommando alle installierten Dateien des Pakets:

  ```
  # pkg_info -L foopackage-1.0.0 | less
  ```

  Achten Sie besonders auf die Manualpages, die Sie in `man/` Verzeichnissen finden und auf Konfigurationsdateien, die in `etc/` abgelegt werden. Manche Pakete enthalten in `doc/` zusätzliche Dokumentation.

  Wenn Sie sich nicht sicher sind, welche Version einer Anwendung Sie gerade installiert haben, können Sie mit dem folgenden Kommando nach der Anwendung suchen:

  ```
  # pkg_info | grep -i foopackage
  ```

  Das Kommando zeigt alle installierten Pakete, deren Paketname *`foopackage`* enthält. Ersetzen Sie *`foopackage`* durch den Namen der Anwendung, die Sie suchen.

- Nachdem Sie die Manualpages der Anwendung gefunden haben, lesen Sie diese bitte mit man(1). Schauen Sie sich auch die Beispiele für Konfigurationsdateien und die zusätzliche Dokumentation, wenn es welche gibt, an.
- Wenn es für die Anwendung eine Webseite gibt, suchen Sie dort nach zusätzlicher Dokumentation wie FAQs (häufig gestellte Fragen). Wenn Sie die Adresse der Webseite nicht kennen, versuchen Sie das folgende Kommando:

  ```
  # pkg_info foopackage-1.0.0
  ```

  Die Ausgabe enthält oft eine Zeile, die mit `WWW:` beginnt und die URL der Webseite enthält.
- Ports, die während des Systemstarts gestartet werden sollen, installieren meist ein Beispielskript im Verzeichnis `/usr/local/etc/rc.d`. Überprüfen Sie dieses Skript. Wenn nötig, passen Sie das Skript an und benennen Sie es um. Weitere Informationen finden Sie in Abschnitt 12.5.

5.7. Kaputte Ports

Stolpern Sie einmal über einen Port, der bei Ihnen nicht funktioniert, könnten Sie zum Beispiel Folgendes tun:

1. Stellen Sie fest, ob die Datenbank mit den Problemberichten (http://www.FreeBSD.org/de/support.html#gnats) bereits einen Lösungsvorschlag enthält. Ist dies der Fall, können Sie die vorgeschlagene Lösung testen.
2. Bitten Sie den Betreuer des Ports um Hilfe. Geben Sie dazu `make maintainer` ein oder lesen Sie das `Makefile` im Verzeichnis des Ports, um an die E-Mail-Adresse zu kommen. Vergessen Sie nicht den Namen und die Version des Ports (schicken Sie die Zeile mit `$FreeBSD:` aus dem `Makefile`) und die Ausgabe bis zur Fehlermeldung mitzuschicken.

   Anmerkung: Einige Ports werden nicht von einer Einzelperson, sondern von einer Mailingliste (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/mailing-list-faq/article.html) betreut. Viele (aber

nicht alle) dieser Adressen haben die Form `<freebsd-NameDerListe@FreeBSD.org>`. Denken Sie daran, wenn Sie Ihre Fragen formulieren.

Dies gilt insbesondere für Ports, die als als Betreuer den Eintrag `<ports@FreeBSD.org>` aufweisen. Derartige Ports haben überhaupt keinen Betreuer. Korrekturen und Unterstützung kommen daher nur von Personen, die diese Mailingliste abonniert haben. Gerade in diesem Bereich werden jederzeit zusätzliche freiwillige Helfer benötigt!

Erhalten Sie auf Ihre Anfrage keine Antwort, können Sie über send-pr(1) einen Problembericht erstellen. Bevor Sie einen solchen Bericht erstellen, sollten Sie den Artikel Writing FreeBSD Problem Reports (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/problem-reports/article.html) lesen.

3. Reparieren Sie ihn! Das FreeBSD Porter's Handbook (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/porters-handbook/index.html) enthält eine detaillierte Beschreibung des Portsystems. Damit sind Sie in der Lage, einen gelegentlich kaputten Port zu reparieren oder einen eigenen Port zu erstellen.

4. Holen Sie sich das Paket von einem FTP-Server in Ihrer Nähe. Die "Basis"-Sammlung aller Pakete liegt auf `ftp.de.FreeBSD.org` im Verzeichnis packages (ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ports/packages/). Aber versuchen Sie *zuerst* einen Spiegel in Ihrer Nähe! Benutzen Sie das Programm pkg_add(1), um Pakete auf Ihrem Rechner zu installieren. Dies hat zudem den Vorteil, dass es schneller geht.

Kapitel 6. Das X-Window-System

Erweitert um X.Orgs X11-Server von Ken Tom und Marc Fonvieille. Übersetzt von Martin Heinen.

6.1. Übersicht

Mit X11 steht unter FreeBSD eine leistungsfähige frei verfügbare grafische Benutzeroberfläche zur Verfügung, die in **Xorg** (sowie in weiteren, hier nicht diskutierten Varianten) implementiert wurde. **Xorg** von der X.Org Foundation ist der voreingestellte Standard-X11-Server, der unter einer Lizenz ähnlich der von FreeBSD steht. Zusätzlich sind einige kommerzielle X-Server für FreeBSD verfügbar.

Auskunft über von X11 unterstützte Video-Hardware gibt die Webseite Xorg (http://www.x.org/).

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- die Komponenten des X-Window-Systems und ihr Zusammenspiel kennen.
- Wissen, wie X11 installiert und konfiguriert wird.
- Wissen, wie Sie verschiedene Window-Manager installieren und benutzen.
- Wissen, wie TrueType®-Schriftarten mit X11 benutzt werden.
- Wissen, wie Sie die grafische Anmeldung (**XDM**) einrichten.

Bevor Sie dieses Kapitel lesen, sollten Sie

- wissen, wie Sie Software Dritter installieren (Kapitel 5).

6.2. X-Grundlagen

Anwendern anderer grafischer Benutzeroberflächen, wie Microsoft Windows oder Mac OS, kommt X beim ersten Mal oft befremdlich vor.

Man braucht kein weitreichendes Verständnis der X-Komponenten und Ihres Zusammenspiels, um X anzuwenden. Um die Stärken von X auszunutzen, sollten Sie allerdings die Grundlagen verstehen.

6.2.1. Warum heißt es X?

X ist nicht die erste grafische Benutzeroberfläche, die für UNIX geschrieben wurde. Die Entwickler von X arbeiteten vorher an einem anderen System, das W (von engl. *window*: Fenster) hieß. X ist schlicht der nächste Buchstabe im Alphabet.

X wird “X”, “X-Window-System” oder “X11” genannt. Sagen Sie bitte nicht “X-Windows”: das kommt bei einigen Leuten schlecht an (die Hilfeseite X(7) führt dies näher aus).

6.2.2. Das Client/Server-Modell von X

X wurde von Anfang an netzwerktransparent entworfen und verwendet ein Client-Server-Modell. In diesem Modell läuft der Server auf dem Rechner, an dem die Tastatur, der Bildschirm und die Maus angeschlossen ist. Der Server ist

für Dinge wie die Verwaltung des Bildschirms und die Verarbeitung von Tastatur- und Maus-Eingaben sowie anderer Ein- und Ausgabegeräte (beispielsweise könnte ein "Tablet" zur Eingabe oder ein Videoprojektor zur Ausgabe verwendet werden) verantwortlich. Jede X-Anwendung, beispielsweise ein **XTerm** oder **Netscape** ist ein Client. Der Client sendet dem Server Nachrichten wie "Zeichne an diesen Koordinaten ein Fenster" und der Server sendet dem Client Nachrichten der Art "Der Benutzer hat gerade den Ok-Knopf gedrückt".

In kleinen Umgebungen laufen der X-Server und die X-Clients auf demselben Rechner. Es ist aber durchaus möglich, den X-Server auf einem weniger leistungsfähigen Arbeitsplatzrechner laufen zu lassen und die X-Anwendungen (die Clients) auf dem leistungsfähigen und teuren Server der Arbeitsgruppe zu betreiben. In diesem Fall kommunizieren der X-Server und die X-Clients über das Netz.

Dieses Modell verwirrt viele Leute, die erwarten, dass der X-Server der dicke Rechner im Maschinenraum und der X-Client ihr Arbeitsplatzrechner ist.

Merken Sie sich einfach, dass der X-Server der Rechner mit dem Bildschirm und der Maus ist und die X-Clients Programme sind, die in den Fenstern laufen.

Das X-Protokoll ist unabhängig vom verwendeten Betriebssystem und Rechnertyp. Ein X-Server kann durchaus auch unter Microsoft Windows oder Apples Mac OS betrieben werden, wie viele kostenlose und kommerzielle Anwendungen zeigen.

6.2.3. Der Window-Manager

Die X-Philosophie "Werkzeuge statt Richtlinien" ist wie die UNIX-Philosophie. Es wird nicht vorgeschrieben, wie eine Aufgabe zu lösen ist, stattdessen erhält der Benutzer Werkzeuge, über die er frei verfügen kann.

Dies geht so weit, dass X nicht bestimmt, wie Fenster auf dem Bildschirm auszusehen haben, wie sie mit der Maus zu verschieben sind, welche Tastenkombination benutzt werden muss, um zwischen den Fenstern zu wechseln (z.B. **Alt**+**Tab** unter Microsoft Windows), oder ob die Fensterrahmen Schaltflächen zum Schließen haben.

X gibt die Verantwortung für all diese Sachen an eine Anwendung ab, die *Window-Manager* genannt wird. Unter X gibt es zahlreiche Window-Manager: **AfterStep**, **Blackbox**, **ctwm**, **Enlightenment**, **fvwm**, **Sawfish**, **twm**, **Window Maker** um nur einige zu nennen. Jeder dieser Window-Manager sieht anders aus: Manche stellen virtuelle Bildschirme zur Verfügung, in anderen lassen sich die Tastenkombinationen zur Verwaltung des Bildschirms anpassen, einige besitzen eine Startleiste oder etwas Ähnliches und in manchen lässt sich das Aussehen und Verhalten über die Anwendung von *Themes* beliebig einstellen. Die eben genannten Window-Manager und viele weitere finden Sie in der Kategorie `x11-wm` der Ports-Sammlung.

Die grafischen Benutzeroberflächen **KDE** und **GNOME** besitzen eigene Window-Manager, die in den grafischen Arbeitsplatz integriert sind.

Die Window-Manager werden unterschiedlich konfiguriert. Einige erwarten eine manuell erstellte Konfigurationsdatei, andere bieten grafische Werkzeuge für die meisten Konfigurationsarbeiten an. Die Konfigurationsdatei von **Sawfish** ist sogar in einem Lisp-Dialekt geschrieben.

> **Fokus:** Der Window-Manager ist für die Methode, mit der ein Fenster den Fokus bekommt, verantwortlich. Jedes System, das Fenster verwendet, muss entscheiden, wie ein Fenster aktiviert wird, damit es Eingaben empfangen kann. Das aktive Fenster sollte zudem sichtbar gekennzeichnet werden.
>
> Eine geläufige Methode, den Fokus zu wechseln, wird "click-to-focus" genannt. Die Methode wird in Microsoft Windows benutzt: Ein Fenster wird aktiv, wenn es mit der Maus angeklickt wird.

X legt nicht fest, wie der Fokus einzustellen ist, stattdessen bestimmt der Window-Manager welches Fenster den Fokus zu einem gegebenen Zeitpunkt erhält. Alle Window-Manager stellen die Methode “click-to-focus” bereit, die meisten stellen auch noch andere Methoden bereit.

Verbreitete Methoden, den Fokus einzustellen, sind:

focus-follows-mouse

: Den Fokus hat das Fenster, unter dem sich der Mauszeiger befindet. Das muss nicht unbedingt das Fenster, sein, das sich vorne befindet. Wird der Mauszeiger in ein anderes Fenster bewegt, so erhält dieses Fenster den Fokus, ohne das es angeklickt werden muss.

sloppy-focus

: Diese Methode erweitert die Methode “focus-follows-mouse”. Wenn die Maus mit “focus-follows-mouse” aus dem Fenster auf die Oberfläche bewegt wird, verliert das aktive Fenster den Fokus. Da dann kein Fenster mehr den Fokus hat, gehen alle Eingaben verloren. Die Methode “sloppy-focus” wechselt den Fokus nur, wenn sich der Mauszeiger in ein neues Fenster bewegt und nicht, wenn er das aktive Fenster verlässt.

click-to-focus

: Das aktive Fenster wird durch einen Mausklick festgelegt (dabei kann das Fenster vor alle anderen Fenster gesetzt werden). Alle Eingaben werden dann, unabhängig von der Position des Mauszeigers, dem aktiven Fenster zugeordnet.

Viele Window-Manager unterstützen noch andere Methoden, so wie Abwandlungen der hier vorgestellten Methoden. Schauen Sie sich dazu bitte die Hilfeseiten Ihres Window-Managers an.

6.2.4. Widgets

Die X-Philosophie dehnt sich auch auf die Widgets aus, die von den Anwendungen benutzt werden.

Ein *Widget* bezeichnet Objekte, die manipuliert werden können, wie *buttons* (Schaltflächen), *check buttons* (Mehrfachauswahlknopf), *radio buttons* (Einfachauswahlknopf), Icons und Auswahllisten. Unter Microsoft Windows werden Widgets *Controls* genannt.

Microsoft Windows und Apples Mac OS geben strenge Richtlinien für Widgets vor: Von den Entwicklern wird erwartet, dass Sie Anwendungen mit einheitlichem Aussehen und einheitlicher Bedienung (*look and feel*) entwickeln. X gibt weder einen Stil noch Widgets vor, die benutzt werden müssen.

Erwarten Sie daher nicht, dass alle X-Anwendungen gleich aussehen oder sich gleich bedienen lassen. Es gibt mehrere verbreitete Widget-Sammlungen, beispielsweise die Athena-Widgets vom MIT, **Motif®** (abgeschrägte Ecken und drei Grautöne, danach wurden die Widgets von Microsoft Windows entworfen) oder **OpenLook**.

Die meisten neuen X-Anwendungen benutzen heute modern aussehende Widgets, wie Qt, das von **KDE** benutzt wird oder GTK+, das von **GNOME** benutzt wird. Damit wird eine gewisse Einheitlichkeit in Bedienung und Aussehen erreicht, die sicher neuen Benutzern die Arbeit erleichtert.

6.3. X11 installieren

Xorg ist der Standard-X-Server unter FreeBSD. **Xorg** ist der von der X.Org Foundation herausgegebene X-Server des Open-Source X Window Systems. **Xorg** beruht auf **XFree86 4.4RC2** und X11R6.6. Derzeit ist die Version 7.7 von **Xorg** in der Ports-Sammlung vorhanden.

Die nachstehenden Kommandos bauen und installieren **Xorg** aus der Ports-Sammlung:

```
# cd /usr/ports/x11/xorg
# make install clean
```

> **Anmerkung:** Der komplette Bau von **Xorg** benötigt mindestens 4 GB freien Plattenplatz.

Mit pkg_add(1) können Sie X11 direkt von fertigen Paketen installieren. Wenn pkg_add(1) die Pakete herunterlädt, lassen Sie die Versionsnummer aus. pkg_add(1) holt automatisch die aktuelle Version eines Pakets.

Das **Xorg**-Paket holen und installieren Sie wie folgt:

```
# pkg_add -r xorg
```

> **Anmerkung:** Die obigen Beispiele installieren die vollständige X11-Distribution, die unter anderem Server, Clients und Fonts enthält. Für die einzelnen Teile der Distribution gibt es ebenfalls separate Pakete.
>
> Alternativ können Sie `x11/xorg-minimal` verwenden, um eine minimale X11-Distribution zu installieren.

Der Rest dieses Kapitels erklärt, wie Sie X11 konfigurieren und sich eine Arbeitsumgebung einrichten.

6.4. X11 konfigurieren

Beigetragen von Christopher Shumway.

6.4.1. Vorarbeiten

Bevor Sie X11 konfigurieren, benötigen Sie folgende Informationen:

- die Spezifikationen des Monitors
- den Chipset des Grafikadapters
- die Speichergröße des Grafikadapters

Aus den Spezifikationen des Monitors ermittelt X11 die Auflösung und die Wiederholrate für den Betrieb des X-Servers. Die Spezifikationen entnehmen Sie der Dokumentation des Monitors oder der Webseite des Herstellers. Sie benötigen die horizontale und die vertikale Synchronisationsfrequenz.

Der Chipsatz der Grafikkarte bestimmt den Treiber, den X11 verwendet. Die meisten Chipsätze werden automatisch erkannt, Sie brauchen die Information jedoch, wenn die Erkennung fehlschlägt.

Die Speichergröße der Grafikkarte bestimmt die maximal mögliche Auflösung und Farbtiefe.

6.4.2. X11 konfigurieren

Xorg verwendet HAL, um Tastaturen und Mäuse automatisch zu erkennen. Die Ports `sysutils/hal` und `devel/dbus` werden als Abhängigkeiten von `x11/xorg` installiert, müssen aber durch die folgenden Einträge in `/etc/rc.conf` aktiviert werden:

```
hald_enable="YES"
dbus_enable="YES"
```

Diese Dienste sollten (entweder manuell oder durch einen Neustart) gestartet werden, bevor mit der weiteren Konfiguration oder Verwendung von **Xorg** begonnen wird.

Xorg kann oft schon ohne weitere Konfigurationsschritte laufen, indem am Prompt folgendes eingegeben wird:

```
% startx
```

Die automatische Konfiguration kann mit bestimmter Hardware fehlschlagen oder gewisse Dinge nicht so einrichten, wie gewünscht. In diesen Fällen ist eine manuelle Konfiguration notwendig.

> **Anmerkung:** Grafische Oberflächen wie **GNOME**, **KDE** oder **Xfce** besitzen eigene Werkzeuge, die es dem Benutzer erlauben, auf einfache Art und Weise die Bildschirmparameter wie die Auflösung zu ändern. Falls die Standardkonfiguration für Sie nicht akzeptabel ist und die Installation einer grafischen Oberfläche geplant ist, fahren Sie damit fort und benutzen Sie dann das entsprechende Werkzeug für die Bildschirmeinstellungen.

Die X11 Konfiguration spielt sich in mehreren Schritten ab. Dazu erstellen Sie als erstes eine Vorgabe für die Konfigurationsdatei. Setzen Sie dazu als `root` den folgenden Befehl ab:

```
# Xorg -configure
```

Die Vorgabe-Konfiguration wird dann unter dem Namen `xorg.conf.new` im Verzeichnis `/root` gespeichert (das verwendete Verzeichnis wird durch die Umgebungsvariable `$HOME` bestimmt und hängt davon ab, wie Sie zu `root` gewechselt sind). X11 hat in diesem Schritt versucht, die Grafik-Hardware des Systems zu erkennen und eine Konfigurationsdatei ausgeschrieben, die zur Hardware passende Treiber lädt.

Im nächsten Schritt wird geprüft, ob **Xorg** die Grafik-Hardware des Systems verwenden kann. Setzen Sie dazu den folgenden Befehl ab:

```
# Xorg -config xorg.conf.new -retro
```

Wenn jetzt ein graues Raster und der X-Mauszeiger erscheinen, war die Konfiguration erfolgreich. Beenden Sie den Test, indem Sie auf die virtuelle Konsole wechseln, die Sie verwendet haben, um den Test zu starten, durch gleichzeitiges drücken von **Ctrl**+**Alt**+**F*n*** (**F1** für die erste virtuelle Konsole) und drücken anschliessend **Ctrl**+**C**.

> **Anmerkung:** Die Tastenkombination **Ctrl**+**Alt**+**Backspace** kann verwendet werden, um **Xorg** zu beenden. Um diese zu aktivieren, fügen geben Sie entweder den folgenden Befehl von einem X-Terminalemulator ein:
>
> ```
> % setxkbmap -option terminate:ctrl_alt_bksp
> ```
>
> oder erstellen Sie eine Tastaturkonfigurationsdatei für **hald**, `x11-input.fdi` genannt, und legen Sie diese im Verzeichnis `/usr/local/etc/hal/fdi/policy` ab. Diese Datei sollte die folgenden Zeilen enthalten:
>
> ```
> <?xml version="1.0" encoding="iso-8859-1"?>
> ```

```
<deviceinfo version="0.2">
   <device>
     <match key="info.capabilities" contains="input.keyboard">
           <merge key="input.x11_XkbOptions" type="string">terminate:ctrl_alt_bksp</merge>
     </match>
   </device>
</deviceinfo>
```

Sie müssen anschliessend ihren Computer neu starten, um **hald** zu zwingen, diese Datei einzulesen.

Die folgende Zeile muss ebenfalls zu `xorg.conf.new` hinzugefügt werden, entweder in den Abschnitt `ServerLayout` oder `ServerFlags`:

```
Option  "DontZap"       "off"
```

Wenn die Maus nicht funktioniert, prüfen Sie, ob die Maus konfiguriert wurde. Die Mauskonfiguration wird in Abschnitt 2.10.10 im FreeBSD-Installationskapitel beschrieben. In neueren **Xorg**-Versionen werden die `InputDevice`-Abschnitte in `xorg.conf` ignoriert, um stattdessen die automatisch erkannten Geräte zu verwenden. Um das alte Verhalten wiederherzustellen, fügen Sie die folgende Zeile zum `ServerLayout`- oder dem `ServerFlags`-Abschnitt dieser Datei hinzu:

```
Option "AutoAddDevices" "false"
```

Eingabegeräte können dann wie in den vorherigen Versionen konfiguriert werden, zusammen mit anderen benötigen Optionen (z.B. wechseln des Tastaturlayouts).

Anmerkung: Wie zuvor erwähnt, wird standardmässig der **hald**-Dienst automatisch Ihre Tastatur erkennen. Es kann passieren, dass ihr Tastaturlayout oder das Modell nicht korrekt erkannt wird. Grafische Oberflächen wie **GNOME**, **KDE** oder **Xfce** stellen Werkzeuge für die Konfiguration der Tastatur bereit. Es ist allerdings auch möglich, die Tastatureigenschaften direkt zu setzen, entweder mit Hilfe von setxkbmap(1) oder mit einer Konfigurationsregel von **hald**.

Wenn Sie zum Beispiel eine PC 102-Tasten Tastatur mit französischem Layout verwenden möchten, mössen Sie eine Tastaturkonfigurationsdatei für **hald**, genannt `x11-input.fdi`, im Verzeichnis `/usr/local/etc/hal/fdi/policy` ablegen. Diese Datei sollte die folgenden Zeilen enthalten:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<deviceinfo version="0.2">
   <device>
     <match key="info.capabilities" contains="input.keyboard">
           <merge key="input.x11_options.XkbModel" type="string">pc102</merge>
           <merge key="input.x11_options.XkbLayout" type="string">fr</merge>
     </match>
   </device>
</deviceinfo>
```

Wenn diese Datei bereits existiert, kopieren Sie nur die Zeilen in diese Datei, welche die Tastaturkonfiguration betreffen.

Sie mössen Ihren Computer neu starten, um **hald** zu zwingen, diese Datei einzulesen.

Es ist möglich, die gleiche Konfiguration von einem X-Terminal oder einem Skript über den folgenden Befehl heraus zu tätigen:

```
% setxkbmap -model pc102 -layout fr
```

Die Datei `/usr/local/share/X11/xkb/rules/base.lst` listet die verschiedenen Tastatur- und Layoutoptionen auf, die Ihnen zur Verfügung stehen.

Als Nächstes passen Sie `xorg.conf.new` an. Öffnen Sie die Datei in einem Editor, wie emacs(1) oder ee(1) und fügen Sie die Synchronisationsfrequenzen des Monitors ein. Die Frequenzen werden im Abschnitt `"Monitor"` eingetragen:

```
Section "Monitor"
        Identifier   "Monitor0"
        VendorName   "Monitor Vendor"
        ModelName    "Monitor Model"
        HorizSync    30-107
        VertRefresh  48-120
EndSection
```

Unter Umständen fehlen die Schlüsselwörter `HorizSync` und `VertRefresh`, die Sie dann nachtragen müssen. Geben Sie, wie im Beispiel gezeigt, die horizontale Synchronisationsfrequenz hinter `HorizSync` und die vertikale Synchronisationsfrequenz hinter `VertRefresh` an.

X unterstützt die Energiesparfunktionen (DPMS, Energy Star) Ihres Monitors. Mit xset(1) können Sie Zeitschranken für die DPMS-Modi “standby”, “suspend”, “off” vorgeben, oder diese zwingend aktivieren. Die DPMS-Funktionen können Sie mit der nachstehenden Zeile im `"Monitor"`-Abschnitt aktivieren:

```
Option       "DPMS"
```

Die gewünschte Auflösung und Farbtiefe stellen Sie im Abschnitt `"Screen"` ein:

```
Section "Screen"
        Identifier "Screen0"
        Device     "Card0"
        Monitor    "Monitor0"
        DefaultDepth 24
        SubSection "Display"
                Viewport  0 0
                Depth     24
                Modes     "1024x768"
        EndSubSection
EndSection
```

Mit `DefaultDepth` wird die Farbtiefe des X-Servers vorgegeben. Mit der Option `-depth` von Xorg(1) lässt sich die vorgegebene Farbtiefe überschreiben. `Modes` gibt die Auflösung für die angegebene Farbtiefe an. Die Farbtiefe im Beispiel beträgt 24 Bits pro Pixel, die zugehörige Auflösung ist 1024x768 Pixel. Beachten Sie, dass in der Voreinstellung nur Standard-VESA-Modi der Grafikkarte angegeben werden können.

Sichern Sie die Konfigurationsdatei und testen Sie die Konfiguration wie oben beschrieben.

Anmerkung: Bei der Fehlersuche sind Ihnen die Protokolle des X11-Servers behilflich. In den Protokollen wird die gefundene Graphik-Hardware protokolliert. Die Protokolle von **Xorg** heißen `/var/log/Xorg.0.log`. Die Dateinamen enthalten eine laufende Nummer, der Name variiert daher von `Xorg.0.log` zu `Xorg.8.log`.

Wenn alles funktioniert hat, installieren Sie die Datei an einen Ort, an dem Xorg(1) sie findet. Normalerweise wird die Konfigurationsdatei unter `/etc/X11/xorg.conf` oder `/usr/local/etc/X11/xorg.conf` gespeichert:

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Damit ist die X11-Konfiguration beendet und **Xorg** kann nun mithilfe von startx(1) gestartet werden. Alternativ können Sie X11 auch mit xdm(1) starten.

6.4.3. Spezielle Konfigurationen

6.4.3.1. Konfiguration des Intel® i810 Graphics Chipsets

Der Intel i810-Chipset benötigt den Treiber `agpgart`, die AGP-Schnittstelle von X11. Weitere Informationen finden sich in agp(4).

Ab jetzt kann die Hardware wie jede andere Grafikkarte auch konfiguriert werden. Der Treiber agp(4) kann nicht nachträglich mit kldload(8) in einen laufenden Kernel geladen werden. Er muss entweder fest im Kernel eingebunden sein oder beim Systemstart über `/boot/loader.conf` geladen werden.

6.4.3.2. Einen Widescreen-Monitor einsetzen

Dieser Abschnitt geht über die normalen Konfigurationsarbeiten hinaus und setzt einiges an Vorwissen voraus. Selbst wenn die Standardwerkzeuge zur X-Konfiguration bei diesen Geräten nicht zum Erfolg führen, sollten sich in den Logdateien genug Informationen finden, mit denen Sie letztlich doch einen funktionierenden X-Server konfigurieren können. Alles, was Sie dazu noch benötigen, ist ein Texteditor.

Aktuelle Widescreen-Formate (wie WSXGA, WSXGA+, WUXGA, WXGA, WXGA+, und andere mehr) unterstützen Seitenverhältnisse wie 16:10 oder 10:9, die unter X Probleme verursachen können. Bei einem Seitenverhältnis von 16:10 sind beispielsweise folgende Auflösungen möglich:

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

Diese Konfiguration könnte so einfach sein wie das zusätzliche Anlegen eines Eintrags einer dieser Auflösungen als ein möglicher `Mode` in `Section "Screen"`:

```
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
       Viewport  0 0
       Depth     24
       Modes     "1680x1050"
EndSubSection
```

```
EndSection
```

Xorg ist normalerweise intelligent genug, um die Informationen zu den erlaubten Auflösungen über I2C/DDC zu beziehen, und weiß daher, welche Auflösungen und Frequenzen Ihr Widescreen-Monitor unterstützt.

Wenn diese `ModeLines` in den Treiberdateien nicht vorhanden sind, kann es sein, dass Sie **Xorg** beim Finden der korrekten Werte unterstützen müssen. Dazu extrahieren Sie die benötigten Informationen aus der Datei `/var/log/Xorg.0.log` und erzeugen daraus eine funktionierende `ModeLine`. Dazu suchen Sie in dieser Datei nach Zeilen ähnlich den folgenden:

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border: 0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border: 0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max 170 MHz
```

Diese Informationen werden auch als EDID-Informationen bezeichnet. Um daraus eine funktionierende `ModeLine` zu erzeugen, müssen Sie lediglich die Zahlen in die korrekte Reihenfolge bringen:

```
ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

Die korrekte `ModeLine` in `Section "Monitor"` würde für dieses Beispiel folgendermaßen aussehen:

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

Nachdem diese Äderungen durchgeführt sind, sollte X auch auf Ihrem neuen Widescreen-Monitor starten.

6.5. Schriftarten in X11 benutzen

Beigetragen von Murray Stokely.

6.5.1. Type 1 Schriftarten

Die Schriftarten, die mit X11 geliefert werden, eignen sich ganz und gar nicht für Desktop-Publishing-Anwendungen. Große Schriftarten zeigen bei Präsentationen deutliche Treppenstufen und die kleinen Schriftarten in **Netscape** sind fast unleserlich. Es gibt allerdings mehrere hochwertige Type 1 Schriftarten (PostScript®), die mit X11 benutzt werden können. Beispielsweise enthalten die URW-Schriftarten (`x11-fonts/urwfonts`) hochwertige Versionen gängiger Type 1 Schriftarten (zum Beispiel Times Roman®, Helvetica®, Palatino®). Die Sammlung Freefonts (`x11-fonts/freefonts`) enthält noch mehr Schriftarten, doch sind diese für den Einsatz in Grafik-Programmen wie **The Gimp** gedacht. Es fehlen auch einige Schriftarten, sodass sich die Sammlung nicht für den alltäglichen Gebrauch eignet. Weiterhin kann X11 leicht so konfiguriert werden, dass es TrueType-Schriftarten verwendet. Mehr dazu erfahren Sie in der Hilfeseite X(7) und im Abschnitt TrueType Schriftarten.

Die Type 1 Schriftarten lassen sich aus der Ports-Sammlung wie folgt installieren:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

Analog lassen sich Freefont und andere Sammlungen installieren. Die neuen Schriftarten müssen Sie in die Konfigurationsdatei des X-Servers im Verzeichnis `/etc/X11` eintragen. Die Konfigurationsdatei von **Xorg** heißt `xorg.conf`. Fügen Sie die folgende Zeile hinzu:

```
FontPath "/usr/local/lib/X11/fonts/URW/"
```

Sie können aber auch in der X-Sitzung das folgende Kommando absetzen:

```
% xset fp+ /usr/local/lib/X11/fonts/URW
% xset fp rehash
```

Dann kennt der X-Server die neuen Schriftarten nur bis zum Ende der Sitzung. Wenn die Änderung dauerhaft sein soll, müssen Sie die Kommandos in `~/.xinitrc` eintragen, wenn Sie X mit `startx` starten, oder in `~/.xsession`, wenn Sie **XDM** benutzen. Sie können die Schriftarten auch in die neue Datei `/usr/local/etc/fonts/local.conf`, die im Abschnitt Anti-aliasing beschrieben wird, eintragen.

6.5.2. TrueType®-Schriftarten

Xorg kann TrueType-Schriftarten mithilfe von zwei Modulen darstellen. Im folgenden Beispiel wird das Freetype-Modul benutzt, da es besser mit anderen Werkzeugen, die TrueType-Schriftarten darstellen, übereinstimmt. Das Freetype-Modul aktivieren Sie im Abschnitt `"Module"` von `/etc/X11/xorg.conf` durch Einfügen der Zeile:

```
Load  "freetype"
```

Erstellen Sie ein Verzeichnis für die TrueType-Schriftarten (z.B. `/usr/local/lib/X11/fonts/TrueType`) und kopieren Sie alle Schriftarten dorthin. Die Schriftarten müssen im UNIX/MS-DOS/Windows-Format vorliegen, Schriftarten von einem Macintosh können Sie nicht direkt übernehmen. Die Schriftarten müssen noch im Katalog `fonts.dir` erfasst werden. Den Katalog erzeugen Sie mit dem Kommando `ttmkfdir` aus dem Port `x11-fonts/ttmkfdir`:

```
# cd /usr/local/lib/X11/fonts/TrueType
# ttmkfdir -o fonts.dir
```

Geben Sie dem System das TrueType-Verzeichnis, wie im Abschnitt Type 1 Schriftarten beschrieben, bekannt:

```
% xset fp+ /usr/local/lib/X11/fonts/TrueType
% xset fp rehash
```

Oder fügen Sie eine `FontPath`-Zeile in die Datei `xorg.conf` ein.

Das war's. Jetzt sollten **Netscape**, **Gimp**, **StarOffice™** und alle anderen X-Anwendungen die TrueType-Schriftarten benutzen. Extrem kleine Schriftarten (Webseiten, die mit hoher Auflösung betrachtet werden) und sehr große Schriftarten (in **StarOffice**) sollten jetzt viel besser aussehen.

6.5.3. Anti-aliasing

Aktualisiert von Joe Marcus Clarke.

Alle Schriftarten in X11, die in den Verzeichnissen `/usr/local/lib/X11/fonts/` und `~/.fonts/` gefunden werden, werden automatisch für Anti-aliasing an Anwendungen zur Verfügung gestellt, die Xft beherrschen. Die meisten aktuellen Anwendungen beherrschen Xft, dazu gehören auch **KDE**, **GNOME** und **Firefox**.

In der Datei `/usr/local/etc/fonts/local.conf` werden die Schriftarten, die mit dem Anti-aliasing-Verfahren benutzt werden sollen und die Eigenschaften des Verfahrens festgelegt. In diesem Abschnitt wird nur die grundlegende Konfiguration von Xft beschrieben. Weitere Details entnehmen Sie bitte der Hilfeseite fonts-conf(5).

Die Datei `local.conf` ist ein XML-Dokument. Achten Sie beim Editieren der Datei daher auf die richtige Groß- und Kleinschreibung und darauf, dass alle Tags geschlossen sind. Die Datei beginnt mit der üblichen XML-Deklaration gefolgt von einer DOCTYPE-Definition und dem `<fontconfig>`-Tag:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

Wie vorher erwähnt, stehen schon alle Schriftarten in `/usr/local/lib/X11/fonts/` und `~/.fonts/` für Anwendungen, die Xft unterstützen, zur Verfügung. Wenn Sie ein Verzeichnis außerhalb dieser beiden Bäume benutzen wollen, fügen Sie eine Zeile wie die nachstehende zu `/usr/local/etc/fonts/local.conf` hinzu:

```
<dir>/path/to/my/fonts</dir>
```

Wenn Sie neue Schriftarten hinzugefügt haben, müssen Sie den Schriftarten-Cache neu aufbauen:

```
# fc-cache -f
```

Das Anti-aliasing-Verfahren zeichnet Ränder leicht unscharf, dadurch werden kleine Schriften besser lesbar und der Treppenstufen-Effekt bei wird großen Schriften vermieden. Auf normale Schriftgrößen sollte das Verfahren aber nicht angewendet werden, da dies die Augen zu sehr anstrengt. Um kleinere Schriftgrößen als 14 Punkt von dem Verfahren auszunehmen, fügen Sie in `local.conf` die nachstehenden Zeilen ein:

```
<match target="font">
    <test name="size" compare="less">
        <double>14</double>
    </test>
    <edit name="antialias" mode="assign">
        <bool>false</bool>
    </edit>
</match>
<match target="font">
    <test name="pixelsize" compare="less" qual="any">
        <double>14</double>
    </test>
    <edit mode="assign" name="antialias">
        <bool>false</bool>
    </edit>
</match>
```

Das Anti-aliasing-Verfahren kann die Abstände einiger Fixschriften falsch darstellen, dies fällt besonders unter **KDE** auf. Sie können das Problem umgehen, indem Sie die Abstände dieser Schriften auf den Wert `100` festsetzen. Fügen Sie die nachstehenden Zeilen hinzu:

```
<match target="pattern" name="family">
    <test qual="any" name="family">
        <string>fixed</string>
    </test>
    <edit name="family" mode="assign">
        <string>mono</string>
    </edit>
 </match>
 <match target="pattern" name="family">
     <test qual="any" name="family">
         <string>console</string>
     </test>
     <edit name="family" mode="assign">
         <string>mono</string>
     </edit>
 </match>
```

Damit werden die Namen der gebräuchlichen Fixschriften auf `"mono"` abgebildet. Für diese Schriften setzen Sie dann den Abstand fest:

```
 <match target="pattern" name="family">
     <test qual="any" name="family">
         <string>mono</string>
     </test>
     <edit name="spacing" mode="assign">
         <int>100</int>
     </edit>
 </match>
```

Bestimmte Schriftarten, wie Helvetica, können Probleme mit dem Anti-Aliasing-Verfahren verursachen. In der Regel erscheinen diese Schriftarten dann vertikal halbiert. Im schlimmsten Fall stürzen Anwendungen als Folge davon ab. Sie vermeiden dies, indem Sie betroffene Schriftarten in `local.conf` von dem Verfahren ausnehmen:

```
 <match target="pattern" name="family">
     <test qual="any" name="family">
         <string>Helvetica</string>
     </test>
     <edit name="family" mode="assign">
         <string>sans-serif</string>
     </edit>
 </match>
```

Wenn Sie `local.conf` editiert haben, stellen Sie bitte sicher, dass die Datei mit dem Tag `</fontconfig>` endet. Ist das nicht der Fall, werden die Änderungen nicht berücksichtigt.

Benutzer können eigene Einstellungen in der Datei `~/.fonts.conf` vornehmen. Achten Sie auch hier auf die richtige XML-Syntax.

Mit einem LCD können Sie *sub-pixel sampling* anstelle von Anti-aliasing einsetzen. Dieses Verfahren behandelt die horizontal getrennten Rot-, Grün- und Blau-Komponenten eines Pixels gesondert und verbessert damit (teilweise sehr wirksam) die horizontale Auflösung. Die nachstehende Zeile in `local.conf` aktiviert diese Funktion:

```
<match target="font">
    <test qual="all" name="rgba">
        <const>unknown</const>
    </test>
    <edit name="rgba" mode="assign">
        <const>rgb</const>
    </edit>
</match>
```

Anmerkung: Abhängig von der Art Ihres Bildschirms müssen Sie anstelle von `rgb` eines der folgenden verwenden: `bgr`, `vrgb` oder `vbgr`. Experimentieren Sie und vergleichen, was besser aussieht.

6.6. Der X-Display-Manager

Beigetragen von Seth Kingsley.

6.6.1. Einführung

Der *X-Display-Manager* (**XDM**), eine optionale Komponente des X-Window-Systems, verwaltet Sitzungen. Er kann mit vielen Komponenten, wie minimal ausgestatteten X-Terminals, Arbeitsplatz-Rechnern und leistungsfähigen Netzwerkservern, nutzbringend eingesetzt werden. Da das X-Window-System netzwerktransparent ist, gibt es zahlreiche Möglichkeiten, X-Clients und X-Server auf unterschiedlichen Rechnern im Netz laufen zu lassen. **XDM** stellt eine grafische Anmeldemaske zur Verfügung, in der Sie den Rechner, auf dem eine Sitzung laufen soll, auswählen können und in der Sie die nötigen Autorisierungs-Informationen, wie Benutzername und Passwort, eingeben können.

Die Funktion des X-Display-Managers lässt sich mit der von getty(8) (siehe Abschnitt 27.3.2) vergleichen. Er meldet den Benutzer am ausgesuchten System an, startet ein Programm (meist einen Window-Manager) und wartet darauf, dass dieses Programm beendet wird, das heißt der Benutzer die Sitzung beendet hat. Nachdem die Sitzung beendet ist, zeigt **XDM** den grafischen Anmeldebildschirm für den nächsten Benutzer an.

6.6.2. XDM einrichten

Um **XDM** verwenden zu können, installieren Sie den Port `x11/xdm` (dieser wird standardmässig nicht in aktuellen **Xorg**-Versionen mitinstalliert). Der **XDM**-Dæmon befindet sich dann in `/usr/local/bin/xdm` und kann jederzeit von `root` gestartet werden. Er verwaltet dann den X-Bildschirm des lokalen Rechners. **XDM** lässt sich bequem mit einem Eintrag in `/etc/ttys` (siehe Abschnitt 27.3.2.1) bei jedem Start des Rechners aktivieren. In `/etc/ttys` sollte schon der nachstehende Eintrag vorhanden sein:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

In der Voreinstellung ist dieser Eintrag nicht aktiv. Um den Eintrag zu aktivieren, ändern Sie den Wert in Feld 5 von `off` zu `on` und starten Sie init(8) entsprechend der Anleitung in Abschnitt 27.3.2.2 neu. Das erste Feld gibt den Namen des Terminals an, auf dem das Programm läuft. Im Beispiel wird `ttyv8` verwendet, das heißt **XDM** läuft auf dem neunten virtuellen Terminal.

6.6.3. XDM konfigurieren

Das Verhalten und Aussehen von **XDM** steuern Sie mit Konfigurationsdateien, die im Verzeichnis `/usr/local/lib/X11/xdm` stehen. Üblicherweise finden Sie dort die folgenden Dateien vor:

Datei	Beschreibung
`Xaccess`	Regelsatz, der zur Autorisierung von Clients benutzt wird.
`Xresources`	Vorgabewerte für X-Ressourcen.
`Xservers`	Liste mit lokalen und entfernten Bildschirmen, die verwaltet werden.
`Xsession`	Vorgabe für das Startskript der Sitzung.
`Xsetup_*`	Skript, das dazu dient, Anwendungen vor der Anmeldung zu starten.
`xdm-config`	Konfiguration für alle auf der Maschine verwalteten Bildschirme.
`xdm-errors`	Fehlermeldungen des Servers.
`xdm-pid`	Die Prozess-ID des gerade laufenden **XDM**-Prozesses.

Im Verzeichnis `/usr/local/lib/X11/xdm` befinden sich auch noch Skripten und Programme, die zum Einrichten der **XDM**-Oberfläche dienen. Der Zweck dieser Dateien und der Umgang mit ihnen wird in der Hilfeseite xdm(1) erklärt. Wir gehen im Folgenden nur kurz auf ein paar der Dateien ein.

Die vorgegebene Einstellung zeigt ein rechteckiges Anmeldefenster, in dem der Rechnername in großer Schrift steht. Darunter befinden sich die Eingabeaufforderungen `Login:` und `Password:`. Mit dieser Maske können Sie anfangen, wenn Sie das Erscheinungsbild von **XDM** verändern wollen.

6.6.3.1. Xaccess

Verbindungen zu **XDM** werden über das “X Display Manager Connection Protocol” (XDMCP) hergestellt. XDMCP-Verbindungen von entfernten Maschinen werden über den Regelsatz in `Xaccess` kontrolliert. Diese Datei wird allerdings ignoriert, wenn in `xdm-config` keine Verbindungen entfernter Maschinen erlaubt sind (dies ist auch die Voreinstellung).

6.6.3.2. Xresources

In dieser Datei kann das Erscheinungsbild der Bildschirmauswahl und der Anmeldemasken festgelegt werden. Das Format entspricht den Dateien im Verzeichnis `app-defaults`, die in der X11-Dokumentation beschrieben sind.

6.6.3.3. Xservers

Diese Datei enthält eine Liste entfernter Maschinen, die in der Bildschirmauswahl angeboten werden.

6.6.3.4. Xsession

Dieses Skript wird vom **XDM** aufgerufen, nachdem sich ein Benutzer erfolgreich angemeldet hat. Üblicherweise besitzt jeder Benutzer eine angepasste Version dieses Skripts in `~/.xsession`, das dann anstelle von `Xsession` ausgeführt wird.

6.6.3.5. Xsetup_*

Diese Skripten werden automatisch ausgeführt bevor die Bildschirmauswahl oder die Anmeldemasken angezeigt werden. Für jeden lokalen Bildschirm gibt es ein Skript, dessen Namen aus `Xsetup_` gefolgt von der Bildschirmnummer gebildet wird (zum Beispiel `Xsetup_0`). Normalerweise werden damit ein oder zwei Programme, wie `xconsole`, im Hintergrund gestartet.

6.6.3.6. xdm-config

Diese Datei enthält Einstellungen, die für jeden verwalteten Bildschirm zutreffen. Das Format entspricht dem der Dateien aus `app-defaults`.

6.6.3.7. xdm-errors

Die Ausgaben jedes X-Servers, den **XDM** versucht zu starten, werden in dieser Datei gesammelt. Wenn ein von **XDM** verwalteter Bildschirm aus unbekannten Gründen hängen bleibt, sollten Sie in dieser Datei nach Fehlermeldungen suchen. Für jede Sitzung werden die Meldungen auch in die Datei `~/.xsession-errors` des Benutzers geschrieben.

6.6.4. Einrichten eines Bildschirm-Servers auf dem Netzwerk

Damit sich Clients mit dem Bildschirm-Server verbinden können, muss der Zugriffsregelsatz editiert und der Listener aktiviert werden. Die Vorgabewerte sind sehr restriktiv eingestellt. Damit **XDM** Verbindungen annimmt, kommentieren Sie eine Zeile in der `xdm-config` Datei aus:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

Starten Sie danach **XDM** neu. Beachten Sie, dass Kommentare in den Ressourcen-Konfigurationsdateien mit einem `!` anstelle des sonst üblichen Zeichens # beginnen. Wenn Sie strengere Zugriffskontrollen einrichten wollen, sehen Sie sich die Beispiele in `Xaccess` und die Hilfeseite xdm(1) an.

6.6.5. XDM ersetzen

Es gibt mehrere Anwendungen, die **XDM** ersetzen können, zum Beispiel **kdm**, der Teil von **KDE** ist und später in diesem Kapitel besprochen wird. **kdm** ist ansprechender gestaltet und bietet neben einigen Schnörkeln die Möglichkeit, den zu verwendenden Window-Manager bei der Anmeldung auszuwählen.

6.7. Grafische Oberflächen

Beigetragen von Valentino Vaschetto.

Dieser Abschnitt beschreibt verschiedene grafische Oberflächen, die es für X unter FreeBSD gibt. Eine Oberfläche (*desktop environment*) kann alles von einem einfachen Window-Manager bis hin zu kompletten Anwendungen wie **KDE** oder **GNOME** sein.

6.7.1. GNOME

6.7.1.1. Über GNOME

GNOME ist eine benutzerfreundliche Oberfläche, mit der Rechner leicht benutzt und konfiguriert werden können. **GNOME** besitzt eine Leiste, mit der Anwendungen gestartet werden und die Statusinformationen anzeigen kann. Programme und Daten können auf der Oberfläche abgelegt werden und Standardwerkzeuge stehen zur Verfügung. Es gibt Konventionen, die es Anwendungen leicht machen, zusammenzuarbeiten und ein konsistentes Erscheinungsbild garantieren. Benutzer anderer Betriebssysteme oder anderer Arbeitsumgebungen sollten mit der leistungsfähigen grafischen Oberfläche von **GNOME** sehr gut zurechtkommen. Auf der Webseite FreeBSD GNOME Project (http://www.FreeBSD.org/gnome) finden Sie weitere Informationen über GNOME auf FreeBSD. Zusätzlich finden Sie dort umfassende FAQs zur Installation, Konfiguration und zum Betrieb von **GNOME**.

6.7.1.2. GNOME installieren

Am einfachsten installieren Sie **GNOME** als Paket oder über die Ports-Sammlung.

Wenn Sie das **GNOME**-Paket über das Netz installieren wollen, setzen Sie den nachstehenden Befehl ab:

```
# pkg_add -r gnome2
```

Wenn Sie den Quellcode von **GNOME** übersetzen wollen, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/x11/gnome2
# make install clean
```

Damit **GNOME** korrekt funktioniert, muss das `/proc`-Dateisystem eingehängt sein. Fügen Sie daher die folgende Zeile in `/etc/fstab` ein, damit procfs(5) beim Systemstart automatisch eingehängt wird:

```
proc           /proc       procfs  rw  0   0
```

Nachdem **GNOME** installiert ist, muss der X-Server **GNOME** anstelle eines Window-Managers starten.

Der einfachste Weg, **GNOME** zu starten, ist **GDM**, der GNOME Display Manager. **GDM** wird zwar als Teil des **GNOME**-Desktops installiert, ist aber in der Voreinstellung deaktiviert. Um **GDM** zu aktivieren, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
gdm_enable="YES"
```

Nach einem Systemneustart wird **GDM** ab sofort automatisch gestartet.

In der Regel ist es ratsam, alle **GNOME**-Dienste beim Start von **GDM** zu aktivieren. Um dies zu erreichen, fügen Sie die folgende Zeile in `/etc/rc.conf` ein:

```
gnome_enable="YES"
```

GNOME kann auch von der Kommandozeile gestartet werden, wenn Sie eine entsprechend konfigurierte `.xinitrc` in Ihrem Heimatverzeichnis besitzen. Existiert eine solche Version, ersetzen Sie den Aufruf des Window-Managers durch **/usr/local/bin/gnome-session**. Wenn `.xinitrc` nicht gesondert angepasst wurde, reicht es, den nachstehenden Befehl abzusetzen:

```
% echo "/usr/local/bin/gnome-session" > ~/.xinitrc
```

Rufen Sie danach `startx` auf, um die **GNOME** Oberfläche zu starten.

> **Anmerkung:** Wenn Sie einen älteren Display-Manager wie **XDM** verwenden, müssen Sie anders vorgehen. Legen Sie eine ausführbare `.xsession` an, die das Kommando zum Start von **GNOME** enthält. Ersetzen Sie dazu den Start des Window-Managers durch **/usr/local/bin/gnome-session**:
>
> ```
> % echo "#!/bin/sh" > ~/.xsession
> % echo "/usr/local/bin/gnome-session" >> ~/.xsession
> % chmod +x ~/.xsession
> ```

Sie können den Display-Manager auch so konfigurieren, dass der Window-Manager beim Anmelden gewählt werden kann. Im Abschnitt Details zu KDE wird das für **kdm**, den Display-Manager von **KDE** erklärt.

6.7.2. KDE

6.7.2.1. Über KDE

KDE ist eine moderne, leicht zu benutzende Oberfläche, die unter anderem Folgendes bietet:

- eine schöne und moderne Oberfläche,
- eine Oberfläche, die völlig netzwerktransparent ist,
- ein integriertes Hilfesystem, das bequem und konsistent Hilfestellungen bezüglich der Bedienung der **KDE**-Oberfläche und ihrer Anwendungen gibt,
- ein konstantes Erscheinungsbild (*look and feel*) aller **KDE**-Anwendungen,
- einheitliche Menüs, Werkzeugleisten, Tastenkombinationen und Farbschemata,
- Internationalisierung: **KDE** ist in mehr als 40 Sprachen erhältlich,

- durch Dialoge gesteuerte zentrale Konfiguration der Oberfläche,
- viele nützliche **KDE**-Anwendungen.

In **KDE** ist mit **Konqueror** auch ein Webbrowser enthalten, der sich durchaus mit anderen Webbrowsern auf UNIX-Systemen messen kann. Weitere Informationen über **KDE** erhalten Sie auf den KDE-Webseiten (http://www.kde.de/). Auf der Webseite KDE on FreeBSD (http://freebsd.kde.org/) finden Sie weitere FreeBSD-spezifische Informationen über KDE.

Es sind zwei Versionen von **KDE** unter FreeBSD verfügbar. Version 3 ist schon seit einiger Zeit erhältlich und ist sehr ausgereift. Version 4, die nächste Generation, ist ebenfalls über die Ports-Sammlung verfügbar. Beide Versionen können sogar gleichzeitig installiert werden.

6.7.2.2. KDE installieren

Am einfachsten installieren Sie **KDE**, wie jede andere grafische Oberfläche auch, als Paket oder über die Ports-Sammlung.

Um **KDE3** über das Netz zu installieren, setzen Sie den nachstehenden Befehl ab:

```
# pkg_add -r kde
```

Um **KDE4** über das Netzwerk zu installieren, geben Sie folgendes ein:

```
# pkg_add -r kde4
```

pkg_add(1) installiert automatisch die neuste Version einer Anwendung.

Benutzen Sie die Ports-Sammlung, wenn Sie den Quellcode von **KDE3** übersetzen wollen:

```
# cd /usr/ports/x11/kde3
# make install clean
```

Um **KDE4** aus dem Quellcode zu übersetzen, geben Sie folgendes ein:

```
# cd /usr/ports/x11/kde4
# make install clean
```

Nachdem **KDE** installiert ist, muss der X-Server **KDE** anstelle eines Window-Managers starten. Legen Sie dazu die Datei `.xinitrc` an:

Für **KDE3**:

```
% echo "exec startkde" > ~/.xinitrc
```

Für **KDE4**:

```
% echo "exec /usr/local/kde4/bin/startkde" > ~/.xinitrc
```

Wenn das X-Window-System danach mit `startx` gestartet wird, erscheint die **KDE**-Oberfläche.

Wird ein Display-Manager wie **XDM** benutzt, muss `.xsession` angepasst werden. Eine Anleitung für **kdm** folgt gleich in diesem Kapitel.

6.7.3. Details zu KDE

Wenn **KDE** erst einmal installiert ist, erschließen sich die meisten Sachen durch das Hilfesystem oder durch Ausprobieren. Benutzer von Windows oder Mac OS werden sich sehr schnell zurecht finden.

Die beste Referenz für **KDE** ist die Online-Dokumentation. **KDE** besitzt einen eigenen Webbrowser, sehr viele nützliche Anwendungen und ausführliche Dokumentation. Der Rest dieses Abschnitts beschäftigt sich daher mit Dingen, die schlecht durch einfaches Ausprobieren erlernbar sind.

6.7.3.1. Der KDE-Display-Manager

Der Administrator eines Mehrbenutzersystems will den Benutzern vielleicht eine grafische Anmeldung wie mit XDM ermöglichen. **KDE** besitzt einen eigenen Display-Manager, der schöner aussieht und auch über mehr Optionen verfügt. Insbesondere können sich die Benutzer die Oberfläche für die Sitzung (beispielsweise **KDE** oder **GNOME**) aussuchen.

Die Art und Weise, wie **kdm** aktiviert wird, hängt dabei von der von Ihnen eingesetzten **KDE**-Version ab.

Für **KDE3** müssen die `ttyv8`-Zeile wie folgt anpassen:

```
ttyv8 "/usr/local/bin/kdm -nodaemon" xterm on secure
```

Verwenden Sie hingegen **KDE4**, müssen Sie folgende Zeilen in die Datei `/etc/rc.conf` aufnehmen:

```
local_startup="${local_startup} /usr/local/kde4/etc/rc.d"
kdm4_enable="YES"
```

6.7.4. Xfce

6.7.4.1. Über Xfce

Xfce ist eine grafische Oberfläche, die auf den GTK+-Bibliotheken, die auch von **GNOME** benutzt werden, beruht. Die Oberfläche ist allerdings weniger aufwändig und für diejenigen gedacht, die eine schlichte und effiziente Oberfläche wollen, die dennoch einfach zu benutzen und zu konfigurieren ist. Die Oberfläche sieht ähnlich wie **CDE** aus, das in kommerziellen UNIX Systemen verwendet wird. Einige Merkmale von **Xfce** sind:

- eine schlichte einfach zu benutzende Oberfläche,
- vollständig mit Mausoperationen konfigurierbar, Unterstützung von *drag and drop*,
- ähnliche Hauptleiste wie **CDE**, die Menüs enthält und über die Anwendungen gestartet werden können,
- integrierter Window-Manager, Datei-Manager und Sound-Manager, **GNOME**-compliance-Modul,
- mit *Themes* anpassbar (da GTK+ benutzt wird),
- schnell, leicht und effizient: ideal für ältere oder langsamere Maschinen oder Maschinen mit wenig Speicher.

Weitere Information über **Xfce** erhalten Sie auf der Xfce-Webseite (http://www.xfce.org/).

6.7.4.2. Xfce installieren

Das **Xfce**-Paket installieren Sie mit dem nachstehenden Kommando:

```
# pkg_add -r xfce4
```

Mit der Ports-Sammlung können Sie auch den Quellcode übersetzen:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Damit beim nächsten Start des X-Servers **Xfce** benutzt wird, setzen Sie das folgende Kommando ab:

```
% echo "/usr/local/bin/startxfce4" > ~/.xinitrc
```

Wenn Sie einen Display-Manager benutzen, erstellen Sie die Datei `.xsession`, wie im GNOME Abschnitt beschrieben. Verwenden Sie jetzt allerdings das Kommando `/usr/local/bin/startxfce4`. Sie können auch den Display-Manager wie im kdm Abschnitt beschrieben, so konfigurieren, dass die Oberfläche für die Sitzung ausgewählt werden kann.

II. Oft benutzte Funktionen

Nach den Grundlagen beschäftigt sich das FreeBSD-Handbuch mit oft benutzten Funktionen von FreeBSD. Die Kapitel behandeln die nachstehenden Themen:

- Zeigen Ihnen beliebte und nützliche Werkzeuge wie Browser, Büroanwendungen und Programme zum Anzeigen von Dokumenten.
- Zeigen Ihnen Multimedia-Werkzeuge für FreeBSD.
- Erklären den Bau eines angepassten FreeBSD-Kernels, der die Systemfunktionen erweitert.
- Beschreiben ausführlich das Drucksystem, sowohl für direkt angeschlossene Drucker als auch für Netzwerkdrucker.
- Erläutern, wie Sie Linux-Anwendungen auf einem FreeBSD-System laufen lassen.

Damit Sie einige Kapitel verstehen, sollten Sie vorher andere Kapitel gelesen haben. Die Übersicht zu jedem Kapitel zählt die Voraussetzungen für das erolgreiche Durcharbeiten des Kapitels auf.

Kapitel 7. Desktop-Anwendungen

Beigetragen von Christophe Juniet. Übersetzt von Martin Heinen.

7.1. Übersicht

FreeBSD bietet eine reiche Auswahl an Desktop-Anwendungen, wie Browser und Textverarbeitungen, die als Pakete oder mit der Ports-Sammlung installiert werden. Gerade neue Benutzer erwarten Anwendungen mit einer grafischen Benutzeroberfläche an ihrem Arbeitsplatz. Dieses Kapitel zeigt Ihnen, wie Sie einige der beliebtesten Desktop-Anwendungen mühelos installieren.

Wenn Sie Ports installieren, beachten Sie, dass dabei die Quelltexte der Programme übersetzt werden. Abhängig von dem Programm und der Geschwindigkeit Ihrer Maschinen kann das sehr lange dauern. Wenn Ihnen das Übersetzen zu lange dauert, können Sie die meisten Programme der Ports-Sammlung auch als fertige Pakete installieren.

Da FreeBSD binär kompatibel zu Linux ist, können Sie zahlreiche für Linux entwickelte Desktop-Anwendungen einsetzen. Bevor Sie allerdings Linux-Anwendungen installieren, sollten Sie das Kapitel 11 lesen. Wenn Sie nach einem bestimmten Port suchen, zum Beispiel mit whereis(1), beachten Sie, dass die Namen vieler Programme, die die Linux-Binärkompatibilität benutzen, mit `linux-` anfangen. Wir gehen im Folgenden davon aus, dass Sie die Linux-Binärkompatibilität aktiviert haben, bevor Sie Linux-Anwendungen installieren.

Dieses Kapitel behandelt Anwendungen aus den Bereichen:

- Browser (**Firefox**, **Opera**, **Konqueror**), **Chromium**)
- Büroanwendungen (**KOffice**, **AbiWord**, **The GIMP**, **OpenOffice.org**, **LibreOffice**)
- Dokumentformate(**Acrobat Reader®**, **gv**, **Xpdf**, **GQview**)
- Finanzsoftware (**GnuCash**, **Gnumeric**, **Abacus**)

Bevor Sie dieses Kapitel lesen, sollten Sie

- Software Dritter installieren können (Kapitel 5) und
- Linux-Anwendungen installieren können (Kapitel 11).

Wie Sie Multimedia-Anwendungen einrichten, wird in einem gesonderten Kapitel erklärt. Wie Sie E-Mail einrichten und benutzen, wird in Kapitel 29 beschrieben.

7.2. Browser

FreeBSD besitzt keinen vorinstallierten Browser, stattdessen enthält das www (http://www.FreeBSD.org/ports/www.html)-Verzeichnis der Ports-Sammlung Browser, die Sie installieren können. Wenn Ihnen das Übersetzen der Browser zu lange dauert, bei einigen Browsern dauert das wirklich lange, installieren Sie die Pakete, die es für viele Browser gibt.

KDE und **GNOME** enthalten schon HTML-Browser. Das Einrichten dieser grafischen Benutzeroberflächen ist in Abschnitt 6.7 beschrieben.

Wenn Sie besonders schlanke Browser benötigen, suchen Sie in der Ports-Sammlung nach `www/dillo2`, `www/links` oder `www/w3m`.

Dieser Abschnitt behandelt die nachstehenden Anwendungen:

Anwendung	Ressourcenbedarf	Installationsaufwand aus den Ports	wichtige Abhängigkeiten
Firefox	mittel	hoch	**Gtk+**
Opera	niedrig	niedrig	Es gibt eine FreeBSD- und eine Linux-Version. Die Linux-Version hängt von der Linux-Kompatibilität (*Linux Binary Compatibility*) und **linux-openmotif** ab.
Konqueror	mittel	hoch	**KDE**-Biliotheken
Chromium	mittel	mittel	**Gtk+**

7.2.1. Firefox

Firefox ist ein moderner, freier und stabiler Open-Source Browser, der vollständig auf FreeBSD portiert wurde. Er bietet eine dem HTML-Standard konforme Anzeige, Browserfenster als Tabs, Blockierung von Werbefenstern, Erweiterungen, verbesserte Sicherheit und mehr. **Firefox** basiert auf der **Mozilla** Codebasis.

Das Paket können Sie mit dem nachstehenden Befehl installieren:

```
# pkg_add -r firefox
```

Damit installieren Sie **Firefox** 10.0, wenn Sie stattdessen **Firefox** 3.6 einsetzen möchten, geben Sie folgenden Befehl ein:

```
# pkg_add -r firefox36
```

Alternativ können Sie auch die Ports-Sammlung verwenden, um das Programm aus dem Quellcode zu installieren:

```
# cd /usr/ports/www/firefox
# make install clean
```

Ersetzen Sie im vorherigen Kommando `firefox` durch `firefox36`, falls Sie **Firefox** 3.6 verwenden wollen.

7.2.2. Firefox und das Java™-Plugin

> **Anmerkung:** Dieser und die beiden nächsten Abschnitte gehen davon aus, dass Sie **Firefox** bereits installiert haben.

Die Schritte zur Installation des Plugins hängen davon, welche **Firefox** Sie installiert haben.

Installieren Sie das **OpenJDK 6** über die Ports-Sammlung:

```
# cd /usr/ports/java/openjdk6
# make install clean
```

Danach installieren Sie den Port `java/icedtea-web`:

```
# cd /usr/ports/java/icedtea-web
# make install clean
```

Stellen Sie dabei sicher, dass Sie jeweils die Standardoptionen verwenden.

Starten Sie nun Ihren Browser, geben Sie in der Adresszeile `about:plugins` ein und bestätigen Sie diese Eingabe mit der **Enter**-Taste. Dadurch wird eine Seite geöffnet, die alle installierten Plugins auflistet. In dieser Liste sollte sich nun auch das **Java™**-Plugin befinden.

Wird das Plugin nicht gefunden, muss für jeden Benutzer der folgende Befel ausgeführt werden:

```
% ln -s /usr/local/lib/IcedTeaPlugin.so \
  $HOME/.mozilla/plugins/
```

7.2.3. Firefox und das Adobe® Flash™-Plugin

Das Adobe® Flash™-Plugin ist für FreeBSD nicht verfügbar. Es existiert jedoch ein Software-Layer (ein sogenannter Wrapper), der es erlaubt, die Linux-Version des Plugins unter FreeBSD einzusetzen. Dieser Wrapper unterstützt außerdem das Adobe Acrobat®-Plugin, das RealPlayer®-Plugin und andere mehr.

Je nachdem, welche Version von FreeBSD Sie verwenden, sind unterschiedliche Schritte notwendig:

1. Für FreeBSD 7.X

   Installieren Sie den Port `www/nspluginwrapper`. Dieser Port setzt voraus, dass Sie den Port `emulators/linux_base-fc4` bereits installiert haben, der sehr gross ist.

   Anschließend installieren Sie den Port `www/linux-flashplugin9`. Dadurch wird Flash 9.X installiert, denn diese Version läuft zuverlässig auf FreeBSD 7.X.

2. Für FreeBSD 8.X oder Neuere

   Installieren Sie den Port `www/nspluginwrapper`. Dieser Port benötigt den `emulators/linux_base-f10` Port, der sehr gross ist.

   Als nächstes installieren Sie Flash 11.X aus dem Port `www/linux-f10-flashplugin11`.

   Für diese Version muss der folgende symbolische Link angelegt werden:

   ```
   # ln -s /usr/local/lib/npapi/linux-f10-flashplugin/libflashplayer.so \
     /usr/local/lib/browser_plugins/
   ```

   Falls das Verzeichnis `/usr/local/lib/browser_plugins` auf Ihrem System nicht existiert, müssen Sie es manuell anlegen.

Sobald der richtige Flash-Port passend zu ihrer FreeBSD Version installiert ist, muss das Plugin von jedem Benutzer mittels `nspluginwrapper` installiert werden:

```
% nspluginwrapper -v -a -i
```

Das Linux Prozessdateisystem, linprocfs(5), muss unter `/compat/linux/proc` eingehängt werden, wenn Sie Flash-Animationen abspielen möchten. Dies kann mittels des folgenden Kommandos geschehen:

```
# mount -t linprocfs linproc /compat/linux/proc
```

Dieser Schritt kann automatisiert zur Bootzeit ablaufen, indem Sie die passende Zeile in `/etc/fstab` eintragen:

```
linproc   /compat/linux/proc  linprocfs       rw      0       0
```

Rufen Sie dann Ihren Browser auf und geben in der Adresszeile `about:plugins` ein. Diese Eingabe muss mit der **Enter**-Taste bestätigt werden. Danach wird eine Seite geladen, auf der alle installierten Plugins aufgelistet werden.

7.2.4. Firefox und das Swfdec Flash-Plugin

Swfdec ist die Bibliothek zum Dekodieren und Rendern von Flash Animationen. Swfdec-Mozilla ist ein Plugin für **Firefox**-Browser, welches die Swfdec-Bibliothek zum Abspielen von SWF-Dateien benutzt. Momentan befindet sie sich noch in der Entwicklung.

Wenn Sie diese nicht übersetzen können oder wollen, dann installieren Sie einfach das Paket aus dem Netz:

```
# pkg_add -r swfdec-plugin
```

Wenn das Paket nicht verfügbar ist, können Sie es auch über die Ports-Sammlung bauen und installieren:

```
# cd /usr/ports/www/swfdec-plugin
# make install clean
```

Starten Sie anschliessend ihren Browser neu, damit dieses Plugin aktiviert wird.

7.2.5. Opera

Opera ist ein schneller, vollwertiger und standardkonformer Browser, der wie Mozilla über einen eingebauten E-Mail- und Newsreader verfügt. Zusätzlich sind ein IRC-Client, ein RSS/Atom-Feeds-Reader sowie weitere Programme enthalten. Dennoch handelt es sich bei **Opera** weiterhin um ein relativ kleines und sehr schnelles Programmpaket. Sie haben die Wahl zwei Versionen dieses Browsers: Der “nativen” FreeBSD-Version und der Linux-Version.

Wenn Sie das Web mit der FreeBSD-Version von **Opera** erkunden wollen, installieren Sie das Paket:

```
# pkg_add -r opera
```

Einige FTP-Server haben nicht alle Pakete, Sie können **Opera** aber über die Ports-Sammlung installieren:

```
# cd /usr/ports/www/opera
# make install clean
```

Wenn Sie die Linux-Version des Browsers verwenden wollen, ersetzen Sie in den Beispielen `opera` durch `linux-opera`.

Das Adobe Flash-Plugin ist für FreeBSD nicht verfügbar. Es gibt aber eine Linux-Version des Plugins, die auch unter FreeBSD installiert werden kann. Dazu installieren Sie zuerst den Port `www/linux-f10-flashplugin11`, danach den Port `www/opera-linuxplugins`:

```
# cd /usr/ports/www/linux-f10-flashplugin11
# make install clean
# cd /usr/ports/www/opera-linuxplugins
# make install clean
```

Danach sollte das Plugin installiert sein. Um dies zu überprüfen, starten Sie den Browser und geben in die Adresszeile `opera:plugins` ein und bestätigen diese Eingabe mit der **Return**-Taste. Dadurch erhalten Sie eine Liste aller derzeit installierter Plugins.

Um das **Java**-Plugin zu installieren, folgen Sie bitte den entsprechenden Anweisungen für Firefox.

7.2.6. Konqueror

Konqueror ist Teil von **KDE**, kann aber außerhalb von **KDE** benutzt werden, wenn der Port `x11/kdebase3` installiert ist. **Konqueror** ist mehr als nur ein Browser. Sie können das Programm weiters zur Dateiverwaltung und zum Abspielen von Multimedia-Dateien benutzen.

Der Port `misc/konq-plugins` installiert verschiedene Plugins für **Konqueror**.

Konqueror kann **Flash**-Seiten darstellen. Wie Sie die **Flash**-Unterstützung aktiviern, können Sie unter http://freebsd.kde.org/howtos/konqueror-flash.php nachlesen.

7.2.7. Chromium

Chromium ist ein quelloffenes Browserprojekt mit dem Ziel ein sicheres, schnelleres und stabileres Surferlebnis im Web zu ermöglichen. **Chromium** ermöglicht surfen mit Tabs, Blockieren von Pop-Ups, Erweiterungen und vieles mehr. **Chromium** ist das Open Source Projekt, welches auf dem Google Chrome Webbrowser basiert.

Chromium kann als Paket durch die Eingabe des folgenden Befehls installiert werden:

```
# pkg_add -r chromium
```

Als Alternative kann **Chromium** aus dem Quellcode durch die Ports Collection übersetzt werden:

```
# cd /usr/ports/www/chromium
# make install clean
```

> **Anmerkung: Chromium** wird als `/usr/local/bin/chrome` installiert und nicht als `/usr/local/bin/chromium`.

7.2.8. Chromium und das Java-Plug-In

> **Anmerkung:** Dieser Abschnitt setzt voraus, dass **Chromium** bereits installiert ist.

Installieren Sie **OpenJDK 6** mit Hilfe der Ports Collection durch Eingabe von:

```
# cd /usr/ports/java/openjdk6
# make install clean
```

Als nächstes installieren Sie `java/icedtea-web` aus der Ports Collection:

```
# cd /usr/ports/java/icedtea-web
```

```
# make install clean
```

Starten Sie **Chromium** und geben Sie `about:plugins` in die Addresszeile ein. IcedTea-Web sollte dort als eines der installierten Plug-Ins aufgelistet sein.

Falls **Chromium** das IcedTea-Web Plug-In nicht anzeigt, geben Sie das folgende Kommando ein und starten Sie den Webbrowser anschliessend neu:

```
# mkdir -p /usr/local/share/chromium/plugins
# ln -s /usr/local/lib/IcedTeaPlugin.so \
   /usr/local/share/chromium/plugins/
```

7.2.9. Chromium und das Adobe Flash-Plug-In

> **Anmerkung:** Dieser Abschnitt setzt voraus, dass **Chromium** bereits installiert ist.

Die Konfiguration von **Chromium** und Adobe Flash ist ähnlich zur Anleitung für Firefox. Für genauere Hinweise zur Installation von Adobe Flash auf FreeBSD, wenden Sie sich bitte an diesen Abschnitt. Es sollte keine weitere Konfiguration notwendig sein, da **Chromium** in der Lage ist, Plug-Ins von anderen Browsern mit zu benutzen.

7.3. Büroanwendungen

Neue Benutzer suchen oft ein komplettes Office-Paket oder eine leicht zu bedienende Textverarbeitung. Einige Benutzeroberflächen wie **KDE** enthalten zwar ein Office-Paket, diese werden in der Standardeinstellung unter FreeBSD aber nicht installiert. Unabhängig von der verwendeten Benutzeroberfläche können Sie diverse Office-Pakete aber jederzeit über die Ports-Sammlung installlieren.

Dieser Abschnitt behandelt die nachstehenden Anwendungen:

Anwendung	Ressourcenbedarf	Installationsaufwand aus den Ports	wichtige Abhängigkeiten
KOffice	niedrig	hoch	**KDE**
AbiWord	niedrig	niedrig	**Gtk+** oder **GNOME**
The Gimp	niedrig	hoch	**Gtk+**
OpenOffice.org	hoch	enorm	**JDK™**, **Mozilla**
LibreOffice	etwas hoch	enorm	**Gtk+**, **KDE**/ **GNOME** oder **JDK**

7.3.1. KOffice

Die KDE-Gemeinschaft stellt ein Office-Paket bereit, das auch außerhalb von **KDE** eingesetzt werden kann. Es besteht aus vier, von anderen Office-Paketen bekannten, Komponenten: **KWord** ist die Textverarbeitung, **KSpread** die Tabellenkalkulation, mit **KPresenter** werden Präsentationen erstellt und **Kontour** ist ein Zeichenprogramm.

Stellen Sie vor der Installation des neusten **KOffice** sicher, dass Sie eine aktuelle Version von **KDE** besitzen.

Mit dem folgenden Kommando installieren Sie das **KOffice**-Paket für **KDE4**:

```
# pkg_add -r koffice-kde4
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie bitte die Ports-Sammlung. Wenn Sie beispielsweise **KOffice** für **KDE4** installieren wollen, setzen Sie die nachstehendenen Befehle ab:

```
# cd /usr/ports/editors/koffice-kde4
# make install clean
```

7.3.2. AbiWord

AbiWord ist eine freie Textverarbeitung, die ähnlich wie **Microsoft Word** ist. Sie können damit Artikel, Briefe, Berichte, Notizen usw. verfassen. Das Programm ist sehr schnell, besitzt viele Funktionen und ist sehr benutzerfreundlich.

AbiWord kann viele Dateiformate, unter anderem nicht offene wie `.doc` von Microsoft, importieren und exportieren.

Das **AbiWord**-Paket installieren Sie wie folgt:

```
# pkg_add -r AbiWord
```

Sollte das Paket nicht zur Verfügung stehen, können Sie das Programm mit der Ports-Sammlung, die zudem aktueller als die Pakete ist, übersetzen. Gehen Sie dazu folgendermaßen vor:

```
# cd /usr/ports/editors/AbiWord
# make install clean
```

7.3.3. The GIMP

The GIMP ist ein sehr ausgereiftes Bildverarbeitungsprogramm mit dem Sie Bilder erstellen oder retuschieren können. Sie können es sowohl als einfaches Zeichenprogramm als auch zum retuschieren von Fotografien benutzen. Das Programm besitzt eine eingebaute Skriptsprache und es existieren sehr viele Plug-Ins. **The GIMP** kann Bilder in zahlreichen Formaten lesen und speichern und stellt Schnittstellen zu Scannern und grafischen Tabletts zur Verfügung.

Sie installieren das Paket mit dem nachstehenden Befehl:

```
# pkg_add -r gimp
```

Benutzen Sie die Ports-Sammlung, wenn Ihr FTP-Server das Paket nicht bereitstellt. Im Verzeichnis graphics (http://www.FreeBSD.org/ports/graphics.html) finden Sie das Handbuch **The Gimp Manual**. Sie können alles mit den folgenden Befehlen installieren:

```
# cd /usr/ports/graphics/gimp
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```

Anmerkung: Die Entwickler-Version von **The GIMP** finden Sie im Verzeichnis graphics (http://www.FreeBSD.org/ports/graphics.html) der Ports-Sammlung. Das Handbuch ist im HTML-Format (`graphics/gimp-manual-html`) erhältlich.

7.3.4. OpenOffice.org

OpenOffice.org enthält alles, was von einem Office-Paket erwartet wird: Textverarbeitung, Tabellenkalkulation, Präsentation und ein Zeichenprogramm. Die Bedienung gleicht anderen Office-Paketen und das Programm kann zahlreiche Dateiformate importieren und exportieren. Es gibt lokalisierte Versionen mit angepassten Menüs, Rechtschreibkontrollen und Wörterbüchern.

Die Textverarbeitung von **OpenOffice.org** speichert Dateien im XML-Format. Dadurch wird die Verwendbarkeit der Dateien auf anderen Systemen erhöht und die Handhabung der Daten vereinfacht. Die Tabellenkalkulation besitzt eine Makrosprache und eine Schnittstelle zu Datenbanken. **OpenOffice.org** läuft auf Windows, Solaris™, Linux, FreeBSD und Mac OS X. Weitere Informationen über **OpenOffice.org** finden Sie auf der OpenOffice.org Website (http://www.openoffice.org/). Spezifische Informationen für FreeBSD finden Sie auf der Webseite FreeBSD OpenOffice.org Porting Team (http://porting.openoffice.org/freebsd/). Von dort können Sie auch direkt das OpenOffice-Paket herunterladen.

OpenOffice.org installieren Sie wie folgt:

```
# pkg_add -r openoffice.org
```

Anmerkung: Diese Art der Installation sollte mit einer -RELEASE-Version funktionieren. Verwenden Sie eine andere Version, sollten Sie die Internetseite des FreeBSD **OpenOffice.org** Porting Teams besuchen und das entsprechende Paket herunterladen und über pkg_add(1) installieren, wobei Sie zwischen der aktuellen Version und der Entwicklerversion wählen können.

Nachdem das Paket installiert ist, müssen Sie lediglich folgenden Befehl eingeben, um **OpenOffice.org** zu starten:

```
% openoffice.org
```

Anmerkung: Nach dem ersten Start werden Ihnen einige Fragen gestellt. Außerdem wird in Ihrem Heimatverzeichnis der neue Unterordner `.openoffice.org` angelegt.

Falls die **OpenOffice.org**-Pakete nicht zur Verfügung stehen, können Sie immer noch die Ports-Sammlung benutzen. Beachten Sie aber bitte, dass Sie sehr viel Plattenplatz und Zeit benötigen, um die Quellen zu übersetzen.

```
# cd /usr/ports/editors/openoffice-3
# make install clean
```

Anmerkung: Wenn Sie ein lokalisierte Version bauen wollen, ersetzen Sie den letzten Befehl durch die folgende Zeile:

```
# make LOCALIZED_LANG=Ihre_Sprache install clean
```

Dabei ersetzen Sie *`Ihre_Sprache`* durch den korrekten ISO-Code. Eine Liste der unterstützten Codes enthält die Datei `files/Makefile.localized`, die sich im Portsverzeichnis befindet.

Nachdem die Installation abgeschlossen ist, können Sie **OpenOffice.org** durch folgenden Befehl starten:

```
% openoffice.org
```

7.3.5. LibreOffice

LibreOffice ist ein als freie Software verfügbares Office-Paket, welches von The Document Foundation (http://www.documentfoundation.org/) entwickelt wird, das mit anderen grossen Office-Paketen kompatibel ist und auf einer Vielzahl von Plattformen lauffähig ist. Es ist ein Fork von **OpenOffice.org** unter neuem Namen, der alle notwendigen Anwendungen in einem kompletten Büroanwendungspaket enthält: eine Textverarbeitung, eine Tabellenkalkulation, ein Präsentationsmanager, ein Zeichenprogramm, ein Datenbankmanagementprogramm und ein Werkzeug zum Erstellen und Bearbeiten von mathematischen Formeln. Es steht in einer Reihe von Sprachen zur Verfügung; die Internationalisierung wurde auf die Oberfläche, Rechtschreibkorrektur und die Wörterbücher ausgeweitet.

Das Textverarbeitungsprogramm von **LibreOffice** benutzt ein natives XML-Dateiformat für erhöhte Portabilität und Flexibilität. Die Tabellenkalkulation enthält eine Makrosprache und kann mit externen Datenbanken Verbindungen herstellen. **LibreOffice** ist bereits stabil genug und läuft nativ auf Windows, Linux, FreeBSD und Mac OS X. Weitere Informationen zu **LibreOffice** können auf der LibreOffice Webseite (http://www.libreoffice.org/) abgerufen werden.

Um **LibreOffice** als Paket zu installieren, geben Sie folgenden Befehl ein:

```
# pkg_add -r libreoffice
```

Anmerkung: Dies sollte funktionieren, wenn Sie eine -RELEASE-Version von FreeBSD einsetzen.

Sobald das Paket installiert ist, geben Sie das folgende Kommando ein, um **LibreOffice** zu starten:

```
% libreoffice
```

Anmerkung: Während des ersten Starts werden Sie ein paar Fragen gestellt bekommen und es wird ein Verzeichnis `.libreoffice` in Ihrem Heimatverzeichnis erstellt.

Wenn die **LibreOffice**-Pakete nicht verfügbar sind, haben Sie immer noch die Möglichkeit, den Port zu verwenden. Jedoch müssen Sie bedenken, dass dies eine Menge Speicherplatz benötigt und viel Zeit in Anspruch nimmt, bis der Port fertig gebaut ist.

```
# cd /usr/ports/editors/libreoffice
# make install clean
```

Anmerkung: Wenn Sie eine Version in Ihrer Sprache bauen möchten, ersetzen Sie das vorhergehende Kommando mit dem folgenden:

```
# make LOCALIZED_LANG=ihre_Sprache install clean
```

Sie müssen *ihre_Sprache* mit dem richtigen ISO-Code für ihre Sprache ersetzen. Eine Liste von unterstützten Sprachcodes sind im `Makefile` des Ports als `pre-fetch`-Target verfügbar.

Sobald dies abgeschlossen ist, kann **LibreOffice** mit dem folgenden Befehl gestartet werden:

```
% libreoffice
```

7.4. Anzeigen von Dokumenten

Einige neuere Dokumentformate, die sich aktuell großer Beliebtheit erfreuen, können Sie sich mit den im Basissystem enthaltenen Programmen und Werkzeugen nicht ansehen. Dieser Abschnitt behandelt Programme, mit denen Sie sich Dokumente in unterschiedlichsten Formaten ansehen können.

Die nachstehenden Anwendungen werden behandelt:

Anwendung	Ressourcenbedarf	Installationsaufwand aus den Ports	wichtige Abhängigkeiten
Acrobat Reader	niedrig	niedrig	Linux Binary Compatibility
gv	niedrig	niedrig	**Xaw3d**
Xpdf	niedrig	niedrig	**FreeType**
GQview	niedrig	niedrig	**Gtk+** oder **GNOME**

7.4.1. Acrobat Reader®

Viele Dokumente werden heute im “Portable Document Format” (PDF) zur Verfügung gestellt. PDF-Dokumente schauen Sie sich am Besten mit dem Programm **Acrobat Reader** an, das von Adobe für Linux freigegeben wurde. Da Linux-Programme unter FreeBSD laufen, steht Ihnen das Programm auch hier zur Verfügung.

Um **Acrobat Reader 8** über die Ports-Sammlung zu installieren, geben Sie Folgendes ein:

```
# cd /usr/ports/print/acroread8
# make install clean
```

Aufgrund der Lizenzbedinungen ist eine Paketversion leider nicht verfügbar.

7.4.2. gv

gv kann PostScript- und PDF-Dokumente anzeigen. Es stammt von **ghostview** ab, besitzt aber wegen der **Xaw3d**-Bibliothek eine schönere Benutzeroberfläche. In **gv** können Sie viele Operationen durchführen: Sie können die Ausrichtung und die Papiergröße eines Dokuments ändern, das Dokument skalieren oder die Kantenglättung (*Anti-Aliasing*) aktivieren. Fast jede Operation kann sowohl mit der Tastatur als auch mit der Maus durchgeführt werden.

Installieren Sie das **gv**-Paket wie folgt:

```
# pkg_add -r gv
```

Benutzen Sie die Ports-Sammlung, wenn das Paket nicht zur Verfügung steht:

```
# cd /usr/ports/print/gv
# make install clean
```

7.4.3. Xpdf

Ein schlankes und effizientes Programm zum Betrachten von PDF-Dateien ist **Xpdf**. Es benötigt wenige Ressourcen und ist sehr stabil. Da das Programm die Standard X-Zeichensätze benutzt, ist es nicht auf Motif oder ein anderes X-Toolkit angewiesen.

Das **Xpdf**-Paket können Sie mit dem folgenden Kommando installieren:

```
# pkg_add -r xpdf
```

Wenn das Paket nicht verfügbar ist, oder Sie lieber die Ports-Sammlung benutzen möchten, gehen Sie wie folgt vor:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

Wenn Sie nach Abschluss der Installation **Xpdf** starten, öffnen Sie das Menü mit der rechten Maustaste.

7.4.4. GQview

Mit **GQview** lassen sich Bilder verwalten. Unter anderem können Sie sich Bilder (auch auf dem ganzen Bildschirm) anschauen, ein externes Werkzeug aufrufen und eine Vorschau (*thumbnail*) erzeugen. Weiterhin können Sie automatisch ablaufende Präsentationen erstellen und grundlegende Dateioperationen durchführen, Bildersammlungen verwalten und doppelte Bilder aufspüren. **GQview** ist internationalisiert, das heißt es berücksichtigt die Spracheinstellungen des Systems.

Wenn Sie das **GQview**-Paket installieren wollen, geben Sie das folgende Kommando ein:

```
# pkg_add -r gqview
```

Ist das Paket nicht erhältlich, oder wenn Sie die Ports-Sammlung bevorzugen, setzen Sie die folgenden Kommandos ab:

```
# cd /usr/ports/graphics/gqview
# make install clean
```

7.5. Finanzsoftware

Wenn Sie, warum auch immer, Ihre Finanzen mit einem FreeBSD Arbeitsplatz verwalten wollen, stehen Ihnen verschiedene Anwendungen zur Verfügung. Einige von ihnen unterstützen verbreitete Formate, darunter

Dateiformate, die von **Quicken®** oder **Excel** verwendet werden.

Dieser Abschnitt behandelt die folgenden Anwendungen:

Anwendung	Ressourcenbedarf	Installationsaufwand aus den Ports	wichtige Abhängigkeiten
GnuCash	niedrig	hoch	**GNOME**
Gnumeric	niedrig	hoch	**GNOME**
Abacus	niedrig	niedrig	**Tcl/Tk**
KMyMoney	niedrig	hoch	**KDE**

7.5.1. GnuCash

GnuCash ist Teil des **GNOME**-Projekts, dessen Ziel es ist, leicht zu bedienende und doch leistungsfähige Anwendungen zu erstellen. Mit **GnuCash** können Sie Ihre Einnahmen und Ausgaben, Ihre Bankkonten und Wertpapiere verwalten. Das Programm ist leicht zu bedienen und genügt dennoch hohen Ansprüchen.

GnuCash stellt ein Register, ähnlich dem in einem Scheckheft und ein hierarchisches System von Konten zur Verfügung. Eine Transaktion kann in einzelne Teile aufgespaltet werden. **GnuCash** kann Quicken-Dateien (QIF) importieren und einbinden. Weiterhin unterstützt das Programm die meisten internationalen Formate für Zeitangaben und Währungen. Die Bedienung des Programms kann durch zahlreiche Tastenkombinationen und dem automatischen Vervollständigen von Eingaben beschleunigt werden.

Das **GnuCash**-Paket installieren Sie wie folgt:

```
# pkg_add -r gnucash
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

7.5.2. Gnumeric

Gnumeric ist eine Tabellenkalkulation, die Teil der **GNOME** Benutzeroberfläche ist. Das Programm kann Eingaben anhand des Zellenformats oder einer Folge von Eingaben vervollständigen. Dateien verbreiteter Formate, wie die von **Excel**, **Lotus 1-2-3** oder **Quattro Pro** lassen sich importieren. Grafiken erstellt **Gnumeric** mit dem Programm `math/guppi`. **Gnumeric** besitzt viele eingebaute Funktionen und Zellenformate (zum Beispiel die üblich verwendeten, wie Zahl, Währung, Datum oder Zeit).

Installieren Sie das **Gnumeric**-Paket mit dem folgenden Kommando:

```
# pkg_add -r gnumeric
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

7.5.3. Abacus

Abacus ist eine kleine und leicht zu bedienende Tabellenkalkulation. Die vordefinierten Funktionen stammen aus verschiedenen Bereichen wie der Statistik, der Wirtschaft und der Mathematik. Das Programm kann Dateien im **Excel** Dateiformat importieren und exportieren sowie Ausgaben in PostScript erzeugen.

Installieren Sie das **Abacus**-Paket mit dem folgenden Kommando:

```
# pkg_add -r abacus
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

7.5.4. KMyMoney

Bei **KMyMoney** handelt es sich ein Programm zur Verwaltung der persönlichen Finanzen, das unter **KDE** entwickelt wird. **KMyMoney** hat das Ziel, alle wichtigen Funktionen zu bieten, die auch von kommerziellen Programmen zur Verwaltung der persönlichen Finanzen unterstützt werden. Weiters zählen einfache Benutzung sowie korrekte doppelte Buchführung zu den herausragenden Fähigkeiten dieses Programms. **KMyMoney** unterstützt den Import von Datendateien im Format *Quicken Interchange Format (QIF)*, kann Investionen verfolgen, unterstützt verschiedene Währungen und bietet umfangreiche Reportmöglichkeiten. OFX-Import wird über ein separates Plugin realisiert.

Um **KMyMoney** über das FreeBSD-Paketsystem zu installieren, geben Sie Folgendes ein:

```
# pkg_add -r kmymoney2
```

Sollte das Paket nicht verfügbar sein, können Sie das Programm auch über die Ports-Sammlung installieren:

```
# cd /usr/ports/finance/kmymoney2
# make install clean
```

7.6. Zusammenfassung

FreeBSD wird von Internet Service Providern wegen seiner Schnelligkeit und Stabilität eingesetzt, es ist aber auch zum Einrichten eines Arbeitsplatzes geeignet. Mit tausenden Anwendungen, die als Pakete (http://www.FreeBSD.org/applications.html) oder Ports (http://www.FreeBSD.org/ports/index.html) zur Verfügung stehen, können Sie sich einen Arbeitsplatz nach Ihren Wünschen einrichten.

Die folgende Aufstellung fasst die in diesem Kapitel besprochenen Anwendungen zusammen:

Anwendung	Paket-Name	Port-Name
Opera	`opera`	`www/opera`
Firefox	`firefox`	`www/firefox`
Chromium	`chromium`	`www/chromium`
KOffice	`koffice-kde4`	`editors/koffice-kde4`

Anwendung	Paket-Name	Port-Name
AbiWord	`abiword`	`editors/abiword`
The GIMP	`gimp`	`graphics/gimp`
OpenOffice.org	`openoffice`	`editors/openoffice.org-3`
LibreOffice	`libreoffice`	`editors/libreoffice`
Acrobat Reader	`acroread`	`print/acroread8`
gv	`gv`	`print/gv`
Xpdf	`xpdf`	`graphics/xpdf`
GQview	`gqview`	`graphics/gqview`
GnuCash	`gnucash`	`finance/gnucash`
Gnumeric	`gnumeric`	`math/gnumeric`
Abacus	`abacus`	`deskutils/abacus`
KMyMoney	`kmymoney2`	`finance/kmymoney2`

Kapitel 8. Multimedia

Überarbeitet von Ross Lippert.

8.1. Übersicht

FreeBSD unterstützt viele unterschiedliche Soundkarten, die Ihnen den Genuss von Highfidelity-Klängen auf Ihrem Computer ermöglichen. Dazu gehört unter anderem die Möglichkeit, Tonquellen in den Formaten MPEG Audio Layer 3 (MP3), WAV, Ogg Vorbis und vielen weiteren Formaten aufzunehmen und wiederzugeben. Darüber hinaus enthält die FreeBSD Ports-Sammlung Anwendungen, die Ihnen das Bearbeiten Ihrer aufgenommenen Tonspuren, das Hinzufügen von Klangeffekten und die Kontrolle der angeschlossenen MIDI-Geräte erlauben.

Wenn Sie etwas Zeit investieren, können Sie mit FreeBSD auch Videos und DVDs abspielen. Im Vergleich zu Audio-Anwendungen gibt es weniger Anwendungen zum Kodieren, Konvertieren und Abspielen von Video-Formaten. Es gab, als dieses Kapitel geschrieben wurde, keine Anwendung, die einzelne Video-Formate ähnlich wie `audio/sox` konvertieren konnte. Allerdings ändert sich die Software in diesem Umfeld sehr schnell.

In diesem Kapitel wird das Einrichten von Soundkarten besprochen. Kapitel 6 beschreibt die Installation und Konfiguration von X11 und das Einrichten von Videokarten. Hinweise zur Verbesserung der Wiedergabe finden sich in diesem Kapitel.

Dieses Kapitel behandelt die folgenden Punkte:

- Die Konfiguration des Systems damit Ihre Soundkarte erkannt wird.
- Wie Sie die Funktion einer Soundkarte testen können.
- Wie Sie Fehler in den Einstellungen von Soundkarten finden.
- Wie Sie MP3s und andere Audio-Formate wiedergeben und erzeugen.
- Die Video-Unterstützung des X-Servers.
- Gute Anwendungen, die Videos abspielen und kodieren.
- Die Wiedergabe von DVDs, `.mpg`- und `.avi`-Dateien.
- Wie Sie CDs und DVDs in Dateien rippen.
- Die Konfiguration von TV-Karten.
- Das Einrichten von Scannern.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Wissen, wie Sie einen neuen Kernel konfigurieren und installieren (Kapitel 9).

> **Warnung:** Der Versuch eine Audio-CD mit mount(8) einzuhängen erzeugt mindestens einen Fehler; schlimmstenfalls kann es zu einer Kernel-Panic kommen. Die Medien besitzen eine andere Kodierung als normale ISO-Dateisysteme.

8.2. Soundkarten einrichten

Von Moses Moore. Aktualisiert von Marc Fonvieille. Übersetzt von Benedikt Köhler und Uwe Pierau.

8.2.1. Den Soundtreiber einrichten

Zunächst sollten Sie in Erfahrung bringen, welches Soundkartenmodell Sie besitzen, welchen Chip die Karte benutzt und ob es sich um eine PCI- oder ISA-Karte handelt. FreeBSD unterstützt eine Reihe von PCI- als auch von ISA-Karten. Die Hardware-Notes (http://www.FreeBSD.org/releases/9.1R/hardware.html) zählen alle unterstützten Karten und deren Treiber auf.

Um Ihre Soundkarte benutzen zu können, müssen Sie den richtigen Gerätetreiber laden. Sie haben zwei Möglichkeiten, den Treiber zu laden: Am einfachsten ist es, das Modul mit kldload(8) zu laden. Sie können dazu die Kommandozeile verwenden:

```
# kldload snd_emu10k1
```

Alternativ können Sie auch einen Eintrag in der Datei `/boot/loader.conf` erstellen:

```
snd_emu10k1_load="YES"
```

Beide Beispiele gelten für eine Creative SoundBlaster® Live! Soundkarte. Weitere ladbare Soundmodule sind in der Datei `/boot/defaults/loader.conf` aufgeführt. Wenn Sie nicht sicher sind, welchen Gerätetreiber Sie laden müssen, laden Sie den Treiber `snd_driver`:

```
# kldload snd_driver
```

Der Treiber `snd_driver` ist ein Meta-Treiber, der alle gebräuchlichen Treiber lädt und die Suche nach dem richtigen Treiber vereinfacht. Weiterhin können alle Treiber über `/boot/loader.conf` geladen werden.

Wollen Sie feststellen, welcher Treiber für Ihre Soundkarte vom Metatreiber `snd_driver` geladen wurde, sollten Sie sich mit `cat /dev/sndstat` den Inhalt der Datei `/dev/sndstat` ansehen.

Alternativ können Sie die Unterstützung für die Soundkarte direkt in den Kernel einkompilieren. Diese Methode im nächsten Abschnitt beschrieben. Weiteres über den Bau eines Kernels erfahren Sie im Kapitel Kernelkonfiguration.

8.2.1.1. Soundkarten in der Kernelkonfiguration einrichten

Zuerst müssen Sie sound(4), den Treiber für das Audio-Framework in die Kernelkonfiguration aufnehmen. Fügen Sie dazu die folgende Zeile in die Kernelkonfigurationsdatei ein:

```
device sound
```

Als Nächstes müssen Sie den richtigen Treiber in die Kernelkonfiguration einfügen. Den Treiber entnehmen Sie bitte der Liste der unterstützen Soundkarten aus den Hardware-Notes (http://www.FreeBSD.org/releases/9.1R/hardware.html). Zum Beispiel wird die Creative SoundBlaster Live! Soundkarte vom Treiber snd_emu10k1(4) unterstützt. Für diese Karte verwenden Sie die nachstehende Zeile:

```
device snd_emu10k1
```

Die richtige Syntax für die Zeile lesen Sie bitte in der Hilfeseite des entsprechenden Treibers nach. Die korrekte Syntax für alle unterstützten Treiber finden Sie außerdem in der Datei `/usr/src/sys/conf/NOTES`.

Nicht PnP-fähige ISA-Soundkarten benötigen (wie alle anderen ISA-Karten auch) weiterhin Angaben zu den Karteneinstellungen (wie IRQ und I/O-Port). Die Karteneinstellungen tragen Sie in die Datei `/boot/device.hints` ein. Während des Systemstarts liest der loader(8) diese Datei und reicht die Einstellungen an den Kernel weiter. Für eine alte Creative SoundBlaster 16 ISA-Karte, die sowohl den snd_sbc(4)- als auch den `snd_sb16`-Treiber benötigt, fügen Sie folgende Zeilen in die Kernelkonfigurationsdatei ein:

```
device snd_sbc
device snd_sb16
```

In die Datei `/boot/device.hints` tragen Sie für diese Karte zusätzlich die folgenden Einstellungen ein:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

In diesem Beispiel benutzt die Karte den I/O-Port `0x220` und den IRQ `5`.

Die Manualpage sound(4) sowie des jeweiligen Treibers beschreiben die Syntax der Einträge in der Datei `/boot/device.hints`.

Das Beispiel verwendet die vorgegebenen Werte. Falls Ihre Karteneinstellungen andere Werte vorgeben, müssen Sie die Werte in der Kernelkonfiguration anpassen. Weitere Informationen zu dieser Soundkarte entnehmen Sie bitte der Manualpage snd_sbc(4).

8.2.2. Die Soundkarte testen

Nachdem Sie den neuen Kernel gestartet oder das erforderliche Modul geladen haben, sollte Ihre Soundkarte in den Systemmeldungen (dmesg(8)) auftauchen. Zum Beispiel:

```
pcm0: <Intel ICH3 (82801CA)> port 0xdc80-0xdcbf,0xd800-0xd8ff irq 5 at device 31.5 on pci0
pcm0: [GIANT-LOCKED]
pcm0: <Cirrus Logic CS4205 AC97 Codec>
```

Den Status der Karte können Sie über die Datei `/dev/sndstat` prüfen:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm)
Installed devices:
pcm0: <Intel ICH3 (82801CA)> at io 0xd800, 0xdc80 irq 5 bufsz
16384
kld snd_ich (1p/2r/0v channels duplex default)
```

Die Ausgaben können auf Ihrem System anders aussehen. Wenn das Gerät `pcm` nicht erscheint, prüfen Sie bitte Ihre Konfiguration. Stellen sie sicher, dass Sie den richtigen Treiber gewählt haben. Abschnitt 8.2.2.1 beschreibt häufig auftretende Probleme.

Wenn alles glatt lief, haben Sie nun eine funktionierende Soundkarte. Wenn ein CD-ROM oder DVD-ROM-Laufwerk an Ihrer Soundkarte angeschlossen ist, können Sie jetzt mit cdcontrol(1) eine CD abspielen:

```
% cdcontrol -f /dev/acd0 play 1
```

Es gibt viele Anwendungen, wie `audio/workman`, die eine bessere Benutzerschnittstelle besitzen. Um sich MP3-Audiodateien anzuhören, können Sie eine Anwendung wie `audio/mpg123` installieren.

Eine weitere schnelle Möglichkeit die Karte zu prüfen, ist es, Daten an das Gerät `/dev/dsp` zu senden:

```
% cat Datei > /dev/dsp
```

Für *`Datei`* können Sie eine beliebige Datei verwenden. Wenn Sie einige Geräusche hören, funktioniert die Soundkarte.

> **Anmerkung:** Die Gerätedateien `/dev/dsp*` werden automatisch erzeugt, wenn sie das erste Mal benötigt werden. Werden sie nicht verwendet, sind sie hingegen nicht vorhanden und tauchen daher auch nicht in der Ausgabe von ls(1) auf.

Die Einstellungen des Mixers können Sie mit dem Kommando mixer(8) verändern. Weiteres lesen Sie bitte in der Hilfeseite mixer(8) nach.

8.2.2.1. Häufige Probleme

Fehler	Lösung
`sb_dspwr(XX) timed out`	Der I/O Port ist nicht korrekt angegeben.
`bad irq XX`	Der IRQ ist falsch angegeben. Stellen Sie sicher, dass der angegebene IRQ mit dem Sound IRQ übereinstimmt.
`xxx: gus pcm not attached, out of memory`	Es ist nicht genug Speicher verfügbar, um das Gerät zu betreiben.
`xxx: can't open /dev/dsp!`	Überprüfen Sie mit `fstat | grep dsp` ob eine andere Anwendung das Gerät geöffnet hat. Häufige Störenfriede sind **esound** oder die Sound-Unterstützung von **KDE**.

Ein weiterer Fall ist der, dass moderne Graphikkarten oft auch ihre eigenen Soundtreiber mit sich führen, um HDMI oder ähnliches zu verwenden. Diese Audiogeräte werden manchmal vor der eigentlichen, separaten Soundkarte aufgeführt und dadurch nicht als das Standardgerät zum Abspielen von Tönen benutzt. Um zu prüfen, ob das bei Ihnen der Fall ist, führen Sie **dmesg** aus und suchen Sie nach der Zeichenfolge `pcm`. Die Ausgabe sieht in etwa so aus wie folgt:

```
...
hdac0: HDA Driver Revision: 20100226_0142
hdac1: HDA Driver Revision: 20100226_0142
hdac0: HDA Codec #0: NVidia (Unknown)
hdac0: HDA Codec #1: NVidia (Unknown)
hdac0: HDA Codec #2: NVidia (Unknown)
hdac0: HDA Codec #3: NVidia (Unknown)
pcm0: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 0 nid 1 on hdac0
pcm1: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 1 nid 1 on hdac0
pcm2: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 2 nid 1 on hdac0
pcm3: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 3 nid 1 on hdac0
hdac1: HDA Codec #2: Realtek ALC889
pcm4: <HDA Realtek ALC889 PCM #0 Analog> at cad 2 nid 1 on hdac1
pcm5: <HDA Realtek ALC889 PCM #1 Analog> at cad 2 nid 1 on hdac1
```

```
pcm6: <HDA Realtek ALC889 PCM #2 Digital> at cad 2 nid 1 on hdac1
pcm7: <HDA Realtek ALC889 PCM #3 Digital> at cad 2 nid 1 on hdac1
...
```

Hier wurde die Graphikkarte (`NVidia`) vor der Soundkarte (`Realtek ALC889`) aufgeführt. Um die Soundkarte als Standardabspielgerät einzusetzen, ändern Sie `hw.snd.default_unit` auf die Einheit, welche für das Abspielen benutzt werden soll, wie folgt:

```
# sysctl hw.snd.default_unit=n
```

Hier repräsentiert `n` die Nummer der Soundkarte, die verwendet werden soll, in diesem Beispiel also `4`. Sie können diese Änderung dauerhaft machen, indem Sie die folgende Zeile zu der `/etc/sysctl.conf` Datei hinzufügen:

```
hw.snd.default_unit=4
```

8.2.3. Mehrere Tonquellen abspielen

Beigetragen von Munish Chopra.

Oft sollen mehrere Tonquellen gleichzeitig abgespielt werden, auch wenn beispielsweise **esound** oder **artsd** das Audiogerät nicht mit einer anderen Anwendung teilen können.

Unter FreeBSD können mit sysctl(8) *virtuelle Tonkanäle* eingerichtet werden. Virtuelle Kanäle mischen die Tonquellen im Kernel (so können mehr Kanäle als von der Hardware unterstützt benutzt werden).

Die Anzahl der virtuellen Kanäle können Sie als Benutzer `root` wie folgt einstellen:

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

Im Beispiel werden vier virtuelle Kanäle eingerichtet, eine im Normalfall ausreichende Anzahl. Sowohl `dev.pcm.0.play.vchans=4` und `dev.pcm.0.rec.vchans=4` sind die Anzahl der virtuellen Kanäle des Geräts `pcm0`, die fürs Abspielen und Aufnehmen verwendet werden und sie können konfiguriert werden, sobald das Gerät existiert. `hw.snd.maxautovchans` ist die Anzahl der virtuellen Kanäle, die einem Gerät zugewiesen werden, wenn es durch kldload(8) eingerichtet wird. Da das Modul `pcm` unabhängig von den Hardware-Treibern geladen werden kann, gibt `hw.snd.maxautovchans` die Anzahl der virtuellen Kanäle an, die später eingerichtete Geräte erhalten. Lesen Sie dazu pcm(4) für weitere Informationen.

> **Anmerkung:** Sie können die Anzahl der virtuellen Kanäle nur ändern, wenn das Gerät nicht genutzt wird. Schließen Sie daher zuerst alle Programme (etwa Musikabspielprogramme oder Sound-Daemonen), die auf dieses Gerät zugreifen.

Die korrekte `pcm`-Gerätedatei wird automatisch zugeteilt, wenn ein Programm das Gerät `/dev/dsp0` anfordert.

8.2.4. Den Mixer einstellen

Beigetragen von Josef El-Rayes.

Die Voreinstellungen des Mixers sind im Treiber pcm(4) fest kodiert. Es gibt zwar viele Anwendungen und Dienste, die den Mixer einstellen können und die eingestellten Werte bei jedem Start wieder setzen, am einfachsten ist es allerdings, die Standardwerte für den Mixer direkt im Treiber einzustellen. Der Mixer kann in der Datei `/boot/device.hints` eingestellt werden:

```
hint.pcm.0.vol="50"
```

Die Zeile setzt die Lautstärke des Mixers beim Laden des Moduls pcm(4) auf den Wert `50`.

8.3. MP3-Audio

Ein Beitrag von Chern Lee. Übersetzt von Benedikt Köhler.

MP3 (MPEG Layer 3 Audio) ermöglicht eine Klangwiedergabe in CD-ähnlicher Qualität, was Sie sich auf Ihrem FreeBSD-Rechner nicht entgehen lassen sollten.

8.3.1. MP3-Player

XMMS (X Multimedia System) ist bei weitem der beliebteste MP3-Player für X11. **WinAmp**-Skins können auch mit **XMMS** genutzt werden, da die Benutzerschnittstelle fast identisch mit der von Nullsofts **WinAmp** ist. Daneben unterstützt **XMMS** auch eigene Plugins.

XMMS kann als `multimedia/xmms` Port oder Package installiert werden.

Die Benutzerschnittstelle von **XMMS** ist leicht zu erlernen und enthält eine Playlist, einen graphischen Equalizer und vieles mehr. Diejenigen, die mit WinAmp vertraut sind, werden **XMMS** sehr leicht zu benutzen finden.

Der Port `audio/mpg123` ist ein alternativer, kommandozeilenorientierter MP3-Player.

mpg123 kann ausgeführt werden, indem man das zu benutzende Sound Device und die abzuspielende MP3-Datei auf der Kommandozeile angibt. Wenn ihr Sound Device beispielsweise `/dev/dsp1.0` lautet und Sie die MP3-Datei *`Foobar-GreatestHits.mp3`* hören wollen, geben Sie Folgendes ein:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15).  Written and copyrights by Michael Hipp.
Uses code from various people.  See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!




Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

8.3.2. CD-Audio Tracks rippen

Bevor man eine ganze CD oder einen CD-Track in das MP3-Format umwandeln kann, müssen die Audiodaten von der CD auf die Festplatte gerippt werden. Dabei werden die CDDA (CD Digital Audio) Rohdaten in WAV-Dateien kopiert.

Die Anwendung `cdda2wav` die im `sysutils/cdrtools` Paket enthalten ist, kann zum Rippen der Audiodaten und anderen Informationen von CDs genutzt werden.

Wenn die Audio CD in dem Laufwerk liegt, können Sie mit folgendem Befehl (als `root`) eine ganze CD in einzelne WAV-Dateien (eine Datei für jeden Track) rippen:

```
# cdda2wav -D 0,1,0 -B
```

cdda2wav unterstützt auch ATAPI (IDE) CD-ROM-Laufwerke. Um von einem IDE-Laufwerk zu rippen, übergeben Sie auf der Kommandozeile statt der SCSI-IDs den Gerätenamen. Das folgende Kommando rippt den 7. Track:

```
# cdda2wav -D /dev/acd0 -t 7
```

Der Schalter `-D 0,1,0` bezieht sich auf das SCSI Device `0,1,0`, das sich aus dem Ergebnis des Befehls `cdrecord -scanbus` ergibt.

Um einzelne Tracks zu rippen, benutzen Sie den `-t` Schalter wie folgt:

```
# cdda2wav -D 0,1,0 -t 7
```

Dieses Beispiel rippt den siebten Track der Audio CD-ROM. Um mehrere Tracks zu rippen, zum Beispiel die Tracks eins bis sieben, können Sie wie folgt einen Bereich angeben:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Mit dd(1) können Sie ebenfalls Audio-Stücke von ATAPI-Laufwerken kopieren. Dies wird in Abschnitt 19.6.5 erläutert.

8.3.3. MP3-Dateien kodieren

Gegenwärtig ist **Lame** der meistbenutzte MP3-Encoder. **Lame** finden Sie unter `audio/lame` im Ports-Verzeichnis.

Benutzen Sie die WAV-Dateien, die sie von CD gerippt haben, und wandeln sie mit dem folgenden Befehl die Datei *audio01.wav* in *audio01.mp3* um:

```
# lame -h -b 128 \
--tt "Foo Liedtitel" \
--ta "FooBar Künstler" \
--tl "FooBar Album" \
--ty "2001" \
--tc "Gerippt und kodiert von Foo" \
--tg "Musikrichtung" \
audio01.wav audio01.mp3
```

128 kbits ist die gewöhnliche MP3-Bitrate. Viele bevorzugen mit 160 oder 192 kbits eine höhere Qualität. Je höher die Bitrate ist, desto mehr Speicherplatz benötigt die resultierende MP3-Datei, allerdings wird die Qualität dadurch auch besser. Der Schalter `-h` verwendet den “higher quality but a little slower” (höhere Qualität, aber etwas langsamer) Modus. Die Schalter, die mit `--t` beginnen, sind ID3-Tags, die in der Regel Informationen über das Lied

enthalten und in die MP3-Datei eingebettet sind. Weitere Optionen können in der Manualpage von **Lame** nachgelesen werden.

8.3.4. MP3-Dateien dekodieren

Um aus MP3-Dateien eine Audio CD zu erstellen, müssen diese in ein nicht komprimiertes WAV-Format umgewandelt werden. Sowohl **XMMS** als auch **mpg123** unterstützen die Ausgabe der MP3-Dateien in unkomprimierte Dateiformate.

Dekodieren mit **XMMS**:

1. Starten Sie **XMMS**.
2. Klicken Sie mit der rechten Maustaste, um das **XMMS**-Menu zu öffnen.
3. Wählen Sie `Preference` im Untermenü `Options`.
4. Ändern Sie das Output-Plugin in “Disk Writer Plugin”.
5. Drücken Sie `Configure`.
6. Geben Sie ein Verzeichnis ein (oder wählen Sie browse), in das Sie die unkomprimierte Datei schreiben wollen.
7. Laden Sie die MP3-Datei wie gewohnt in **XMMS** mit einer Lautstärke von 100% und einem abgeschalteten EQ.
8. Drücken Sie `Play` und es wird so aussehen, als spiele **XMMS** die MP3-Datei ab, aber keine Musik ist zu hören. Der Player überspielt die MP3-Datei in eine Datei.
9. Vergessen Sie nicht, das Output-Plugin wieder in den Ausgangszustand zurückzusetzen um wieder MP3-Dateien anhören zu können.

Mit **mpg123** nach stdout schreiben:

1. Geben Sie `mpg123 -s` *`audio01.mp3`* `>` *`audio01.pcm`* ein.

XMMS schreibt die Datei im WAV-Format aus während **mpg123** die MP3-Datei in rohe PCM-Audiodaten umwandelt. **cdrecord** kann mit beiden Formaten Audio-CDs erstellen, burncd(8) kann nur rohe PCM-Audiodaten verarbeiten. Der Dateikopf von WAV-Dateien erzeugt am Anfang des Stücks ein Knacken. Sie können den Dateikopf mit dem Werkzeug **SoX**, das sich als Paket oder aus dem Port `audio/sox` installieren lässt, entfernen:

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Lesen Sie Abschnitt 19.6 in diesem Handbuch, um mehr Informationen zur Benutzung von CD-Brennern mit FreeBSD zu erhalten.

8.4. Videos wiedergeben

Beigetragen von Ross Lippert.

Die Wiedergabe von Videos ist ein neues, sich schnell entwickelndes, Anwendungsgebiet. Seien Sie geduldig, es wird nicht alles so glatt laufen, wie bei den Audio-Anwendungen.

Bevor Sie beginnen, sollten Sie das Modell Ihrer Videokarte und den benutzten Chip kennen. Obwohl **Xorg** viele Videokarten unterstützt, können nur einige Karten Videos schnell genug wiedergeben. Eine Liste der Erweiterungen, die der X-Server für eine Videokarte unterstützt, erhalten Sie unter laufendem X11 mit dem Befehl xdpyinfo(1).

Halten Sie eine kurze MPEG-Datei bereit, mit der Sie Wiedergabeprogramme und deren Optionen testen können. Da einige DVD-Spieler in der Voreinstellung das DVD-Gerät mit `/dev/dvd` ansprechen oder diesen Namen fest einkodiert haben, wollen Sie vielleicht symbolische Links auf die richtigen Geräte anlegen:

```
# ln -sf /dev/acd0 /dev/dvd
# ln -sf /dev/acd0 /dev/rdvd
```

Wegen devfs(5) gehen gesondert angelegte Links wie diese bei einem Neustart des Systems verloren. Damit die symbolischen Links automatisch beim Neustart des Systems angelegt werden, fügen Sie die folgenden Zeilen in `/etc/devfs.conf` ein:

```
link acd0 dvd
link acd0 rdvd
```

Zum Entschlüsseln von DVDs müssen bestimmte DVD-ROM-Funktionen aufgerufen werden und schreibender Zugriff auf das DVD-Gerät erlaubt sein.

X11 benutzt Shared-Memory und Sie sollten die nachstehenden sysctl(8)-Variablen auf die gezeigten Werte erhöhen:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

8.4.1. Video-Schnittstellen

Es gibt einige Möglichkeiten, Videos unter X11 abzuspielen. Welche Möglichkeit funktioniert, hängt stark von der verwendeten Hardware ab. Ebenso hängt die erzielte Qualität von der Hardware ab. Die Videowiedergabe unter X11 ist ein aktuelles Thema, sodass jede neue Version von **Xorg** wahrscheinlich erhebliche Verbesserungen enthält.

Gebräuchliche Video-Schnittstellen sind:

1. X11: normale X11-Ausgabe über Shared-Memory.
2. XVideo: Eine Erweiterung der X11-Schnittstelle, die Videos in jedem X11-Drawable anzeigen kann.
3. SDL: Simple Directmedia Layer.
4. DGA: Direct Graphics Access.
5. SVGAlib: Eine Schnittstelle zur Grafikausgabe auf der Konsole.

8.4.1.1. XVideo

Die Erweiterung *XVideo* (auch Xvideo, Xv oder xv) von **Xorg** erlaubt die beschleunigte Wiedergabe von Videos in jedem Drawable. Diese Erweiterung liefert auch auf weniger leistungsfähigen Systemen (beispielsweise einem PIII 400 MHz Laptop) eine gute Wiedergabe.

Ob die Erweiterung läuft, entnehmen Sie der Ausgabe von `xvinfo`:

```
% xvinfo
```

XVideo wird untertsützt, wenn die Ausgabe wie folgt aussieht:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
```

```
    type: RGB (packed)
    depth: 0
    red, green, blue masks: 0x1f, 0x7e0, 0xf800
  id: 0x31313259 (Y211)
    guid: 59323131-0000-0010-8000-00aa00389b71
    bits per pixel: 6
    number of planes: 3
    type: YUV (packed)
  id: 0x0
    guid: 00000000-0000-0000-0000-000000000000
    bits per pixel: 0
    number of planes: 0
    type: RGB (packed)
    depth: 1
    red, green, blue masks: 0x0, 0x0, 0x0
```

Einige der aufgeführten Formate (wie YUV2 oder YUV12) existieren in machen XVideo-Implementierungen nicht. Dies kann zu Problemen mit einigen Spielern führen.

XVideo wird wahrscheinlich von Ihrer Karte nicht unterstützt, wenn die die Ausgabe wie folgt aussieht:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

Wenn die XVideo-Erweiterung auf Ihrer Karte nicht läuft, wird es nur etwas schwieriger, die Anforderungen für die Wiedergabe von Videos zu erfüllen. Abhängig von Ihrer Videokarte und Ihrem Prozessor können Sie dennoch zufriedenstellende Ergebnisse erzielen. Sie sollten vielleicht die weiterführenden Quellen in Abschnitt 8.4.3 zu Rate ziehen, um die Geschwindigkeit Ihres Systems zu steigern.

8.4.1.2. Simple Directmedia Layer

Die Simple Directmedia Layer, SDL, ist eine zwischen Microsoft Windows, BeOS und UNIX portable Schnittstelle. Mit dieser Schnittstelle können Anwendungen plattformunabhängig und effizient Ton und Grafik benutzen. SDL bietet eine hardwarenahe Schnittstelle, die manchmal schneller als die X11-Schnittstelle sein kann.

SDL finden Sie in den Ports im Verzeichnis `devel/sdl12`.

8.4.1.3. Direct Graphics Access

Die X11-Erweiterung Direct Graphics Access (DGA) erlaubt es Anwendungen, am X-Server vorbei direkt in den Framebuffer zu schreiben. Da die Anwendung und der X-Server auf gemeinsame Speicherbereiche zugreifen, müssen die Anwendungen unter dem Benutzer `root` laufen.

Die DGA-Erweiterung kann mit dga(1) getestet werden. Das Kommando `dga` wechselt, jedes Mal wenn eine Taste gedrückt wird, die Farben der Anzeige. Sie können das Programm mit der Taste **q** verlassen.

8.4.2. Video-Anwendungen

Dieser Abschnitt behandelt Anwendungen aus der FreeBSD-Ports-Sammlung, die Videos abspielen. An der Videowiedergabe wird derzeit aktiv gearbeitet, sodass der Funktionsumfang der Anwendungen von dem hier beschriebenen abweichen kann.

Viele unter FreeBSD laufende Videoanwendungen wurden unter Linux entwickelt und befinden sich noch im Beta-Status. Der Betrieb dieser Anwendungen unter FreeBSD stößt vielleicht auf einige der nachstehenden Probleme:

1. Eine Anwendung kann eine Datei einer anderen Anwendung nicht abspielen.
2. Eine Anwendung kann eine selbst produzierte Datei nicht abspielen.
3. Wenn dieselbe Anwendung auf unterschiedlichen Maschinen gebaut wird, wird ein Video unterschiedlich wiedergegeben.
4. Ein vergleichsweise einfacher Filter, wie die Skalierung eines Bildes, führt zu deutlichen Artefakten in der Darstellung.
5. Eine Anwendung stürzt häufig ab.
6. Die Dokumentation wird bei der Installation des Ports nicht installiert. Sie befindet sich entweder auf dem Internet oder im Verzeichnis `work` des Ports.

Viele Anwendungen sind zudem sehr “Linux-lastig”. Probleme entstehen durch die Implementierung von Standard-Bibliotheken in Linux-Distributionen oder dadurch, dass die Anwendung bestimmte Linux-Kernelfunktionen voraussetzt. Diese Probleme werden nicht immer vom Betreuer eines Ports bemerkt und umgangen. In der Praxis entstehen dadurch folgende Probleme:

1. Eigenschaften des Prozessors werden über `/proc/cpuinfo` ermittelt.
2. Die falsche Anwendung von Threads führt dazu, dass sich ein Programm aufhängt statt sich zu beenden.
3. Die Anwendung hängt von anderen Anwendungen ab, die sich noch nicht in der FreeBSD-Ports-Sammlung befinden.

Allerdings arbeiten die Anwendungsentwickler bislang mit den Betreuern der Ports zusammen, sodass zusätzlicher Portierungsaufwand minimiert wird.

8.4.2.1. MPlayer

MPlayer ist ein kürzlich entstandener und sich stark weiterentwickelnder Video-Spieler. Das Hauptaugenmerk des **MPlayer**-Teams liegt auf Geschwindigkeit und Flexibilität auf Linux und anderen UNIX Systemen. Das Projekt entstand weil der Gründer des Teams unzufrieden mit der Geschwindigkeit bestehender Video-Spieler war. Kritiker behaupten, dass die Benutzeroberfläche der einfachen Gestaltung zum Opfer fiel. Wenn Sie sich allerdings erstmal an die Kommandozeilenoptionen und die Tastensteuerung gewöhnt haben, funktioniert die Anwendung sehr gut.

8.4.2.1.1. MPlayer bauen

MPlayer finden Sie in der Ports-Sammlung unter `multimedia/mplayer`. Der Bau von **MPlayer** berücksichtigt die vorhandene Harware und erzeugt ein Programm, das nicht auf ein anderes System übertragbar ist. Es ist daher wichtig, dass Sie das Programm aus den Ports bauen und nicht das fertige Paket installieren. Zusätzlich können Sie auf der Kommandozeile von `make` noch einige Optionen angeben, die im `Makefile` beschrieben sind und am die Anfang des Baus ausgegeben werden:

```
# cd /usr/ports/multimedia/mplayer
# make
N - O - T - E

Take a careful look into the Makefile in order
to learn how to tune mplayer towards you personal preferences!
For example,
make WITH_GTK1
builds MPlayer with GTK1-GUI support.
If you want to use the GUI, you can either install
/usr/ports/multimedia/mplayer-skins
or download official skin collections from
http://www.mplayerhq.hu/homepage/dload.html
```

Für die meisten Benutzer sind die voreingestellten Option in Ordnung. Wenn Sie den XviD-Codec benötigen, müssen Sie auf der Kommandozeile die Option `WITH_XVID` angeben. Das DVD-Gerät können Sie mit der Option `WITH_DVD_DEVICE` angeben. Wenn Sie die Option nicht angeben, wird `/dev/acd0` benutzt.

Als dieser Abschnitt verfasst wurde, baute der **MPlayer**-Port die HTML-Dokumentation sowie die beiden Programme `mplayer` und `mencoder`. Mit `mencoder` können Sie Videodateien umwandeln.

Die HTML-Dokumentation von **MPlayer** ist sehr lehrreich. Wenn Sie in diesem Kapitel Informationen über Video-Hardware oder Schnittstellen vermissen, ist die **MPlayer**-Dokumentation eine ausgezeichnete Quelle. Wenn Sie Informationen über die Video-Unterstützung unter UNIX benötigen, sollten Sie die **MPlayer**-Dokumentation auf jeden Fall lesen.

8.4.2.1.2. MPlayer benutzen

Jeder Benutzer von **MPlayer** muss in seinem Heimatverzeichnis das Verzeichnis `.mplayer` anlegen. Dieses Verzeichnis können Sie wie folgt anlegen:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

Die Kommandozeilenoptionen von `mplayer` sind in der Hilfeseite aufgeführt. Eine genaue Beschreibung befindet sich in der HTML-Dokumentation. In diesem Abschnitt wird nur der normale Gebrauch von `mplayer` beschrieben.

Um eine Datei, wie *`testfile.avi`*, unter verschiedenen Video-Schnittstellen abzuspielen, benutzen Sie die Option `-vo`:

```
% mplayer -vo xv testfile.avi

% mplayer -vo sdl testfile.avi

% mplayer -vo x11 testfile.avi

# mplayer -vo dga testfile.avi

# mplayer -vo 'sdl:dga' testfile.avi
```

Es lohnt sich, alle Option zu testen. Die erzielte Geschwindigkeit hängt von vielen Faktoren ab und variiert beträchtlich je nach eingesetzter Hardware.

Wenn Sie eine DVD abspielen wollen, ersetzen Sie *testfile.avi* durch `-dvd://`*N Gerät*. *N* ist die Nummer des Stücks, das Sie abspielen wollen und *Gerät* gibt den Gerätenamen des DVD-ROMs an. Das nachstehende Kommando spielt das dritte Stück von `/dev/dvd`:

```
# mplayer -vo dga -dvd://3 /dev/dvd
```

> **Anmerkung:** Das standardmäßig verwendete DVD-Laufwerk kann beim Bau des **MPlayer**-Ports mit der Option `WITH_DVD_DEVICE` festgelegt werden. Die Voreinstellung verwendet das Gerät `/dev/acd0`. Genaueres finden Sie im `Makefile` des Ports.

Die Tastenkombinationen zum Abbrechen, Anhalten und Weiterführen der Wiedergabe entnehmen Sie bitte der Ausgabe von `mplayer -h` oder der Hilfeseite.

Weitere nützliche Optionen für die Wiedergabe sind `-fs -zoom` zur Wiedergabe im Vollbild-Modus und `-framedrop` zur Steigerung der Geschwindigkeit.

Damit die Kommandozeile von `mplayer` kurz bleibt, kann ein Benutzer Vorgaben in der Datei `.mplayer/config` hinterlegen:

```
vo=xv
fs=yes
zoom=yes
```

Schließlich kann `mplayer` noch DVD-Stücke in `.vob`-Dateien rippen. Das zweite Stück einer DVD wandeln Sie wie folgt in eine Datei um:

```
# mplayer -dumpstream -dumpfile out.vob -dvd://2 /dev/dvd
```

Die Ausgabedatei `out.vob` wird im MPEG-Format abgespeichert und kann mit anderen Werkzeugen aus diesem Abschnitt bearbeitet werden.

8.4.2.1.3. mencoder

Sie sollten die HTML-Dokumentation lesen, bevor Sie `mencoder` benutzen. Es gibt zwar eine Hilfeseite, die aber ohne die HTML-Dokumentation nur eingeschräkt nützlich ist. Es gibt viele Möglichkeiten die Qualität zu verbessern, die Bitrate zu verringern und Formate zu konvertieren. Einige davon haben erhebliche Auswirkungen auf die Geschwindigkeit der Wiedergabe. Zum Start finden Sie im Folgenden einige Kommandozeilen. Die erste kopiert einfach eine Datei:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Falsche Kombinationen von Kommandozeilenparametern ergeben eventuell Dateien, die selbst `mplayer` nicht mehr abspielen kann. Wenn Sie in eine Datei rippen, sollten Sie daher auf jeden Fall die Option `-dumpfile` von `mplayer` verwenden.

Die nachstehende Kommandozeile wandelt die Datei *input.avi* nach MPEG4 mit MPEG3 für den Ton um (hierfür wird der Ports `audio/lame` benötigt):

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
  -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Die Ausgabedatei lässt sowohl mit `mplayer` als auch `xine` abspielen.

Wenn Sie *`input.avi`* durch `-dvd://1 /dev/dvd` ersetzen und das Kommando unter `root` laufen lassen, können Sie ein DVD-Stück direkt konvertieren. Da Sie wahrscheinlich beim ersten Mal unzufrieden mit den Ergebnissen sind, sollten Sie das Stück zuerst in eine Datei schreiben und anschließend die Datei weiterverarbeiten.

8.4.2.2. Der Video-Spieler xine

Der Video-Spieler **xine** ist ein Projekt mit großem Umfang. Das Projekt will nicht nur ein Programm für alle Video-Anwendungen bieten, sondern auch eine wiederverwendbare Bibliothek und ein Programm, das durch Plugins erweiterbar ist. Das Programm steht als fertiges Paket oder als Port unter `multimedia/xine` zur Verfügung.

Der `multimedia/xine`-Spieler hat noch ein paar Ecken und Kanten, macht aber insgesamt einen guten Eindruck. Für einen reibungslosen Betrieb benötigt **xine** entweder eine schnelle CPU oder die XVideo-Erweiterung. Das GUI ist etwas schwerfällig.

Zurzeit gibt es kein **xine**-Modul, das CSS-kodierte DVDs abspielen kann und sich in der FreeBSD Ports-Sammlung befindet.

xine ist benutzerfreundlicher als **MPlayer**, bietet allerdings nicht soviele Möglichkeiten. Am schnellsten läuft **xine** mit der XVideo-Erweiterung.

In der Voreinstellung startet **xine** eine grafische Benutzeroberfläche. Über Menüs können Sie Dateien öffnen:

```
% xine
```

Alternativ können Sie das Programm auch ohne GUI aufrufen und Dateien direkt abspielen:

```
% xine -g -p mymovie.avi
```

8.4.2.3. Die transcode-Werkzeuge

transcode ist kein Spieler, sondern eine Sammlung von Werkzeugen zur Umwandlung von Video- und Sounddateien. **transcode** mischt Video-Dateien und kann kaputte Video-Dateien reparieren. Die Werkzeuge werden als Filter verwendet, das heißt die Ein- und Ausgaben verwenden `stdin/stdout`.

Beim Bau von **transcode** über den Port `multimedia/transcode` können zwar zahreiche Optionen angegeben werden. Empfehlenswert ist es aber, den Bau mit folgendem Befehl zu starten:

```
# make WITH_OPTIMIZED_CFLAGS=yes WITH_LIBA52=yes WITH_LAME=yes WITH_OGG=yes \
WITH_MJPEG=yes -DWITH_XVID=yes
```

Diese Einstellungen sollen für die meisten Anwender ausreichend sein.

Um die Fähigkeiten von `transcode` zu illustrieren, wird im folgenden Beispiel eine DivX-Datei in eine PAL MPEG-1-Datei konvertiert:

```
% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa
```

Die daraus resultierende MPEG-Datei, *`output_vcd.mpg`*, kann beispielsweise mit **MPlayer** abgespielt werden. Sie können sie sogar als Video-CD auf eine CD-R brennen. Wenn Sie diese Funktion benötigen, müssen Sie zusätzlich die beiden Programme `multimedia/vcdimager` und `sysutils/cdrdao` installieren.

Zwar gibt es eine Manualpage zu `transcode`, Sie sollen aber auf jeden Fall auch die Informationen und Beispiele im transcode-Wiki (http://www.transcoding.org/cgi-bin/transcode) lesen.

8.4.3. Weiterführende Quellen

Die Video-Software für FreeBSD entwickelt sich sehr schnell. Es ist wahrscheinlich, dass die hier angesprochenen Probleme bald gelöst sind. Bis dahin müssen Anwender, die das meiste aus den Audio- und Video-Fähigkeiten von FreeBSD machen wollen, Informationen aus mehreren FAQs und Tutorien zusammensuchen und verschiedene Anwendungen nebeneinander betreiben. Dieser Abschnitt weist auf weitere Informationsquellen hin.

Die MPlayer-Dokumentation (http://www.mplayerhq.hu/DOCS/) ist sehr aufschlussreich. Die Dokumente sollten wahrscheinlich von jedem gelesen werden, der hohe Fachkenntnisse über Video auf UNIX Systemen erlangen will. Die **MPlayer**-Mailinglisten reagiert feindselig auf Personen, die es nicht für nötig halten, die Dokumentation zu lesen. Wenn Sie Fehlerberichte an die Liste schicken wollen, lesen Sie bitte vorher die ausgezeichnete Dokumentation (RTFM).

Das xine HOWTO (http://dvd.sourceforge.net/xine-howto/en_GB/html/howto.html) enthält allgemein gültige Hinweise zur Verbesserung der Wiedergabegeschwindigkeit.

Schließlich gibt es noch weitere vielversprechende Anwendungen, die Sie vielleicht ausprobieren wollen:

- Avifile (http://avifile.sourceforge.net/) gibt es schon als Port `multimedia/avifile`.
- Ogle (http://www.dtek.chalmers.se/groups/dvd/) wurde ebenfalls schon portiert: `multimedia/ogle`.
- Xtheater (http://xtheater.sourceforge.net/).
- `multimedia/dvdauthor`, ist ein Open-Source-Paket, mit dem Sie DVDs erstellen können.

8.5. TV-Karten einrichten

Beigetragen von Josef El-Rayes. Überarbeitet von Marc Fonvieille.

8.5.1. Einführung

Mit TV-Karten können Sie mit Ihrem Rechner über Kabel oder Antenne fernsehen. Die meisten Karten besitzen einen RCA- oder S-Video-Eingang. Einige Karten haben auch einen FM-Radio-Empfänger.

Der bktr(4)-Treiber von FreeBSD unterstützt PCI-TV-Karten mit einem Brooktree Bt848/849/878/879 oder einem Conexant CN-878/Fusion 878a Chip. Die Karte sollte einen der unterstützten Empfänger besitzen, die in der Hilfeseite bktr(4) aufgeführt sind.

8.5.2. Den Treiber einrichten

Um Ihre Karte zu benutzen, müssen Sie den bktr(4)-Treiber laden. Fügen Sie die nachstehende Zeile in die Datei `/boot/loader.conf` ein:

```
bktr_load="YES"
```

Sie können den Treiber für die TV-Karte auch fest in den Kernel compilieren. Erweitern Sie dazu Ihre Kernelkonfiguration um die folgenden Zeilen:

```
device   bktr
device iicbus
device iicbb
device smbus
```

Die zusätzlichen Treiber werden benötigt, da die Komponenten der Karte über einen I2C-Bus verbunden sind. Bauen und installieren Sie dann den neuen Kernel.

Anschließend müssen Sie Ihr System neu starten. Während des Neustarts sollte Ihre TV-Karte erkannt werden:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

Abhängig von Ihrer Hardware können die Meldungen natürlich anders aussehen. Sie sollten aber prüfen, dass der Empfänger richtig erkannt wird. Die entdeckten Geräte lassen sich mit sysctl(8) oder in der Kernelkonfigurationsdatei überschreiben. Wenn Sie beispielsweise einen Philips-SECAM-Empfänger erzwingen wollen, fügen Sie die folgende Zeile zur Kernelkonfigurationsdatei hinzu:

```
options OVERRIDE_TUNER=6
```

Alternativ können Sie direkt sysctl(8) benutzen:

```
# sysctl hw.bt848.tuner=6
```

Weitere Informationen zu den verschiedenen Optionen finden Sie in bktr(4) sowie in der Datei `/usr/src/sys/conf/NOTES`.

8.5.3. Nützliche Anwendungen

Um die TV-Karte zu benutzen, müssen Sie eine der nachstehenden Anwendungen installieren:

- `multimedia/fxtv` lässt das Fernsehprogramm in einem Fenster laufen und kann Bilder, Audio und Video aufzeichnen.
- `multimedia/xawtv` eine weitere TV-Anwendung, mit den gleichen Funktionen wie **fxtv**.
- `misc/alevt` dekodiert und zeigt Videotext/Teletext an.
- Mit `audio/xmradio` lässt sich der FM-Radio-Empfänger, der sich auf einigen TV-Karten befindet, benutzen.

- `audio/wmtune` ein leicht zu bedienender Radio-Empfänger.

Weitere Anwendungen finden Sie in der FreeBSD Ports-Sammlung.

8.5.4. Fehlersuche

Wenn Sie Probleme mit Ihrer TV-Karte haben, prüfen Sie zuerst, ob der Video-Capture-Chip und der Empfänger auch wirklich vom bktr(4)-Treiber unterstützt werden. Prüfen Sie dann, ob Sie die richtigen Optionen verwenden. Weitere Hilfe erhalten Sie auf der Mailingliste freebsd-multimedia (http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia) und in deren Archiven.

8.6. MythTV

MythTV ist ein Open Source PVR-Softwareprojekt.

Es ist in der Linux-Welt als komplexe Anwendung mit vielen Abhängigkeiten bekannt und deshalb schwierig zu installieren. Das FreeBSD Portssystem vereinfacht diesen Prozess sehr stark, jedoch müssen manche Komponenten manuell eingerichtet werden. Dieser Abschnitt soll dazu dienen, bei der Einrichtung von MythTV zu helfen.

8.6.1. Hardware

MythTV wurde entwickelt, um V4L zu verwenden, so dass auf Videoeingabegeräte wie Kodierer und Empfänger zugegriffen werden kann. Aktuell funktioniert MythTV am besten mit USB DVB-S/C/T Karten, die von `multimedia/webcamd` unterstützt werden, weil **webcamd** eine V4L-Anwendung zur Verfügung stellt, die als Benutzerprogramm läft. Jede DVB-Karte, welche von **webcamd** unterstützt wird, sollte mit MythTV funktionieren, jedoch gibt es eine Liste von Karten, die hier (http://wiki.freebsd.org/WebcamCompat) abgerufen werden kann. Es existieren auch Treiber für Hauppauge-Karten in den folgenden Paketen: `multimedia/pvr250` und `multimedia/pvrxxx`, allerdings liefern diese nur eine Treiberschnittstelle, die nicht dem Standard entspricht und die nicht mit MythTV-Versionen grösser als 0.23 funktionieren.

HTPC (http://wiki.freebsd.org/HTPC) enthält eine Liste von allen verfügbaren DVB-Treibern.

8.6.2. Abhängigkeiten

Da MythTV flexibel und modular aufgebaut ist, ist der Benutzer in der Lage, das Frontend und Backend auf unterschiedlichen Rechnern laufen zu lassen.

Für das Frontend wird `multimedia/mythtv-frontend`, sowie ein X-Server benötigt, welcher in `x11/xorg` zu finden ist. Idealerweise besitzt der Frontend-Computer auch eine Videokarte, die XvMC unterstützt, sowie optional eine LIRC-kompatible Fernbedienung.

Für das Backend wird `multimedia/mythtv` benötigt, ebenso wie eine MySQL™-Datenbank, sowie zusätzlich einen Empfänger und Speicherplatz für Aufzeichnungen. Das MySQL-Paket sollte automatisch als Abhängigkeit mitinstalliert werden, wenn `multimedia/mythtv` gebaut wird.

8.6.3. MythTV einrichten

Um MythTV zu installieren, befolgen Sie die hier aufgeführten Schritte. Zuerst installieren Sie MythTV aus der Ports-Sammlung:

```
# cd /usr/ports/multimedia/mythtv
# make install
```

Richten Sie anschliessend die MythTV-Datenbank ein:

```
# mysql -uroot -p < /usr/local/share/mythtv/database/mc.sql
```

Konfigurieren Sie dann das Backend:

```
# mythtv-setup
```

Zum Schluss starten Sie das Backend:

```
# echo 'mythbackend_enable="YES"' >> /etc/rc.conf
# /usr/local/etc/rc.d/mythbackend start
```

8.7. Scanner

Beigetragen von Marc Fonvieille.

8.7.1. Einführung

Unter FreeBSD stellt **SANE** (Scanner Access Now Easy) aus der Ports-Sammlung eine einheitliche Schnittstelle (API) für den Zugriff auf Scanner bereit. **SANE** wiederum greift auf Scanner mithilfe einiger FreeBSD-Treiber zu.

FreeBSD unterstützt sowohl SCSI- als auch USB-Scanner. Prüfen Sie vor der Konfiguration mithilfe der Liste der unterstützten Geräte (http://www.sane-project.org/sane-supported-devices.html) ob Ihr Scanner von **SANE** unterstützt wird. Bei Systemen vor FreeBSD 8.X zählt die Hilfeseite uscanner(4) ebenfalls die unterstützten USB-Scanner auf.

8.7.2. Den Kernel für Scanner einrichten

Da sowohl SCSI- als auch USB-Scanner unterstützt werden, werden abhängig von der Schnittstelle unterschiedliche Treiber benötigt.

8.7.2.1. USB-Scanner

Im `GENERIC`-Kernel sind schon alle, für einen USB-Scanner notwendigen, Treiber enthalten. Wenn Sie einen angepassten Kernel benutzen, prüfen Sie, dass die Kernelkonfiguration die nachstehenden Zeilen enthält:

```
device usb
device uhci
device ohci
device ehci
```

Bei Systemen vor FreeBSD 8.X wird ausserdem noch die folgende Zeile benötigt:

```
device uscanner
```

Bei diesen FreeBSD-Versionen liefert das uscanner(4)-Gerät die Unterstützung für USB-Scanner. Seit FreeBSD 8.0 ist diese Unterstützung direkt in der libusb(3)-Bibliothek enthalten.

Nachdem Sie das System mit dem richtigen Kernel neu gestartet haben, stecken Sie den USB-Scanner ein. Danach sollte in den Systemmeldungen (die Sie mit dmesg(8) betrachten können) eine Zeile ähnlich der folgenden erscheinen:

```
ugen0.2: <EPSON> at usbus0
```

bzw. auf einem FreeBSD 7.X System:

```
uscanner0: EPSON EPSON Scanner, rev 1.10/3.02, addr 2
```

Diese Meldung besagt, dass der Scanner entweder die Gerätedatei `/dev/ugen0.2` oder `/dev/uscanner0` benutzt, je nachdem, welche FreeBSD-Version eingesetzt wird. In diesem Beispiel wurde ein EPSON Perfection® 1650 USB-Scanner verwendet.

8.7.2.2. SCSI-Scanner

Wenn Ihr Scanner eine SCSI-Schnittstelle besitzt, ist die Kernelkonfiguration abhängig vom verwendeten SCSI-Controller. Der `GENERIC`-Kernel unterstützt die gebräuchlichen SCSI-Controller. Den richtigen Treiber finden Sie in der Datei `NOTES`. Neben dem Treiber muss Ihre Kernelkonfiguration noch die nachstehenden Zeilen enthalten:

```
device scbus
device pass
```

Nachdem Sie einen Kernel gebaut und installiert haben, sollte der Scanner beim Neustart in den Systemmeldungen erscheinen:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Wenn der Scanner während des Systemstarts ausgeschaltet war, können Sie die Geräteerkennung erzwingen, indem Sie den SCSI-Bus erneut absuchen. Verwenden Sie dazu das Kommando camcontrol(8):

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

Der Scanner wird anschließend in der SCSI-Geräteliste angezeigt:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

Weiteres über SCSI-Geräte lesen Sie bitte in den Hilfeseiten scsi(4) und camcontrol(8) nach.

8.7.3. SANE konfigurieren

SANE besteht aus zwei Teilen, den Backends (`graphics/sane-backends`) und den Frontends (`graphics/sane-frontends`). Das Backend greift auf den Scanner zu. Welches Backend welchen Scanner unterstützt, entnehmen Sie der Liste der unterstützten Geräte (http://www.sane-project.org/sane-supported-devices.html). Der Betrieb eines Scanners ist nur mit dem richtigen Backend möglich. Die Frontends sind die Anwendungen, mit denen gescannt wird (**xscanimage**).

Installieren Sie zuerst den Port oder das Paket `graphics/sane-backends`. Anschließend können Sie mit dem Befehl `sane-find-scanner` prüfen, ob **SANE** Ihren Scanner erkennt:

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

Die Ausgabe zeigt die Schnittstelle und die verwendete Gerätedatei des Scanners. Der Hersteller und das Modell können in der Ausgabe fehlen.

> **Anmerkung:** Bei einigen USB-Scannern müssen Sie die Firmware aktualisieren, dies wird in der Hilfeseite des Backends erklärt. Lesen Sie bitte auch die Hilfeseiten sane-find-scanner(1) und sane(7).

Als nächstes müssen Sie prüfen, ob der Scanner vom Frontend erkannt wird. Die **SANE**-Backends werden mit dem Kommandozeilenwerkzeug scanimage(1) geliefert. Mit diesem Werkzeug können Sie sich Scanner anzeigen lassen und den Scan-Prozess von der Kommandozeile starten. Die Option `-L` zeigt die Scanner an:

```
# scanimage -L
device ‘snapscan:/dev/pass3’ is a AGFA SNAPSCAN 600 flatbed scanner
```

Oder, für das Beispiel mit dem USB-Scanner in Abschnitt 8.7.2.1:

```
# scanimage -L
device ’epson2:libusb:/dev/usb:/dev/ugen0.2’ is a Epson GT-8200 flatbed scanner
```

Diese Ausgabe stammt von einem FreeBSD 8.X System, die Zeile `’epson2:libusb:/dev/usb:/dev/ugen0.2’` nennt das Backend (`epson2`) und die Gerätedatei (`/dev/ugen0.2`), die der Scanner verwendet.

> **Anmerkung:** Erscheint die Meldung, dass kein Scanner gefunden wurde oder wird gar keine Ausgabe erzeugt, konnte scanimage(1) keinen Scanner erkennen. In diesem Fall müssen Sie in der Konfigurationsdatei des Backends das zu benutzende Gerät eintragen. Die Konfigurationsdateien der Backends befinden sich im Verzeichnis `/usr/local/etc/sane.d/`. Erkennungsprobleme treten bei bestimmten USB-Scannern auf.
>
> Mit dem USB-Scanner aus Abschnitt 8.7.2.1 zeigt `sane-find-scanner` unter FreeBSD 8.X die folgende Ausgabe:
>
> ```
> # sane-find-scanner -q
> found USB scanner (UNKNOWN vendor and product) at device /dev/uscanner0
> ```
>
> Der Scanner wurde richtig erkennt, er benutzt eine USB-Schnittstelle und verwendet die Gerätedatei `/dev/uscanner0`. Ob der Scanner vom Frontend erkannt wird, zeigt das nachstehende Kommando:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Da der Scanner nicht erkannt wurde, muss die Datei `/usr/local/etc/sane.d/epson2.conf` editiert werden. Der verwendete Scanner war ein EPSON Perfection 1650, daher wird das `epson2`-Backend benutzt. Lesen Sie bitte alle Kommentare in der Konfigurationsdatei des Backends. Die durchzuführenden Änderungen sind einfach. Kommentieren Sie zunächst alle Zeilen mit der falschen Schnittstelle aus. Da der Scanner eine USB-Schnittstelle besitzt, wurden im Beispiel alle Zeilen, die mit `scsi` anfingen, auskommentiert. Fügen Sie dann die Schnittstelle und den Gerätenamen am Ende der Datei ein. In diesem Beispiel wurde die nachstehende Zeile eingefügt:

```
usb /dev/uscanner0
```

Weitere Hinweise entnehmen Sie bitte der Hilfeseite des Backends. Jetzt können Sie prüfen, ob der Scanner richtig erkannt wird:

```
# scanimage -L
device ‘epson:/dev/uscanner0’ is a Epson GT-8200 flatbed scanner
```

Der Scanner wurde nun erkannt. Es ist nicht wichtig, ob der Hersteller oder das Modell Ihres Scanners korrekt angezeigt werden. Wichtig ist nur die Ausgabe `‘epson:/dev/uscanner0’`, die das richtige Backend und den richtigen Gerätenamen anzeigt.

Wenn `scanimage -L` den Scanner erkannt hat, ist der Scanner eingerichtet und bereit, zu scannen.

Obwohl wir mit scanimage(1) von der Kommandozeile scannen können, ist eine graphische Anwendung zum Scannen besser geeignet. **SANE** bietet ein einfaches und effizientes Werkzeug: **xscanimage** (`graphics/sane-frontends`).

Xsane (`graphics/xsane`) ist eine weitere beliebte graphische Anwendung. Dieses Frontend besitzt erweiterte Funktionen wie den Scan-Modus (beispielsweise Photo, Fax), eine Farbkorrektur und Batch-Scans. Beide Anwendungen lassen sich als **GIMP**-Plugin verwenden.

8.7.4. Den Scanner für Benutzerkonten freigeben

Zuvor wurden alle Tätigkeiten mit `root`-Rechten ausgeführt. Wenn andere Benutzer den Scanner benutzen sollen, müssen sie Lese- und Schreibrechte auf die Gerätedatei des Scanners besitzen. Im Beispiel wird die Datei `/dev/ugen0.2` verwendet, die faktisch nur ein Symlink auf die echte Gerätedatei, `/dev/usb/0.2.0` genannt, darstellt (ein kurzer Blick auf das `/dev`-Verzeichnis bestätigt dies). Sowohl der Symlink als auch die Gerätedatei sind jeweils im Besitz der Gruppen `wheel` und `operator`. Damit der Benutzer *`joe`* auf den Scanner zugreifen kann, muss das Konto in die Gruppe `operator` aufgenommen werden. Allerdings sollten Sie, aus Sicherheitsgründen, genau überlegen, welche Benutzer Sie zu welcher Gruppe hinzufügen, besonders bei der Gruppe `wheel`. Eine bessere Lösung ist es, eine spezielle Gruppe für den Zugriff auf USB-Geräte anzulegen und den Scanner für Mitglieder dieser Gruppe zugänglich zu machen.

Beispielsweise kann man eine *`usb`*-Gruppe verwenden. Der erste Schritt dazu ist das erstellen der Gruppe mit Hilfe des pw(8)-Kommandos:

```
# pw groupadd usb
```

Anschliessend muss der `/dev/ugen0.2`-Symlink und der Gerätename `/dev/usb/0.2.0` für die `usb`-Gruppe mit den richtigen Schreibrechten (`0660` oder `0664`) ausgestattet werden, denn standardmässig kann nur der Besitzer dieser Dateien (`root`) darauf schreiben. All dies kann durch das Hinzufügen der folgenden Zeile in die `/etc/devfs.rules`-Datei erreicht werden:

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0660 group usb
```

FreeBSD 7.X-Anwender benötigen unter Umständen die folgenden Zeilen mit der korrekten Gerätedatei `/dev/uscanner0`:

```
[system=5]
add path uscanner0 mode 660 group usb
```

In die Datei `/etc/rc.conf` fügen Sie noch die folgende Zeile ein:

```
devfs_system_ruleset="system"
```

Starten Sie anschließend Ihr System neu.

Weitere Informationen finden Sie in devfs(8).

Jetzt braucht man nur noch Benutzer der Gruppe *`usb`* hinzufügen, um ihnen Zugriff auf den Scanner zu erlauben:

```
#pw groupmod usb -m joe
```

Weitere Details können Sie in der pw(8)-Manualpage nachlesen.

Kapitel 9. Konfiguration des FreeBSD-Kernels

Erweitert und neu strukturiert von Jim Mock. Ursprünglich veröffentlicht von Jake Hamby. Übersetzt von Robert Altschaffel.

9.1. Übersicht

Der Kernel ist das Herz des FreeBSD Betriebssystems. Er ist verantwortlich für die Speicherverwaltung, das Durchsetzen von Sicherheitsdirektiven, Netzwerkfähigkeit, Festplattenzugriffen und vieles mehr. Obwohl FreeBSD es immer mehr ermöglicht, dynamisch konfiguriert zu werden, ist es ab und an notwendig, den Kernel neu zu konfigurieren und zu kompilieren.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Wieso Sie Ihren Kernel neu konfigurieren sollten.
- Wie Sie eine Kernelkonfigurationsdatei erstellen oder verändern.
- Wie Sie mit der Konfigurationsdatei einen neuen Kernel kompilieren.
- Wie Sie den neuen Kernel installieren.
- Was zu tun ist, falls etwas schiefgeht.

Alle Kommandos, aus den Beispielen dieses Kapitels, müssen mit `root`-Rechten ausgeführt werden.

9.2. Wieso einen eigenen Kernel bauen?

Traditionell besaß FreeBSD einen monolithischen Kernel. Das bedeutet, dass der Kernel ein einziges großes Programm war, das eine bestimmte Auswahl an Hardware unterstützte. Also musste man immer, wenn man das Kernelverhalten verändern wollte, zum Beispiel wenn man neue Hardware hinzufügen wollte, einen neuen Kernel kompilieren, installieren und das System neu starten.

Heutzutage vertritt FreeBSD immer mehr die Idee eines modularen Kernels, bei dem bestimmte Funktionen, je nach Bedarf, als Module geladen werden können. Ein bekanntes Beispiel dafür sind die Module für die PCMCIA-Karten in Laptops, die zum Starten nicht zwingend benötigt und erst bei Bedarf geladen werden.

Trotzdem ist es noch immer nötig, einige statische Kernelkonfigurationen durchzuführen. In einigen Fällen ist die Funktion zu systemnah, um durch ein Modul zu realisiert werden. In anderen Fällen hat eventuell noch niemand ein ladbares Kernelmodul für diese Funktion geschrieben.

Das Erstellen eines angepaßten Kernels ist eines der wichtigsten Rituale für erfahrene BSD-Benutzer. Obwohl dieser Prozess recht viel Zeit in Anspruch nimmt, bringt er doch viele Vorteile für Ihr FreeBSD System. Der `GENERIC`-Kernel muss eine Vielzahl unterschiedlicher Hardware unterstützen, im Gegensatz dazu unterstützt ein angepasster Kernel nur *Ihre* Hardware. Dies hat einige Vorteile:

- Schnellerer Bootvorgang. Da der Kernel nur nach der Hardware des Systems sucht, kann sich die Zeit für einen Systemstart erheblich verkürzen.
- Geringerer Speicherbedarf. Ein eigener Kernel benötigt in der Regel weniger Speicher als ein `GENERIC`-Kernel durch das Entfernen von Funktionen und Gerätetreibern. Das ist vorteilhaft, denn der Kernel verweilt immer im

RAM und verhindert dadurch, dass dieser Speicher von Anwendungen genutzt wird. Insbesondere profitieren Systeme mit wenig RAM davon.

- Zusätzliche Hardwareunterstützung. Ein angepasster Kernel kann Unterstützung für Geräte wie Soundkarten bieten, die im `GENERIC`-Kernel nicht enthalten sind.

9.3. Informationen über die vorhandene Hardware beschaffen

Geschrieben von Tom Rhodes.

Bevor Sie mit der Kernelkonfiguration beginnen, sollten Sie wissen, über welche Hardware Ihr System verfügt. Verwenden Sie derzeit noch ein anderes Betriebssystem, ist es meist sehr einfach, eine Liste der installierten Hardware zu erzeugen. Verwenden Sie beispielsweise Microsoft Windows, können Sie dafür den **Gerätemanager** verwenden, den Sie in der "Systemsteuerung" finden.

> **Anmerkung:** Einige Versionen von Microsoft Windows verfügen über ein **System**-Icon auf dem Desktop, über das Sie den **Gerätemanager** direkt aufrufen können.

Haben Sie außer FreeBSD kein weiteres Betriebssystem, müssen Sie diese Informationen manuell zusammentragen. Eine Möglichkeit, an Informationen über die vorhandene Hardware zu gelangen, ist der Einsatz von dmesg(8) in Kombination mit man(1). Die meisten FreeBSD-Gerätetreiber haben eine eigene Manualpage, die Informationen über die unterstützte Hardware enthält. Während des Systemstarts werden Informationen über die vorhandene Hardware ausgegeben. Die folgenden Zeilen zeigen beispielsweise an, dass der `psm`-Treiber eine angeschlossene Maus gefunden hat:

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

Dieser Treiber muss in Ihrer Kernelkonfigurationsdatei vorhanden sein oder durch das Werkzeug loader.conf(5) geladen werden.

Manchmal zeigt `dmesg` während des Systemstarts nur Systemmeldungen, aber keine Informationen zur gefundenen Hardware an. In diesem Fall können Sie diese Informationen durch das Studium der Datei `/var/run/dmesg.boot` herausfinden.

Eine weitere Möglichkeit bietet das Werkzeug pciconf(8), das ausführliche Informationen bereitstellt. Dazu ein Beispiel:

```
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01 hdr=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

Diese Zeilen, die Sie durch den Aufruf des Befehls `pciconf -lv` erhalten, zeigen, dass der Treiber `ath` eine drahtlose Ethernetkarte gefunden hat. Durch Eingabe des Befehls `man ath` öffnet sich die Manualpage ath(4).

Rufen Sie man(1) mit der Option `-k` auf, können Sie die Datenbank der Manualpages auch durchsuchen. Für das angegebene Beispiel würde dieser Befehl beispielsweise so aussehen:

```
# man -k Atheros
```

Dadurch erhalten Sie eine Liste aller Manualpages, die das angegebene Suchkriterium enthalten:

```
ath(4)                  - Atheros IEEE 802.11 wireless network driver
ath_hal(4)              - Atheros Hardware Access Layer (HAL)
```

Mit diesen Informationen ausgestattet, sollte der Bau eines angepassten Kernel keine allzugroßen Probleme mehr bereiten.

9.4. Kerneltreiber, Subsysteme und Module

Bevor Sie einen angepassten Kernel erstellen, überlegen Sie sich bitte, warum Sie dies tun wollen. Wenn Sie lediglich eine bestimmte Hardwareunterstützung benötigen, existiert diese vielleicht schon als Kernelmodul.

Kernelmodule existieren im Verzeichnis `/boot/kernel` und können dynamisch in den laufenden Kernel über kldload(8) geladen werden. Die meisten, wenn nicht sogar alle, Kerneltreiber besitzen ein spezifisches Modul und eine Manualpage. Beispielsweise erwähnte der letzte Abschnitt den drahtlosen Ethernettreiber `ath`. Dieses Gerät hat die folgende Information in seiner Manualpage:

```
Alternatively, to load the driver as a module at boot time, place the
following line in loader.conf(5):

    if_ath_load="YES"
```

Wie dort angegeben, wird das Modul durch die Zeile `if_ath_load="YES"` in der Datei `/boot/loader.conf` dynamisch beim Systemstart geladen.

Allerdings gibt es in manchen Fällen kein dazugehöriges Modul. Das gilt insbesondere für bestimmte Teilsysteme und sehr wichtige Treiber. Beispielsweise ist das Fast File System (FFS) eine notwendige Kerneloption, genauso wie die Netzwerkunterstützung (INET). Die einzige Möglichkeit, herauszufinden, ob ein Treiber benötigt ist, ist die Überprüfung des jeweiligen Moduls.

> **Warnung:** Es ist erstaunlich einfach, einen defekten Kernel zu erhalten (beispielsweise durch das Entfernen der eingebauten Unterstützung für ein Gerät oder einer Kerneloption). Wenn beispielsweise der ata(4)-Treiber aus der Kernelkonfigurationsdatei entfernt wird, kann ein System, das den ATA-Festplattentreiber benötigt, nicht mehr starten, ohne dass Sie das entsprechende Kernelmodul durch einen Eintrag in `loader.conf` aufnehmen. Wenn Sie nicht sicher sind, wie Sie vorgehen sollen, überprüfen Sie zuerst das Modul. Im Zweifelsfall belassen Sie die Unterstützung für ein bestimmtes Gerät besser im Kernel.

9.5. Erstellen und Installation eines angepassten Kernels

> **Anmerkung:** Sie benötigen den kompletten Quellcodebaum, um den Kernel zu bauen.

Zuerst erläutern wir die Verzeichnisstruktur, in der der Kernel gebaut wird. Die im Folgenden genannten Verzeichnisse sind relativ zum Verzeichnis `/usr/src/sys` angegeben, das Sie auch über den Pfad `/sys` erreichen können. Es existieren mehrere Unterverzeichnisse, die bestimmte Teile des Kernels darstellen, aber die für uns wichtigsten sind *arch*`/conf`, in dem Sie die Konfigurationsdatei für den angepassten Kernel erstellen werden, und `compile`, in dem der Kernel gebaut wird. *arch* kann entweder `i386`, `amd64`, `ia64`, `powerpc`, `sparc64` oder `pc98` (eine in Japan beliebte Architektur) sein. Alles in diesen Verzeichnissen ist nur für die jeweilige Architektur relevant. Der Rest des Codes ist maschinenunabhängig und für alle Plattformen, auf die FreeBSD portiert werden kann, gleich. Beachten Sie die Verzeichnisstruktur, die jedem unterstützten Gerät, jedem Dateisystem und jeder Option ein eigenes Verzeichnis zuordnet.

Die Beispiele in diesem Kapitel verwenden ein i386-System. Wenn Sie ein anderes System benutzen, passen Sie bitte die Pfade entsprechend der Architektur des Systems an.

> **Anmerkung:** Falls Sie kein `/usr/src/`-Verzeichnis vorfinden (oder dieses leer ist), so sind die Quellen nicht installiert. Der einfachste Weg, dies nachzuholen, ist `sysinstall` als `root` auszuführen. Dort wählen Sie Configure, dann Distributions, dann src, und schließlich All. Falls nicht vorhanden, sollten Sie auch noch einen symbolischen Link auf `/usr/src/sys/` anlegen:
>
> ```
> # ln -s /usr/src/sys /sys
> ```

Als nächstes wechseln sie in das Verzeichnis *arch*`/conf` und kopieren die Konfigurationsdatei `GENERIC` in eine Datei, die den Namen Ihres Kernels trägt. Zum Beispiel:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYKERNEL
```

Traditionell ist der Name des Kernels immer in Großbuchstaben. Wenn Sie mehrere FreeBSD mit unterschiedlicher Hardware warten, ist es nützlich, wenn Sie Konfigurationsdatei nach dem Hostnamen der Maschinen benennen. Im Beispiel verwenden wir den Namen *MYKERNEL*.

> **Tipp:** Es ist nicht zu empfehlen die Konfigurationsdatei direkt unterhalb von `/usr/src` abzuspeichern. Wenn Sie Probleme haben, könnten Sie der Versuchung erliegen, `/usr/src` einfach zu löschen und wieder von vorne anzufangen. Wenn Sie so vorgehen, werden Sie kurz darauf merken, dass Sie soeben Ihre Kernelkonfigurationsdatei gelöscht haben.
>
> Editieren Sie immer eine Kopie von `GENERIC`. Änderungen an `GENERIC` können verloren gehen, wenn der Quellbaum aktualisiert wird.
>
> Sie sollten die Konfigurationsdatei an anderer Stelle aufheben und im Verzeichnis *i386* einen Link auf die Datei erstellen.
>
> Beispiel:
>
> ```
> # cd /usr/src/sys/i386/conf
> # mkdir /root/kernels
> # cp GENERIC /root/kernels/MYKERNEL
> # ln -s /root/kernels/MYKERNEL
> ```

Jetzt editieren Sie *MYKERNEL* mit einem Texteditor Ihres Vertrauens. Wenn Sie gerade neu anfangen, ist Ihnen vielleicht nur der **vi** Editor bekannt, der allerdings zu komplex ist, um hier erklärt zu werden. Er wird aber in vielen

Büchern aus der Bibliographie gut erklärt. FreeBSD bietet aber auch einen leichter zu benutzenden Editor, den **ee** an, den Sie, wenn Sie Anfänger sind, benutzen sollten. Sie können die Kommentare am Anfang der Konfigurationsdatei ändern, um die Änderungen gegenüber `GENERIC` zu dokumentieren.

Falls Sie schon einmal einen Kernel unter SunOS oder einem anderen BSD kompiliert haben, werden Sie diese Konfigurationsdatei bereits kennen. Wenn Sie mit einem anderen Betriebssystem wie DOS vertraut sind, könnte die `GENERIC` Konfigurationsdatei Sie verschrecken. In diesen Fall sollten Sie den Beschreibungen im Abschnitt über die Konfigurationsdatei langsam und vorsichtig folgen.

> **Anmerkung:** Wenn Sie die FreeBSD Quellen synchronisieren, sollten Sie immer, bevor Sie etwas verändern, `/usr/src/UPDATING` durchlesen. Diese Datei enthält alle wichtigen Informationen, die Sie beim Aktualisieren beachten müssen. Da `/usr/src/UPDATING` immer zu Ihrer Version der FreeBSD Quellen passt, sind die Informationen dort genauer, als in diesem Handbuch.

Nun müssen Sie die Kernelquellen kompilieren.

Den Kernel bauen

> **Anmerkung:** Sie benötigen den kompletten Quellcodebaum, um den Kernel zu bauen.

1. Wechseln Sie in das Verzeichnis `/usr/src`:

   ```
   # cd /usr/src
   ```

2. Kompilieren Sie den neuen Kernel:

   ```
   # make buildkernel KERNCONF=MYKERNEL
   ```

3. Installieren Sie den neuen Kernel:

   ```
   # make installkernel KERNCONF=MYKERNEL
   ```

> **Tipp:** In der Voreinstellung werden beim Bau eines angepassten Kernels stets *alle* Kernelmodule neu gebaut. Wollen Sie Ihren Kernel schneller bauen oder nur bestimmte Module bauen, sollten Sie `/etc/make.conf` anpassen, bevor Sie Ihren Kernel bauen:
>
> ```
> MODULES_OVERRIDE = linux acpi sound/sound sound/driver/ds1 ntfs
> ```
>
> Wenn Sie diese Variable setzen, werden ausschließlich die hier angegebenen Module gebaut (und keine anderen).
>
> ```
> WITHOUT_MODULES = linux acpi sound ntfs
> ```
>
> Durch das Setzen dieser Variable werden werden alle Module auf oberster Ebene bis auf die angegebenen gebaut. Weitere Variablen, die beim Bau eines Kernels von Interesse sein könnten, finden Sie in make.conf(5).

Der neue Kernel wird im Verzeichnis `/boot/kernel`, genauer unter `/boot/kernel/kernel` abgelegt, während der alte Kernel nach `/boot/kernel.old/kernel` verschoben wird. Um den neuen Kernel zu benutzen, sollten Sie Ihren Rechner jetzt neu starten. Falls etwas schief geht, sehen Sie bitte in dem Abschnitt zur Fehlersuche am Ende

dieses Kapitels nach. Dort sollten Sie auch unbedingt den Abschnitt lesen, der erklärt, was zu tun ist, wenn der neue Kernel nicht startet.

> **Anmerkung:** Im Verzeichnis `/boot` werden andere Dateien, die zum Systemstart benötigt werden, wie der Boot-Loader (loader(8)) und dessen Konfiguration, abgelegt. Module von Fremdherstellern oder angepasste Module werden in `/boot/kernel` abgelegt. Beachten Sie bitte, dass diese Module immer zu dem verwendeten Kernel passen müssen. Module, die nicht zu dem verwendeten Kernel passen, gefährden die Stabilität des Systems.

9.6. Die Kernelkonfigurationsdatei

Aktualisiert von Joel Dahl.

Das Format der Konfigurationsdatei ist recht einfach. Jede Zeile enthält ein Schlüsselwort und ein oder mehrere Argumente. Eine Zeile, die von einen # eingeleitet wird, gilt als Kommentar und wird ignoriert. Die folgenden Abschnitte beschreiben jedes Schlüsselwort in der Reihenfolge, in der es in `GENERIC` auftaucht. Eine ausführliche Liste aller Optionen mit detaillierten Erklärungen finden Sie in der Konfigurationsdatei `NOTES`, die sich in demselben Verzeichnis wie die Datei `GENERIC` befindet. Von der Architektur unabhängige Optionen sind in der Datei `/usr/src/sys/conf/NOTES` aufgeführt.

Es ist möglich, eine `include`-Anweisung in die Kernelkonfigurationsdatei aufzunehmen. Diese erlaubt das lokale Einfügen von anderen Konfigurationsdateien in die aktuelle, was es einfacher macht, kleinere Änderungen an einer existierenden Datei zu vollziehen. Wenn Sie beispielsweise einen `GENERIC`-Kernel mit nur einer kleinen Anzahl von zusätzlichen Optionen und Treibern benötigen, brauchen Sie mit den folgenden Zeilen nur ein kleines Delta im Vergleich zu GENERIC anpassen:

```
include GENERIC
ident MYKERNEL

options         IPFIREWALL
options         DUMMYNET
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPDIVERT
```

Für viele Administratoren bietet dieses Modell entscheidende Vorteile über das bisherige Erstellen von Konfigurationsdateien von Grund auf: die lokalen Konfigurationdateien enthalten auch nur die lokalen Unterschiede zu einem `GENERIC`-Kernel und sobald Aktulaisierungen durchgeführt werden, können neue Eigenschaften, die zu `GENERIC` hinzugefügt werden, auch dem lokalen Kernel angehängt werden, es sei denn, es wird durch `nooptions` oder `nodevice` verhindert. Der übrige Teil dieses Kapitels behandelt die Inhalte einer typischen Konfigurationsdatei und die Rolle, die unterschiedliche Optionen und Geräte dabei spielen.

> **Anmerkung:** Um einen Kernel mit allen möglichen Optionen zu bauen beispielsweise für Testzwecke), führen Sie als `root` die folgenden Befehle aus:
>
> ```
> # cd /usr/src/sys/i386/conf && make LINT
> ```

Das folgende Beispiel zeigt eine `GENERIC` Konfigurationsdatei, die, wo notwendig, zusätzliche Kommentare enthält. Sie sollte der Datei `/usr/src/sys/i386/conf/GENERIC` auf Ihrem System sehr ähnlich sein.

```
machine         i386
```

Gibt die Architektur der Maschine an und muss entweder `amd64`, `i386`, `ia64`, `pc98`, `powerpc` oder `sparc64` sein.

```
cpu          I486_CPU
cpu          I586_CPU
cpu          I686_CPU
```

Die vorigen Zeilen geben den Typ der CPU Ihres Systems an. Sie können mehrere CPU Typen angeben, wenn Sie sich zum Beispiel nicht sicher sind, ob Sie `I586_CPU` oder `I686_CPU` benutzen sollen. Für einen angepassten Kernel ist es aber am besten, wenn Sie nur die CPU angeben, die sich in der Maschine befindet. Der CPU-Typ wird in den Boot-Meldungen ausgegeben, die in der Datei `/var/run/dmesg.boot` gespeichert sind.

```
ident          GENERIC
```

Gibt den Namen Ihres Kernels an. Hier sollten Sie den Namen einsetzen, den Sie Ihrer Konfigurationsdatei gegeben haben. In unserem Beispiel ist das `MYKERNEL`. Der Wert, den Sie `ident` zuweisen, wird beim Booten des neuen Kernels ausgegeben. Wenn Sie den Kernel von Ihrem normal verwendeten Kernel unterscheiden wollen, weil Sie zum Beispiel einen Kernel zum Testen bauen, ist es nützlich, hier einen anderen Namen anzugeben.

```
#To statically compile in device wiring instead of /boot/device.hints
#hints          "GENERIC.hints"         # Default places to look for devices.
```

Unter FreeBSD werden Geräte mit device.hints(5) konfiguriert. In der Voreinstellung überprüft loader(8) beim Systemstart die Datei `/boot/device.hints`. Die Option `hints` erlaubt es, die Gerätekonfiguration statisch in den Kernel einzubinden, sodass die Datei `device.hints` in `/boot` nicht benötigt wird.

```
makeoptions     DEBUG=-g          # Build kernel with gdb(1) debug symbols
```

Der normale Bauprozess von FreeBSD erstellt nur dann einen Kernel, der Debugging-Informationen enthält, wenn Sie die Option `-g` von gcc(1) aktivieren.

```
options          SCHED_ULE        # ULE scheduler
```

Der voreingestellte Scheduler von FreeBSD. Ändern Sie diesen Wert nicht!

```
options          PREEMPTION        # Enable kernel thread preemption
```

Erlaubt es Kernelthreads, vor Threads eigentlich höherer Priorität ausgeführt zu werden. Die Interaktivität des Systems wird dadurch erhölt. Interrupt-Threads werden dabei bevorzugt ausgeführt.

```
options          INET             # InterNETworking
```

Netzwerkunterstützung. Auch wenn Sie nicht planen, den Rechner mit einem Netzwerk zu verbinden, sollten Sie diese Option aktiviert lassen. Die meisten Programme sind mindestens auf die Loopback Unterstützung (Verbindungen mit sich selbst) angewiesen. Damit ist diese Option im Endeffekt notwendig.

```
options          INET6            # IPv6 communications protocols
```

Aktiviert die Unterstützung für das IPv6 Protokoll.

```
options         FFS             # Berkeley Fast Filesystem
```

Das Dateisystem für Festplatten. Wenn Sie von einer Festplatte booten wollen, lassen Sie diese Option aktiviert.

```
options         SOFTUPDATES     # Enable FFS Soft Updates support
```

Mit dieser Option wird die Unterstützung für Soft Updates, die Schreibzugriffe beschleunigen, in den Kernel eingebunden. Auch wenn die Funktion im Kernel ist, muss sie für einzelne Dateisysteme explizit aktiviert werden. Überprüfen Sie mit mount(8), ob die Dateisysteme Soft Updates benutzen. Wenn die Option `soft-updates` nicht aktiviert ist, können Sie die Option nachträglich mit tunefs(8) aktivieren. Für neue Dateisysteme können Sie Option beim Anlegen mit newfs(8) aktivieren.

```
options         UFS_ACL         # Support for access control lists
```

Diese Option aktiviert die Unterstützung für Zugriffskontrolllisten (ACL). Die ACLs hängen von erweiterten Attributen und UFS2 ab, eine genaue Beschreibung finden Sie in Abschnitt 15.11. Die Zugriffskontrolllisten sind in der Voreinstellung aktiviert und sollten auch nicht deaktiviert werden, wenn Sie schon einmal auf einem Dateisystem verwendet wurden, da dies die Zugriffsrechte auf Dateien in unvorhersehbarer Art und Weise ändern kann.

```
options         UFS_DIRHASH     # Improve performance on big directories
```

Diese Option steigert die Geschwindigkeit von Plattenzugriffen auf großen Verzeichnissen. Dadurch verbraucht das System etwas mehr Speicher als vorher. Für stark beschäftigte Server oder Arbeitsplatzrechner sollten Sie diese Option aktiviert lassen. Auf kleineren Systemen, bei denen Speicher eine kostbare Ressource darstellt oder Systemen, auf denen die Geschwindigkeit der Plattenzugriffe nicht wichtig ist, wie Firewalls, können Sie diese Option abstellen.

```
options         MD_ROOT         # MD is a potential root device
```

Diese Option aktiviert die Unterstüztung für ein Root-Dateisystem auf einem speicherbasierten Laufwerk (RAM-Disk).

```
options         NFSCLIENT       # Network Filesystem Client
options         NFSSERVER       # Network Filesystem Server
options         NFS_ROOT        # NFS usable as /, requires NFSCLIENT
```

Das Network Filesystem. Wenn Sie keine Partitionen von einem UNIX File-Server über TCP/IP einhängen wollen, können Sie diese Zeile auskommentieren.

```
options         MSDOSFS         # MSDOS Filesystem
```

Das MS-DOS Dateisystem. Sie können diese Zeile auskommentieren, wenn Sie nicht vorhaben, eine DOS-Partition beim Booten einzuhängen. Das nötige Modul wird ansonsten automatisch geladen, wenn Sie das erste Mal eine DOS-Partition einhängen. Außerdem können Sie mit den ausgezeichneten `emulators/mtools` aus der Ports-Sammlung auf DOS-Floppies zugreifen, ohne diese an- und abhängen zu müssen (`MSDOSFS` wird in diesem Fall nicht benötigt).

```
options         CD9660          # ISO 9660 Filesystem
```

Das ISO 9660 Dateisystem für CD-ROMs. Sie können diese Zeile auskommentieren, wenn Sie kein CD-ROM-Laufwerk besitzen oder nur ab und an CDs einhängen. Das Modul wird automatisch geladen, sobald Sie das erste Mal eine CD einhängen. Für Audio-CDs benötigen Sie dieses Dateisystem nicht.

```
options         PROCFS          # Process filesystem (requires PSEUDOFS)
```

Das Prozessdateisystem. Dies ist ein Pseudo-Dateisystem, das auf `/proc` eingehangen wird und es Programmen wie ps(1) erlaubt, mehr Informationen über laufende Prozesse auszugeben. `PROCFS` sollte von FreeBSD nicht mehr benötigt werden, da die meisten Debug- und Überwachungs-Werkzeuge nicht mehr darauf angewiesen sind. Daher wird das Prozessdateisystem auch nicht mehr automatisch in das System eingebunden.

```
options         PSEUDOFS        # Pseudo-filesystem framework
```

Kernel, die `PROCFS` verwenden, müssen auch die Option `PSEUDOFS` verwenden.

```
options         GEOM_PART_GPT         # GUID Partition Tables.
```

Diese Option ermöglicht eine große Anzahl Partitionen auf einem einzelnen Laufwerk.

```
options         COMPAT_43       # Compatible with BSD 4.3 [KEEP THIS!]
```

Stellt die Kompatibilität zu 4.3BSD sicher. Belassen Sie diese Option, da sich manche Programme recht sonderbar verhalten werden, wenn Sie diese auskommentieren.

```
options         COMPAT_FREEBSD4 # Compatible with FreeBSD4
```

Diese Option stellt sicher, dass Anwendungen, die auf älteren FreeBSD Versionen übersetzt wurden und alte Systemaufrufe verwenden, noch lauffähig sind. Wir empfehlen, diese Option auf allen i386-Systemen zu verwenden, auf denen vielleicht noch ältere Anwendungen laufen sollen. Auf Plattformen, die erst ab FreeBSD 5.0 unterstützt werden (wie ia64 und SPARC®), wird diese Option nicht benötigt.

```
options         COMPAT_FREEBSD5 # Compatible with FreeBSD5
```

Diese Option wird ab FreeBSD 6.X benötigt, um Programme, die unter FreeBSD 5.X-Versionen mit FreeBSD 5.X-Systemaufrufen kompiliert wurden, unter FreeBSD 6.X ausführen zu können.

```
options         COMPAT_FREEBSD6 # Compatible with FreeBSD6
```

Diese Option wird ab FreeBSD 7.X benötigt, um Programme, die unter FreeBSD 6.X-Versionen mit FreeBSD 6.X-Systemaufrufen kompiliert wurden, unter FreeBSD 7.X ausführen zu können.

```
options         COMPAT_FREEBSD7 # Compatible with FreeBSD7
```

Diese Option wird ab FreeBSD 8.X benötigt, um Programme, die unter FreeBSD 7.X-Versionen mit FreeBSD 7.X-Systemaufrufen kompiliert wurden, unter FreeBSD 8.X ausführen zu können.

```
options         SCSI_DELAY=5000  # Delay (in ms) before probing SCSI
```

Dies weist den Kernel an, 5 Sekunden zu warten, bevor er anfängt nach SCSI-Geräten auf dem System zu suchen. Wenn Sie nur IDE-Geräte besitzen, können Sie die Anweisung ignorieren. Sie können versuchen, den Wert zu senken, um den Startvorgang zu beschleunigen. Wenn FreeBSD dann Schwierigkeiten hat, Ihre SCSI-Geräte zu erkennen, sollten Sie den Wert natürlich wieder erhöhen.

```
options         KTRACE          # ktrace(1) support
```

Dies schaltet die Kernel-Prozessverfolgung (engl. *kernel process tracing*) ein, die sehr nützlich bei der Fehlersuche ist.

```
options         SYSVSHM         # SYSV-style shared memory
```

Diese Option aktiviert die Unterstützung für System V Shared-Memory. Die XSHM-Erweiterung von X benötigt diese Option und viele Graphik-Programme werden die Erweiterung automatisch benutzen und schneller laufen. Wenn Sie X benutzen, sollten Sie diese Option auf jeden Fall aktivieren.

```
options         SYSVMSG         # SYSV-style message queues
```

Unterstützung für System V Messages. Diese Option vergrößert den Kernel nur um einige hundert Bytes.

```
options         SYSVSEM         # SYSV-style semaphores
```

Unterstützung für System V Semaphoren. Dies wird selten gebraucht, vergrößert aber den Kernel nur um einige hundert Bytes.

> **Anmerkung:** Die Option `-p` des Kommandos ipcs(1) zeigt Programme an, die diese System V Erweiterungen benutzen.

```
options             _KPOSIX_PRIORITY_SCHEDULING # POSIX P1003_1B real-time extensions
```

Echtzeit-Erweiterungen, die 1993 zu POSIX® hinzugefügt wurden. Bestimmte Programme wie **StarOffice** benutzen diese Erweiterungen.

```
options         KBD_INSTALL_CDEV  # install a CDEV entry in /dev
```

Diese Option erstellt für die Tastatur einen Eintrag im Verzeichnis `/dev`.

```
options         ADAPTIVE_GIANT    # Giant mutex is adaptive.
```

Giant ist der Name einer Sperre (Mutex) die viele Kernel-Ressourcen schützt. Heutzutage ist Giant ein unannehmbarer Engpass, der die Leistung eines Systems beeinträchtigt. Daher wird Giant durch Sperren ersetzt, die einzelne Ressourcen schützen. Die Option `ADAPTIVE_GIANT` fügt Giant zu den Sperren hinzu, auf die gewartet werden kann. Ein Thread, der die Sperre Giant von einem anderen Thread benutzt vorfindet, kann nun weiterlaufen und auf die Sperre Giant warten. Früher wäre der Prozess in den schlafenden Zustand (*sleep*) gewechselt und hätte darauf warten müssen, dass er wieder laufen kann. Wenn Sie sich nicht sicher sind, belassen Sie diese Option.

> **Anmerkung:** Beachten Sie, dass ab FreeBSD 8.0-RELEASE und neuer alle Mutexe in der Voreinstellung adaptiv sein werden, es sei denn, Sie werden durch das Setzen der Option `NO_ADAPTIVE_MUTEXES` explizit als nichtadaptiv deklariert. Als Folge dessen ist Giant nun in in der Voreinstellung ebenfalls adaptiv, daher ist in diesen Versionen die Kerneloption `ADAPTIVE_GIANT` nicht mehr in der Kernelkonfigurationsdatei enthalten.

```
device          apic            # I/O APIC
```

Das apic-Gerält ermöglicht die Benutzung des I/O APIC für die Interrupt-Auslieferung. Das apic-Gerält kann mit Kerneln für Einprozessorsysteme und Mehrprozessorsysteme benutzt werden. Kernel für Mehrprozessorsysteme benötigen diese Option zwingend. Die Unterstützung für Mehrprozessorsysteme aktivieren Sie mit der Option `options SMP`.

Anmerkung: Das apic-Gerät existiert nur unter der i386-Architektur, daher ist es sinnlos, diese Zeile unter einer anderen Architektur in Ihre Kernelkonfigurationsdatei aufzunehmen.

```
device          eisa
```

Fügen Sie diese Zeile ein, wenn Sie ein EISA-Motherboard besitzen. Dies aktiviert die Erkennung und Konfiguration von allen Geräten auf dem EISA Bus.

```
device          pci
```

Wenn Sie ein PCI-Motherboard besitzen, fügen Sie diese Zeile ein. Dies aktiviert die Erkennung von PCI-Karten und die PCI-ISA bridge.

```
# Floppy drives
device          fdc
```

Der Floppy-Controller.

```
# ATA and ATAPI devices
device          ata
```

Dieser Treiber unterstützt alle ATA und ATAPI Geräte. Eine `device ata` Zeile reicht aus und der Kernel wird auf modernen Maschinen alle PCI ATA/ATAPI Geräte entdecken.

```
device          atadisk                 # ATA disk drives
```

Für ATA-Plattenlaufwerke brauchen Sie diese Zeile zusammen mit `device ata`.

```
device          ataraid                 # ATA RAID drives
```

Für ATA-RAID brauchen Sie diese Zeile zusammen mit `device ata`.

```
device          atapicd                 # ATAPI CDROM drives
```

Zusammen mit `device ata` wird dies für ATAPI CD-ROM Laufwerke benötigt.

```
device          atapifd                 # ATAPI floppy drives
```

Zusammen mit `device ata` wird dies für ATAPI Floppy Laufwerke benötigt.

```
device          atapist                 # ATAPI tape drives
```

Zusammen mit `device ata` wird dies für ATAPI Bandlaufwerke benötigt.

```
options         ATA_STATIC_ID           # Static device numbering
```

Erzwingt eine statische Gerätenummer für den Controller; ohne diese Option werden die Nummern dynamisch zugeteilt.

```
# SCSI Controllers
device          ahb        # EISA AHA1742 family
device          ahc        # AHA2940 and onboard AIC7xxx devices
options         AHC_REG_PRETTY_PRINT     # Print register bitfields in debug
```

```
                                        # output.  Adds ~128k to driver.
device          ahd         # AHA39320/29320 and onboard AIC79xx devices
options         AHD_REG_PRETTY_PRINT    # Print register bitfields in debug
                                        # output.  Adds ~215k to driver.
device          amd         # AMD 53C974 (Teckram DC-390(T))
device          isp         # Qlogic family
#device         ispfw       # Firmware for QLogic HBAs- normally a module
device          mpt         # LSI-Logic MPT-Fusion
#device         ncr         # NCR/Symbios Logic
device          sym         # NCR/Symbios Logic (newer chipsets + those of 'ncr'))
device          trm         # Tekram DC395U/UW/F DC315U adapters

device          adv         # Advansys SCSI adapters
device          adw         # Advansys wide SCSI adapters
device          aha         # Adaptec 154x SCSI adapters
device          aic         # Adaptec 15[012]x SCSI adapters, AIC-6[23]60.
device          bt          # Buslogic/Mylex MultiMaster SCSI adapters

device          ncv         # NCR 53C500
device          nsp         # Workbit Ninja SCSI-3
device          stg         # TMC 18C30/18C50
```

SCSI-Controller. Kommentieren Sie alle Controller aus, die sich nicht in Ihrem System befinden. Wenn Sie ein IDE-System besitzen, können Sie alle Einträge entfernen. Die Zeilen mit den `*_REG_PRETTY_PRINT`-Einträgen aktivieren Debugging-Optionen für die jeweiligen Treiber.

```
# SCSI peripherals
device          scbus       # SCSI bus (required for SCSI)
device          ch          # SCSI media changers
device          da          # Direct Access (disks)
device          sa          # Sequential Access (tape etc)
device          cd          # CD
device          pass        # Passthrough device (direct SCSI access)
device          ses         # SCSI Environmental Services (and SAF-TE)
```

SCSI Peripheriegeräte. Kommentieren Sie wieder alle Geräte aus, die Sie nicht besitzen. Besitzer von IDE-Systemen können alle Einträge entfernen.

> **Anmerkung:** Der USB-umass(4)-Treiber und einige andere Treiber benutzen das SCSI-Subsystem obwohl sie keine SCSI-Geräte sind. Belassen Sie die SCSI-Unterstützung im Kernel, wenn Sie solche Treiber verwenden.

```
# RAID controllers interfaced to the SCSI subsystem
device          amr         # AMI MegaRAID
device          arcmsr      # Areca SATA II RAID
device          asr         # DPT SmartRAID V, VI and Adaptec SCSI RAID
device          ciss        # Compaq Smart RAID 5*
device          dpt         # DPT Smartcache III, IV - See NOTES for options
device          hptmv       # Highpoint RocketRAID 182x
device          hptrr       # Highpoint RocketRAID 17xx, 22xx, 23xx, 25xx
device          iir         # Intel Integrated RAID
device          ips         # IBM (Adaptec) ServeRAID
```

```
device          mly        # Mylex AcceleRAID/eXtremeRAID
device          twa        # 3ware 9000 series PATA/SATA RAID

# RAID controllers
device          aac        # Adaptec FSA RAID
device          aacp       # SCSI passthrough for aac (requires CAM)
device          ida        # Compaq Smart RAID
device          mfi        # LSI MegaRAID SAS
device          mlx        # Mylex DAC960 family
device          pst        # Promise Supertrak SX6000
device          twe        # 3ware ATA RAID
```

Unterstützte RAID Controller. Wenn Sie keinen der aufgeführten Controller besitzen, kommentieren Sie die Einträge aus oder entfernen sie.

```
# atkbdc0 controls both the keyboard and the PS/2 mouse
device          atkbdc     # AT keyboard controller
```

Der Tastatur-Controller (`atkbdc`) ist für die Ein- und Ausgabe von AT-Tastaturen und PS/2 Zeigegeräten (z.B. einer Maus) verantwortlich. Dieser Controller wird vom Tastaturtreiber (`atkbd`) und dem PS/2 Gerätetreiber (`psm`) benötigt.

```
device          atkbd      # AT keyboard
```

Zusammen mit dem `atkbdc` Controller bietet der `atkbd` Treiber Zugriff auf AT-Tastaturen.

```
device          psm        # PS/2 mouse
```

Benutzen Sie dieses Gerät, wenn Sie eine Maus mit PS/2 Anschluss besitzen.

```
device          kbdmux        # keyboard multiplexer
```

Basisunterstützung für Tastaturmultiplexer. Verwenden Sie nur eine einzige Tastatur, können Sie diese Zeile aus Ihrer Kernelkonfigurationsdatei entfernen.

```
device          vga        # VGA video card driver
```

Der Grafikkartentreiber.

```
device          splash     # Splash screen and screen saver support
```

Zeigt einen “Splash Screen” beim Booten. Diese Zeile wird auch von den Bildschirmschonern benötigt.

```
# syscons is the default console driver, resembling an SCO console
device          sc
```

`sc` ist in der Voreinstellung der Treiber für die Konsole, die der SCO-Konsole ähnelt. Da die meisten bildschirmorientierten Programme auf die Konsole mit Hilfe einer Datenbank wie `termcap` zugreifen, sollte es keine Rolle spielen, ob Sie diesen Treiber oder `vt`, den `VT220` kompatiblen Konsolentreiber einsetzen. Wenn Sie Probleme mit bildschirmorientierten Anwendungen unter dieser Konsole haben, setzen Sie beim Anmelden die Variable `TERM` auf den Wert `VT220`.

```
# Enable this for the pcvt (VT220 compatible) console driver
#device          vt
```

```
#options         XSERVER          # support for X server on a vt console
#options         FAT_CURSOR       # start with block cursor
```

Der VT220 kompatible Konsolentreiber ist kompatibel zu VT100/102. Auf einigen Laptops, die aufgrund der Hardware inkompatibel zum `sc` Treiber sind, funktioniert dieser Treiber gut. Beim Anmelden sollten Sie die Variable `TERM` auf den Wert `vt100` setzen. Dieser Treiber kann sich als nützlich erweisen, wenn Sie sich über das Netzwerk auf vielen verschiedenen Maschinen anmelden, da dort oft Einträge in `termcap` oder `terminfo` für das `sc` Gerät fehlen. Dagegen sollte `vt100` auf jeder Plattform unterstützt werden.

```
device          agp
```

Fügen Sie diese Zeile ein, wenn Sie eine AGP-Karte besitzen. Damit werden Motherboards mit AGP und AGP GART unterstützt.

```
# Power management support (see NOTES for more options)
#device          apm
```

Unterstützung zur Energieverwaltung. Diese Option ist nützlich für Laptops, allerdings ist sie in `GENERIC` deaktiviert.

```
# Add suspend/resume support for the i8254.
device          pmtimer
```

Zeitgeber für Ereignisse der Energieverwaltung (APM und ACPI).

```
# PCCARD (PCMCIA) support
# PCMCIA and cardbus bridge support
device          cbb             # cardbus (yenta) bridge
device          pccard          # PC Card (16-bit) bus
device          cardbus         # CardBus (32-bit) bus
```

PCMCIA Unterstützung. Wenn Sie einen Laptop benutzen, brauchen Sie diese Zeile.

```
# Serial (COM) ports
device          sio             # 8250, 16[45]50 based serial ports
```

Die seriellen Schnittstellen, die in der MS-DOS- und Windows-Welt `COM` genannt werden.

> **Anmerkung:** Wenn Sie ein internes Modem, das `COM4` benutzt, besitzen und eine serielle Schnittstelle haben, die auf `COM2` liegt, müssen Sie den IRQ des Modems auf 2 setzen (wegen undurchsichtigen technischen Gründen ist IRQ2 gleich IRQ9). Wenn Sie eine serielle Multiport-Karte besitzen, entnehmen Sie bitte die Werte, die Sie in die Datei `/boot/device.hints` einfügen müssen, der Hilfeseite sio(4). Einige Graphikkarten, besonders die auf S3-Chips basierten, benutzen IO-Adressen der Form `0x*2e8` und manche billige serielle Karten dekodieren den 16-Bit IO-Adressraum nicht sauber. Dies führt zu Konflikten und blockiert dann die `COM4`-Schnittstelle.
>
> Jeder seriellen Schnittstelle muss ein eigener IRQ zugewiesen werden (wenn Sie eine Multiport-Karte verwenden, bei der das Teilen von Interrupts unterstützt wird, muss das nicht der Fall sein), daher können in der Voreinstellung `COM3` und `COM4` nicht benutzt werden.

```
# Parallel port
device          ppc
```

Die parallele Schnittstelle auf dem ISA Bus.

```
device          ppbus      # Parallel port bus (required)
```

Unterstützung für den Bus auf der parallelen Schnittstelle.

```
device          lpt        # Printer
```

Unterstützung für Drucker über die parallele Schnittstelle.

> **Anmerkung:** Sie brauchen jede der drei Zeilen, um die Unterstützung für einen Drucker an der parallelen Schnittstelle zu aktivieren.

```
device          plip       # TCP/IP over parallel
```

Der Treiber für das Netzwerkinterface über die parallele Schnittstelle.

```
device          ppi        # Parallel port interface device
```

Allgemeine I/O (“geek port”) und IEEE1284 I/O Unterstützung.

```
#device         vpo        # Requires scbus and da
```

Dies aktiviert den Treiber für ein Iomega Zip Laufwerk. Zusätzlich benötigen Sie noch die Unterstützung für `scbus` und `da`. Die beste Performance erzielen Sie, wenn Sie die Schnittstelle im EPP 1.9 Modus betreiben.

```
#device         puc
```

Aktivieren Sie diesen Treiber, wenn Sie eine serielle oder parallele PCI-Karte besitzen, die vom Treiber puc(4) unterstützt wird.

```
# PCI Ethernet NICs.
device          de         # DEC/Intel DC21x4x (“Tulip”)
device          em         # Intel PRO/1000 adapter Gigabit Ethernet Card
device          ixgb       # Intel PRO/10GbE Ethernet Card
device          txp        # 3Com 3cR990 (“Typhoon”)
device          vx         # 3Com 3c590, 3c595 (“Vortex”)
```

Verschiedene Treiber für PCI-Netzwerkkarten. Geräte, die sich nicht in Ihrem System befinden, können Sie entfernen oder auskommentieren.

```
# PCI Ethernet NICs that use the common MII bus controller code.
# NOTE: Be sure to keep the ’device miibus’ line in order to use these NICs!
device          miibus     # MII bus support
```

Einige PCI 10/100 Ethernet Netzwerkkarten, besonders die, die MII-fähige Transceiver verwenden oder Transceiver-Steuerungen implementieren, die ähnlich wie MII funktionieren, benötigen die Unterstützung für den MII-Bus. Die Zeile `device miibus` fügt dem Kernel die Unterstützung für das allgemeine miibus API und allen PHY-Treibern hinzu.

```
device          bce        # Broadcom BCM5706/BCM5708 Gigabit Ethernet
device          bfe        # Broadcom BCM440x 10/100 Ethernet
```

```
device          bge        # Broadcom BCM570xx Gigabit Ethernet
device          dc         # DEC/Intel 21143 and various workalikes
device          fxp        # Intel EtherExpress PRO/100B (82557, 82558)
device          lge        # Level 1 LXT1001 gigabit ethernet
device          msk        # Marvell/SysKonnect Yukon II Gigabit Ethernet
device          nge        # NatSemi DP83820 gigabit ethernet
device          nve        # nVidia nForce MCP on-board Ethernet Networking
device          pcn        # AMD Am79C97x PCI 10/100 (precedence over 'lnc')
device          re         # RealTek 8139C+/8169/8169S/8110S
device          rl         # RealTek 8129/8139
device          sf         # Adaptec AIC-6915 ("Starfire")
device          sis        # Silicon Integrated Systems SiS 900/SiS 7016
device          sk         # SysKonnect SK-984x & SK-982x gigabit Ethernet
device          ste        # Sundance ST201 (D-Link DFE-550TX)
device          stge       # Sundance/Tamarack TC9021 gigabit Ethernet
device          ti         # Alteon Networks Tigon I/II gigabit Ethernet
device          tl         # Texas Instruments ThunderLAN
device          tx         # SMC EtherPower II (83c170 "EPIC")
device          vge        # VIA VT612x gigabit ethernet
device          vr         # VIA Rhine, Rhine II
device          wb         # Winbond W89C840F
device          xl         # 3Com 3c90x ("Boomerang", "Cyclone")
```

Treiber, die den MII Bus Controller Code benutzen.

```
# ISA Ethernet NICs.  pccard NICs included.
device          cs         # Crystal Semiconductor CS89x0 NIC
# 'device ed' requires 'device miibus'
device          ed         # NE[12]000, SMC Ultra, 3c503, DS8390 cards
device          ex         # Intel EtherExpress Pro/10 and Pro/10+
device          ep         # Etherlink III based cards
device          fe         # Fujitsu MB8696x based cards
device          ie         # EtherExpress 8/16, 3C507, StarLAN 10 etc.
device          lnc        # NE2100, NE32-VL Lance Ethernet cards
device          sn         # SMC's 9000 series of Ethernet chips
device          xe         # Xircom pccard Ethernet

# ISA devices that use the old ISA shims
#device         le
```

Treiber für ISA Ethernet Karten. Schauen Sie in `/usr/src/sys/`*`i386`*`/conf/NOTES` nach, um zu sehen, welche Karte von welchem Treiber unterstützt wird.

```
# Wireless NIC cards
device          wlan       # 802.11 support
```

Generische 802.11-Unterstützung. Diese Zeile wird unbedingt benötigt, wenn Sie WLAN nutzen wollen.

```
device          wlan_wep        # 802.11 WEP support
device          wlan_ccmp       # 802.11 CCMP support
device          wlan_tkip       # 802.11 TKIP support
```

Krypto-Unterstützung für 802.11-Geräte. Sie benötigen diese Zeilen, wenn Sie Ihr drahtloses Netzwerk verschlüsseln und die 802.11-Sicherheitsprotokolle einsetzen wollen.

```
device          an              # Aironet 4500/4800 802.11 wireless NICs
device          ath             # Atheros pci/cardbus NIC's
device          ath_hal         # Atheros HAL (Hardware Access Layer)
device          ath_rate_sample # SampleRate tx rate control for ath
device          awi        # BayStack 660 and others
device          ral        # Ralink Technology RT2500 wireless NICs.
device          wi         # WaveLAN/Intersil/Symbol 802.11 wireless NICs.
#device         wl         # Older non 802.11 Wavelan wireless NIC.
```

Treiber für drahtlose Netzwerkkarten (WLAN).

```
# Pseudo devices
device   loop          # Network loopback
```

Das TCP/IP Loopback Device. Wenn Sie eine Telnet oder FTP Verbindung zu `localhost` (alias `127.0.0.1`) aufbauen, erstellen Sie eine Verbindung zu sich selbst durch dieses Device. Die Angabe dieser Option ist *verpflichtend*.

```
device   random        # Entropy device
```

Kryptographisch sicherer Zufallszahlengenerator.

```
device   ether         # Ethernet support
```

`ether` brauchen Sie nur, wenn Sie eine Ethernet-Karte besitzen. Der Treiber unterstützt das Ethernet-Protokoll.

```
device   sl            # Kernel SLIP
```

`sl` aktiviert die SLIP-Unterstützung. SLIP ist fast vollständig von PPP verdrängt worden, da letzteres leichter zu konfigurieren, besser geeignet für Modem zu Modem Kommunikation und mächtiger ist.

```
device   ppp           # Kernel PPP
```

Dies ist Kernel Unterstützung für PPP-Wählverbindungen. Es existiert auch eine PPP-Version im Userland, die den `tun` Treiber benutzt. Die Userland-Version ist flexibler und bietet mehr Option wie die Wahl auf Anforderung.

```
device   tun           # Packet tunnel.
```

Dies wird vom der Userland PPP benutzt. Die *`Zahl`* hinter `tun` gibt die Anzahl der unterstützten gleichzeitigen Verbindungen an. Weitere Informationen erhalten Sie im Abschnitt PPP dieses Handbuchs.

```
device   pty           # Pseudo-ttys (telnet etc)
```

Dies ist ein "Pseudo-Terminal" oder simulierter Login-Terminal. Er wird von einkommenden `telnet` und `rlogin` Verbindungen, **xterm** und anderen Anwendungen wie **Emacs** benutzt.

```
device   md            # Memory "disks"
```

Pseudo-Gerät für Speicher-Laufwerke.

```
device   gif           # IPv6 and IPv4 tunneling
```

Dieses Gerät tunnelt IPv6 über IPv4, IPv4 über IPv6, IPv4 über IPv4 oder IPv6 über IPv6. Das Gerät `gif` kann die Anzahl der benötigten Geräte automatisch bestimmen (“auto-cloning”).

```
device   faith         # IPv6-to-IPv4 relaying (translation)
```

Dieses Pseudo-Gerät fängt zu ihm gesendete Pakete ab und leitet Sie zu einem Dæmon weiter, der Verkehr zwischen IPv4 und IPv6 vermittelt.

```
# The ‘bpf’ device enables the Berkeley Packet Filter.
# Be aware of the administrative consequences of enabling this!
# Note that ’bpf’ is required for DHCP.
device   bpf           # Berkeley packet filter
```

Das ist der Berkeley Paketfilter. Dieses Pseudo-Gerät kann Netzwerkkarten in den “promiscuous” Modus setzen und erlaubt es damit, Pakete auf einem Broadcast Netzwerk (z.B. einem Ethernet) einzufangen. Die Pakete können auf der Festplatte gespeichert und mit tcpdump(1) untersucht werden.

> **Anmerkung:** Das bpf(4)-Gerät wird von dhclient(8) genutzt, um die IP-Adresse des Default-Routers zu bekommen. Wenn Sie DHCP benutzen, lassen Sie diese Option bitte aktiviert.

```
# USB support
device          uhci          # UHCI PCI->USB interface
device          ohci          # OHCI PCI->USB interface
device          ehci          # EHCI PCI->USB interface (USB 2.0)
device          usb           # USB Bus (required)
#device         udbp          # USB Double Bulk Pipe devices
device          ugen          # Generic
device          uhid          # “Human Interface Devices”
device          ukbd          # Keyboard
device          ulpt          # Printer
device          umass         # Disks/Mass storage - Requires scbus and da
device          ums           # Mouse
device          ural          # Ralink Technology RT2500USB wireless NICs
device          urio          # Diamond Rio 500 MP3 player
device          uscanner      # Scanners
# USB Ethernet, requires mii
device          aue           # ADMtek USB Ethernet
device          axe           # ASIX Electronics USB Ethernet
device          cdce          # Generic USB over Ethernet
device          cue           # CATC USB Ethernet
device          kue           # Kawasaki LSI USB Ethernet
device          rue           # RealTek RTL8150 USB Ethernet
```

Unterstützung für verschiedene USB Geräte.

```
# FireWire support
device          firewire      # FireWire bus code
device          sbp           # SCSI over FireWire (Requires scbus and da)
device          fwe           # Ethernet over FireWire (non-standard!)
```

Verschiedene Firewire-Geräte.

Mehr Informationen und weitere von FreeBSD unterstützte Geräte entnehmen Sie bitte `/usr/src/sys/i386/conf/NOTES`.

9.6.1. Hohe Speicheranforderungen (PAE)

Systeme mit hohen Speicheranforderungen benötigen mehr Speicher als den auf 4 Gigabyte beschränkten User- und Kernel-Adressraum (KVA). Mit dem Pentium Pro und neueren CPUs hat Intel den Adressraum auf 36-Bit erweitert.

Die Physical-Address-Extension (PAE) von Intels Pentium Pro und neueren Prozessoren unterstützt bis zu 64 Gigabyte Speicher. FreeBSD kann diesen Speicher mit der Option `PAE` in der Kernelkonfiguration nutzen. Die Option gibt es in allen aktuellen FreeBSD-Versionen. Wegen Beschräkungen der Intel-Speicherarchitektur wird keine Unterscheidung zwischen Speicher oberhalb oder unterhalb von 4 Gigabyte getroffen. Speicher über 4 Gigabyte wird einfach dem zur Verfügung stehenden Speicher zugeschlagen.

Sie aktivieren PAE im Kernel, indem Sie die folgende Zeile in die Kernelkonfigurationsdatei einfügen:

```
options             PAE
```

> **Anmerkung:** FreeBSD unterstützt PAE nur auf IA-32 Prozessoren. Die PAE-Unterstützung wurde zudem noch nicht hinreichend getestet und befindet sich im Vergleich zu anderen Komponenten von FreeBSD noch im Beta-Stadium.

Die PAE-Unterstützung in FreeBSD ist mit den nachstehenden Einschränkungen verbunden:

- Ein Prozess kann nicht mehr als 4 Gigabyte virtuellen Speicher benutzen.
- Gerätetreiber, die nicht die bus_dma(9)-Schnittstelle benutzen, führen zusammen mit einem PAE-Kernel zu Datenverlusten. Diese Treiber sollen nicht mit einem PAE-Kernel verwendet werden. Daher gibt es unter FreeBSD eine zusätzliche PAE-Kernelkonfigurationsdatei, die alle Treiber enthält, die mit einem PAE-Kernel funktionieren.
- Einige Systemvariablen werden abhängig von der Speichergröße eingestellt. In einem PAE-System mit viel Speicher können die Werte daher zu hoch eingestellt sein. Ein Beispiel ist die sysctl-Variable `kern.maxvnodes`, die die maximale Anzahl von vnodes im Kernel bestimmt. Solche Variablen sollten auf einen angemessenen Wert eingestellt werden.
- Es kann erforderlich sein, den virtuellen Adressraum des Kernels (KVA) zu vergrößern oder, wie oben beschrieben, den Wert einer häufig gebrauchten Kernelvariablen zu verringern. Dies verhindert einen Überlauf des KVAs. Der Adressraum des Kernels kann mit der Kerneloption `KVA_PAGES` vergrößert werden.

Hinweise zur Leistungssteigerung und Stabilität entnehmen Sie bitte der Hilfeseite tuning(7). Die PAE-Unterstützung von FreeBSD wird in der Hilfeseite pae(4) beschrieben.

9.7. Wenn etwas schiefgeht

Es gibt vier Hauptfehlerquellen beim Erstellen eines angepassten Kernels:

`config` verursacht Fehler:

Wenn config(8) misslingt, liegen Fehler in der Kernelkonfigurationsdatei vor. Zum Glück gibt config(8) die die Zeilennummer der Fehlerstelle an, sodass Sie den Fehler schnell finden können. Beispielsweise könnten Sie folgende Fehlermeldung sehen:

```
config: line 17: syntax error
```

Vergleichen Sie die angegebene Zeile mit `GENERIC` und stellen Sie sicher, dass das Schlüsselwort richtig geschrieben ist.

`make` verursacht Fehler:

Wenn `make` misslingt, liegen meistens Fehler in der Konfigurationsdatei vor, die aber nicht schwerwiegend genug für config(8) waren. Überprüfen Sie wiederum Ihre Konfiguration und wenn Sie keinen Fehler entdecken können, schicken Sie eine E-Mail mit Ihrer Kernelkonfiguration an die Mailingliste 'Fragen und Antworten zu FreeBSD' <`de-bsd-questions@de.FreeBSD.org`>. Sie sollten dann schnell Hilfe erhalten.

Der Kernel bootet nicht:

Wenn der Kernel nicht booten will, ist das noch lange kein Grund zur Panik. Denn FreeBSD besitzt exzellente Mechanismen zur Wiederherstellung nach dem Einsatz inkompatibler Kernel. Den Kernel, mit dem Sie booten wollen, können Sie sich im FreeBSD Boot-Loader aussuchen. Dazu wählen Sie im Bootmenü die Option "Escape to a loader prompt". Danach geben Sie den Befehl `boot` *`kernel.old`* oder den Namen eines anderen Kernels ein, der sauber bootet. Für alle Fälle sollten Sie immer einen Kernel, der garantiert bootet, bereit halten.

Nun können Sie die Konfiguration noch einmal überprüfen und den Kernel neu kompilieren. Dazu ist `/var/log/messages` sehr nützlich, da hier sämtliche Kernelmeldungen von jedem erfolgreichen Bootvorgang gespeichert werden. dmesg(8) gibt Ihnen die Kernelmeldungen vom letzten Bootvorgang aus.

> **Anmerkung:** Für den Fall, dass Sie Probleme bei dem Kernelbau bekommen, heben Sie sich immer einen `GENERIC` oder einen anderen Kernel, der garantiert bootet, auf. Der Name dieses Kernels sollte so gewählt sein, dass er beim nächsten Bau nicht überschrieben wird. Sie können sich nicht auf `kernel.old` verlassen, da dieser Kernel durch den zuletzt installierten Kernel, der vielleicht schon kaputt war, während der Installation ersetzt wird. Kopieren Sie den funktionierenden Kernel so schnell wie möglich in das richtige Verzeichnis (`/boot/kernel`). Ansonsten funktionieren Kommandos wie ps(1) nicht. Benennen Sie dazu einfach das Verzeichnis des funktionierenden Kernels um:
>
> ```
> # mv /boot/kernel /boot/kernel.bad
> # mv /boot/kernel.good /boot/kernel
> ```

Der Kernel ist in Ordnung, aber `ps` geht nicht mehr:

Wenn Sie eine andere Version des Kernels installiert haben als die, mit der Ihre Systemwerkzeuge gebaut wurden (beispielsweise einen -CURRENT-Kernel auf einem -RELEASE-System), werden Programme wie ps(1) und vmstat(8) nicht mehr funktionieren. Sie sollten nun das komplette System neu bauen und installieren. Achten Sie darauf, dass die Quellen, aus denen Sie das System bauen, zum installierten Kernel passen. Das ist ein Grund dafür, warum man nie einen Kernel, der nicht zur Systemversion passt, benutzen sollten.

Kapitel 10. Drucken

Beigetragen von Sean Kelly. Restrukturiert und aktualisiert durch Jim Mock. Übersetzt von Stefan Bethke.

10.1. Übersicht

Mit FreeBSD können Sie viele unterschiedliche Drucker benutzen, von den ältesten Nadeldruckern bis zu den neuesten Laserdruckern, und allen möglichen Geräten dazwischen. Auf diese Weise können Sie hochwertige Ausdrucke mit Ihren Programmen erzeugen.

Sie können FreeBSD auch so konfigurieren, dass es Druckaufträge von anderen Computern über Ihr lokales Netzwerk entgegennimmt, seien es Windows-, Mac OS- oder andere FreeBSD-Computer. FreeBSD stellt sicher, dass die Druckaufträge in der richtigen Reihenfolge bearbeitet werden und kann optional ein Deckblatt mit dem Namen des Auftraggebers eines Druckauftrags aufgeben. FreeBSD kann auch Statistiken über die Computer und Benutzer führen, die Ausdrucke in Auftrag geben.

In diesem Kapitel erfahren Sie, wie Sie:

- FreeBSD-Druckerwarteschlangen einrichten.
- Druckfilter installieren, die Druckaufträge je nach Bedarf besonders behandeln und z.B. Dokumente automatisch in eine Form umwandeln, die Ihr Drucker versteht.
- Druckaufträge mit einem Deckblatt versehen können.
- Mit einem Drucker drucken können, der an einen anderen Computer angeschlossen ist.
- Mit einem Drucker drucken können, der direkt an das Netzwerk angeschlossen ist.
- die Größe von Druckaufträgen beschränken können, oder bestimmte Benutzer von den Benutzung des Drucksystems ausschließen können.
- Statistiken aufzeichnen und die Benutzung des Drucksystems nach Benutzern und Computern aufschlüsseln können.
- Probleme beim Drucken diagnostizieren und beheben können.

Bevor Sie dieses Kapitel lesen:

- Machen Sie sich mit der Konfiguration und Installation eines neuen Kernels vertraut (Kapitel 9).

10.2. Einführung

Um einen Drucker mit FreeBSD zu benutzen, können Sie das Berkeley Line Printer Spooling System, das auch als **LPD**-Drucksystem oder nur als **LPD** bekannt ist, verwenden. Dieses System zur Verwaltung von Druckaufträgen ist das Standardsystem in FreeBSD. Dieses Kapitel führt Sie in **LPD** und dessen Konfiguration ein.

Wenn Sie bereits mit **LPD** oder einem anderen Drucksystem vertraut sind, können Sie direkt im Abschnitt Einfache Drucker-Konfiguration weiterlesen.

LPD steuert alle Aspekte rund um die Drucker, die an den Computer angeschlossen sind. Es ist verantwortlich für:

- Die Zugriffskontrolle für direkt und über das Netzwerk angeschlossene Drucker.
- Die Entgegennahme von Dateien, die gedruckt werden sollen; eine so an das Drucksystem übergebene Datei wird als *Druckauftrag* bezeichnet.
- Den gleichzeitigen Zugriff von mehreren Benutzern auf einen Drucker. Alle Druckaufträge werden in einer *Druckerwarteschlange* gesammelt, und nacheinander abgearbeitet.
- Den Druck von *Deckblättern* (auch als *Banner*- oder *Burst*-Seiten bezeichnet), damit Benutzer ihre Druckaufträge schnell innerhalb eines Stapels von ausgedruckten Dokumenten finden können.
- Das Einstellen der korrekten Kommunikations-Parameter für Drucker, die seriell angeschlossen sind.
- Das Senden von Druckaufträgen an ein **LPD**-System auf einem anderen Computer.
- Das Ausführen von speziellen Filtern, um Druckaufträge in die unterschiedlichen Seitenbeschreibungssprachen umzusetzen oder an die Fähigkeiten eines Druckers anzupassen.
- Das Erfassen von Verrechnungsdaten für Druckaufträge.

Sie können **LPD** alle diese Funktionen, oder auch nur einen Teil davon, ausführen lassen, indem Sie die Konfigurationsdatei (`/etc/printcap`) anpassen, und indem Sie spezielle Filterprogramme bereitstellen.

10.2.1. Vorteile des Drucksystems

Wenn Sie der einzige Benutzer sind, der mit Ihrem Computer arbeitet, fragen Sie sich vielleicht, warum Sie die Konfigurationsarbeit für das Drucksystem auf sich nehmen sollten, wenn Sie Deckblätter, Abrechnungsdaten oder Zugriffskontrolle nicht benötigen. Obwohl Sie direkt auf den Drucker zugreifen können, bietet **LPD** eine Reihe von Vorteilen:

- **LPD** druckt im Hintergrund; Sie müssen nicht erst darauf warten, dass Ihr Druckauftrag an den Drucker übermittelt worden ist.
- **LPD** kann Druckaufträge mit Kopf- oder Fußzeilen versehen, oder ein spezielles Dateiformat, wie DVI von TEX, automatisch in ein für den Drucker verständliches Format umwandeln; Sie müssen diese Schritte nicht manuell ausführen.
- Viele freie und kommerzielle Programme, mit denen Sie drucken können, erwarten, mithilfe des **LPD**-Drucksystems zu drucken. Wenn Sie das Drucksystem konfiguriert haben, können Sie einfacher mit neuer oder auch vorhandener Software drucken.

10.3. Grund-Konfiguration

> **Warnung:** Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

Um einen Drucker mit dem **LPD**-Drucksystem benutzen zu können, müssen Sie sowohl Ihren Drucker und die Drucker-Schnittstelle als auch das **LPD**-Drucksystem konfigurieren. Dieser Abschnitt beschreibt zwei Konfigurationen:

- Abschnitt Einfache Drucker-Konfiguration beschreibt, wie Sie einen Drucker an Ihren Computer anschließen und **LPD** so konfigurieren, dass Sie Textdateien zum Drucker senden können.
- Abschnitt Erweiterte Drucker-Konfiguration beschreibt, wie Sie mit speziellen Dateiformaten umgehen können, wie Sie Deckblätter drucken können, wie Sie den Zugriff auf Drucker einschränken können, und wie Sie Verrechnungsdaten aufzeichnen können.

10.3.1. Einfache Drucker-Konfiguration

Dieser Abschnitt beschreibt, wie Sie die **LPD**-Software konfigurieren, um Ihren Drucker zu benutzen. Diese Grundlagen werden erklärt:

- Abschnitt Hardware-Konfiguration erläutert, wie Sie Ihren Drucker an Ihren Computer anschließen können.
- Abschnitt Software-Konfiguration erklärt, wie Sie die **LPD**-Konfigurationsdatei (`/etc/printcap`) anpassen.

Wenn Sie einen Drucker einrichten möchten, der über das Netzwerk angeschlossen ist (anstatt über die serielle oder parallele Schnittstelle), lesen Sie bitte Abschnitt Drucker mit direkter TCP-Schnittstelle.

Obwohl dieser Abschnitt “Grund-Konfiguration” heißt, ist die Konfiguration relativ komplex. Es ist vergleichsweise schwierig, einen Drucker mit Ihrem Computer und dem **LPD**-Drucksystem zu verbinden. Die weiteren Optionen, wie Kopfzeilen oder Deckblätter, sind einfach zu konfigurieren, sobald die Grund-Konfiguration erfolgreich abgeschlossen ist.

10.3.1.1. Hardware-Konfiguration

Dieser Abschnitt beschreibt, über welche Schnittstellen Sie einen Drucker mit Ihrem Computer verbinden können. Er behandelt sowohl die Schnittstellen und Kabel, als auch die Kerneloptionen, die Sie benötigen, um FreeBSD mit Ihrem Drucker kommunizieren zu lassen.

Wenn Sie Ihren Drucker bereits erfolgreich mit einem anderen Betriebssystem auf Ihrem PC eingesetzt haben, können Sie wahrscheinlich mit dem Abschnitt Software-Konfiguration fortfahren.

10.3.1.1.1. Schnittstellen und Kabel

Praktisch alle Drucker unterstützen mindestens eine dieser Schnittstellen:

- *Seriell* angeschlossene Drucker werden über eine serielle Schnittstelle (auch RS-232 oder COM-Schnittstelle genannt) mit Ihrem Computer verbunden. Diese Schnittstelle wird von vielen unterschiedlichen Systemen verwendet. Serielle Kabel sind leicht erhältlich und können auch einfach selbst hergestellt werden. Einige Drucker erfordern möglicherweise ein spezielles Kabel oder besondere Kommunikationseinstellungen. Die meisten seriellen Schnittstellen von PCs besitzen eine maximale Datenübertragungsrate von 115200 bps; zum Ausdruck großer Grafiken sind serielle Drucker daher ungeeignet.
-

Parallel angeschlossene Drucker werden über eine parallele Schnittstelle mit Ihrem Computer verbunden. Diese Schnittstelle wird hauptsächlich von PCs und Workstations benutzt. Die Schnittstelle bietet eine höhere Datenübertragungsrate als serielle Schnittstellen. Kabel sind leicht erhältlich, sind aber vergleichsweise schwer selbst herzustellen. Üblicherweise brauchen keine Kommunikationsparameter festgelegt zu werden; dies macht die Einrichtung sehr einfach.

Die parallele Schnittstelle wird auch als “Centronics”-Schnittstelle bezeichnet, nach dem Namen des Steckverbinders, der hier häufig zum Einsatz kommt.

- USB-Schnittstelle (Universal Serial Bus) bieten noch höhere Geschwindigkeiten als parallele Schnittstellen oder serielle RS-232-Schnittstellen. USB-Kabel sind einfach und billig. Zum Drucken ist die USB-Schnittstelle besser geeignet als serielle oder parallele Schnittstellen, auf vielen UNIX Systemen werden USB-Schnittstellen jedoch nur unzureichend unterstützt. Um Probleme zu vermeiden, sollten Sie sich einen Drucker anschaffen, der sowohl eine USB-Schnittstelle als auch eine parallele Schnittstelle besitzt (viele Drucker besitzen heute beide Schnittstellen).

Im Allgemeinen versenden parallele Schnittstellen Daten nur in eine Richtung (vom Computer zum Drucker), serielle Schnittstellen und USB-Schnittstellen versenden Daten in beide Richtungen. Moderne parallele Schnittstellen (EPP and ECP) übertragen Daten bi-direktional nach dem Standard IEEE 1284.

Ein Drucker kann auf zwei Arten bi-direktional angesprochen werden. Die erste Methode benutzt einen Druckertreiber, der die herstellerspezifische Sprache des Druckers beherrscht. Diese Methode wird oft mit Tintenstrahl-Druckern eingesetzt und dazu benutzt, den Füllstand der Tintenpatronen und andere Status-Informationen auszugeben. Die zweite Methode wird benutzt, wenn der Drucker PostScript beherrscht.

Da ein PostScript-Druckauftrag ein komplettes Programm ist, kann es auch Daten an den Computer zurückliefern, ohne überhaupt eine Seite Papier zu bedrucken. Auf diesem Wege werden auch Probleme wie z.B. ein Papierstau vom Drucker an den Computer übermittelt. Darüberhinaus ist dies die effektivste Methode, um die tatsächlich gedruckte Anzahl an Seiten vom Drucker abzufragen: ein PostScript-Programm ermittelt jeweils vor und direkt nach einem Druckauftrag den Seitenzähler des Druckers, und vergleicht die beiden Zählerwerte.

10.3.1.1.2. Parallele Schnittstellen

Um einen Drucker mit paralleler Schnittstelle an Ihren Computer anzuschließen, verbinden Sie den Drucker mit einer parallelen Schnittstelle Ihres Computers. Die Dokumentation zu Ihrem Drucker oder Computer sollte Ihnen hier weiterhelfen.

Notieren Sie sich, mit welcher parallelen Schnittstelle des Computers Sie den Drucker verbunden haben. Die meisten Computer haben lediglich eine parallele Schnittstelle. Der FreeBSD-Gerätename der ersten Schnittstelle lautet `ppc0`, der der zweiten `ppc1`, und so weiter. Der Gerätename für den Drucker an der ersten parallelen Schnittstelle folgt dem selben Schema und lautet `/dev/lpt0`, usw.

10.3.1.1.3. Serielle Schnittstellen

Um einen Drucker mit serieller Schnittstelle an Ihren Computer anzuschließen, verbinden Sie den Drucker mit einer seriellen Schnittstelle Ihres Computers. Die Dokumentation zu Ihrem Drucker oder Computer sollte Ihnen hier weiterhelfen.

Sollten Sie sich nicht sicher sein, welches das “richtige Kabel” ist, können Sie eine dieser Alternativen ausprobieren:

- Ein *Modemkabel* verbindet alle Anschlüsse an einem Ende des Kabels eins-zu-eins mit den Anschlüssen am anderen Ende des Kabels. Ein solches Kabel wird auch als (engl.) “DTE-to-DCE-” oder “DEE-zu-DÜE-”Kabel bezeichnet.
- Ein *Nullmodemkabel* verbindet einige Signale eins-zu-eins, andere über Kreuz (z.B. Sende- und Empfangsleitung), und verbindet einige weitere direkt im Stecker miteinander. Ein solches Kabel wird auch als (engl.) “DTE-to-DTE-” oder “DEE-zu-DEE-”Kabel bezeichnet.
- Ein *Serielles Druckerkabel* schließlich, das für einige spezielle Drucker benötigt wird, verbindet zusätzliche Signale miteinander, anstatt sie im Stecker zurückzuführen.

Sie sollten auch die Kommunikationsparameter am Drucker einstellen; üblicherweise gibt es dazu DIP-Schalter, oder eine Option in der Menüführung am Drucker. Wählen Sie die höchste *Bitrate* (auch als `bps` Bits pro Sekunde oder *Baudrate* bezeichnet), die sowohl Ihr Drucker als auch Ihr Computer unterstützen. Wählen Sie 7 oder 8 Bits, gerade, ungerade oder keine Parität, und ein oder zwei Stoppbits. Wählen Sie die Art der Flusssteuerung: keine, XON/XOFF (auch als “in-band-” oder “Software”-Flusssteuerung bezeichnet), oder Hardware. Notieren Sie sich diese Einstellungen, damit Sie sie später bei der Software-Konfiguration zur Verfügung haben.

10.3.1.2. Software-Konfiguration

Dieser Abschnitt beschreibt die notwendigen Konfigurationsschritte, damit Sie mit dem FreeBSD-**LPD**-System drucken können.

Diese Schritte müssen Sie ausführen:

1. Konfigurieren Sie Ihren Kernel, soweit notwendig, um die Schnittstelle benutzen zu können, an die Ihr Drucker angeschlossen ist. Abschnitt Kernel-Konfiguration erklärt, welche Optionen Sie benötigen.
2. Konfigurieren Sie die Kommunikationseinstellungen für die parallele Schnittstelle, sofern Sie sie benutzen. Abschnitt Kommunikationseinstellungen für die parallele Schnittstelle enthält die Details.
3. Prüfen Sie, ob Sie Daten an den Drucker senden können. Abschnitt Prüfen der Drucker-Kommunikation führt eine Reihe von Möglichkeiten auf.
4. Konfigurieren Sie **LPD** für Ihren Drucker, indem Sie die Konfigurationsdatei `/etc/printcap` anpassen. Details dazu finden Sie im Abschnitt **LPD** aktivieren: die `/etc/printcap`-Datei.

10.3.1.2.1. Kernel-Konfiguration

Der Betriebssystem-Kernel ist für eine bestimmte Kombination aus Geräten kompiliert. Dies schließt Ihre seriellen oder parallelen Schnittstellen mit ein. Dementsprechend kann es notwendig sein, die Kernelkonfiguration um weitere Schnittstellen zu erweitern.

So können Sie prüfen, ob Ihr Kernel die serielle Schnittstelle unterstützt, an die Sie den Drucker angeschlossen haben:

```
# grep sioN /var/run/dmesg.boot
```

Ersetzen Sie `N` durch die Nummer der seriellen Schnittstelle, beginnend bei Null. Wenn Sie eine Ausgabe ähnlich der folgenden erhalten, unterstützt ihr Kernel diese Schnittstelle:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

Erhalten Sie keine Ausgabe, oder eine Fehlermeldung, wird die Schnittstelle nicht korrekt unterstützt.

So können Sie prüfen, ob Ihr Kernel die parallele Schnittstelle unterstützt, an die Sie den Drucker angeschlossen haben:

```
# grep ppcN /var/run/dmesg.boot
```

Ersetzen Sie *N* durch die Nummer der parallelen Schnittstelle, beginnend bei Null. Wenn Sie eine Ausgabe ähnlich der folgenden erhalten, unterstützt ihr Kernel diese Schnittstelle:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

Erhalten Sie keine Ausgabe, oder eine Fehlermeldung, wird die Schnittstelle nicht korrekt unterstützt.

Gegebenenfalls müssen Sie Ihren Kernel umkonfigurieren und neu kompilieren, damit die von Ihnen gewählte Schnittstelle unterstützt wird.

Um Unterstützung für eine serielle Schnittstelle hinzuzufügen, lesen Sie bitte Kapitel Konfiguration des FreeBSD Kernels. Um eine parallele Schnittstelle hinzuzufügen, lesen Sie bitte ebenfalls jenes Kapitel als auch den folgenden Abschnitt Kommunikationseinstellungen für die parallele Schnittstelle.

10.3.1.3. Kommunikationseinstellungen für die parallele Schnittstelle

Wenn Sie die parallele Schnittstelle zur Kommunikation mit Ihrem Drucker benutzen, haben Sie die Wahl zwischen Interrupt-gesteuerter oder Polling-Datenübertragung. Der generische Druckergerätetreiber lpt(4) in FreeBSD benutzt das ppbus(4)-System, das die parallele Schnittstelle mithilfe des ppc(4)-Treibers steuert.

- Die *Interrupt-gesteuerte* Datenübertragung ist die Voreinstellung im GENERIC-Kernel. Der Treiber benutzt eine IRQ-Leitung, um zu erfahren, wann der Drucker weitere Daten empfangen kann.
- Bei der *Polling-Methode* prüft der Treiber in regelmäßigen Abständen, ob weitere Daten übertragen werden können.

Die Interrupt-gesteuerte Methode ist überlicherweise schneller und verbraucht weniger Rechenzeit als die Polling-Methode, es wird jedoch eine eigene IRQ-Leitung für die Schnittstelle benötigt. Darüberhinaus kann es mit einigen Druckermodellen zu Problemen kommen, wenn die Interrupt-gesteuerte Übertragung zum Einsatz kommt.

Die können die Kommunikationseinstellung entweder in der Kernel-Konfiguration wählen, oder mittels des lptcontrol(8)-Programms zur Laufzeit einstellen.

So legen Sie die Kommunikationseinstellung in der Kernel-Konfiguration fest:

1. Ändern Sie Ihre Kernel-Konfigurationsdatei. Finden Sie die Zeile, die mit `device ppc0` beginnt. Wenn Sie die zweite parallele Schnittstelle konfigurieren möchten, suchen Sie nach `device ppc1`, für die dritte Schnittstelle `ppc2`, usw.
   - Um die Interrupt-Steuerung zu aktivieren, passen Sie die folgende Zeile an:

     `hint.ppc.0.irq="N"`

Ersetzen Sie *N* durch die Nummer der IRQ-Leitung, die dieser parallelen Schnittstelle zugewiesen ist. Stellen Sie sicher, dass Ihre Kernel-Konfigurationsdatei den ppc(4)-Treiber enthält:

```
device ppc
```

- Wenn Sie den Polling-Modus verwenden möchten, entfernen Sie die folgende Zeile aus `/boot/device.hints`:

  ```
  hint.ppc.0.irq="N"
  ```

  Sollte der Treiber die Schnittstelle dennoch im Interrupt-Modus betreiben, könnte dies an der Aktivierung durch das acpi(4)-System in FreeBSD liegen. Bitte prüfen Sie die ACPI- und die BIOS-Konfiguration.

2. Wenn Sie Ihre Kernel-Konfigurationsdatei angepasst haben, kompilieren und installieren Sie nun einen neuen Kernel. Das Kapitel Konfiguration des FreeBSD-Kernels enthält weitere Details dazu.

*So können Sie die Kommunikationseinstellung mit*lptcontrol(8) *ändern:*

1. Um die Interrupt-Steuerung für die Schnittstelle *N* zu aktivieren, geben Sie ein:

   ```
   # lptcontrol -i -d /dev/lptN
   ```

2. Um den Polling-Modus für die Schnittstelle *N* zu aktivieren, geben Sie ein:

   ```
   # lptcontrol -p -d /dev/lptN
   ```

Sie können diesen Befehl in `/etc/rc.local` aufnehmen, damit er bei jedem Systemstart automatisch ausgeführt wird. lptcontrol(8) enthält weitere Informationen.

10.3.1.4. Kommunikation mit den Drucker prüfen

Bevor Sie mit der Konfiguration des **LPD**-Drucksystems fortfahren, sollten Sie sicherstellen, das Sie erfolgreich Daten an Ihren Drucker senden können. Es ist deutlich einfacher, Kommunikations- und Konfigurationsprobleme unabhängig voneinander zu lösen.

Der Drucker kann mit einem Probeausdruck getestet werden. Für alle Drucker, die normalen Text unmittelbar drucken können, bietet sich das Programm lptest(1) an: es produziert alle 96 druckbaren ASCII-Zeichen auf 96 Zeilen.

Für einen PostScript-Drucker (oder andere Drucker, die eine Seitenbeschreibungssprache verwenden) muss ein passendes Programm an den Drucker gesendet werden, z.B. dieses:

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
(Funktioniert dieser Drucker?) show
showpage
```

Sie können dieses ProstScript-Programm in einer Datei speichern, und mit den Beispielen in den folgenden Abschnitt verwenden.

> **Anmerkung:** Nicht alle Drucker, die eine Seitenbeschreibungssprache verwenden, benötigen ein Test-Programm: z.B. HPs PCL (das auch in vielen kompatiblen Druckern zum Einsatz zu kommt), versteht normalen Text. Besondere Escape-Sequenzen werden benutzt, um die erweiterten Möglichkeiten aufzurufen. PostScript-Drucker können in der Regel keinen normalen Text direkt verarbeiten, weil sie ein PostScript-Programm erwarten, das eine Seite produziert.

10.3.1.4.1. Einen Paralleldrucker prüfen

Dieser Abschnitt führt vor, wie Sie die Kommunikation mit Ihrem Drucker über die parallele Schnittstelle prüfen können.

So testen Sie einen Drucker an einer parallelen Schnittstelle:

1. Werden Sie `root` mithilfe des su(1)-Befehls.
2. Senden Sie Testdaten an den Drucker.
   - Wenn Ihr Drucker reinen Text direkt drucken kann, verwenden Sie lptest(1):

     ```
     # lptest > /dev/lptN
     ```

     Ersetzen Sie *`N`* durch die Nummer der parallelen Schnittstelle, an die der Drucker angeschlossen ist (angefangen bei Null).
   - Wenn Ihr Drucker PostScript (oder eine andere Seitenbeschreibungssprache) versteht, senden Sie ein passendes Testprogramm an den Drucker. Geben Sie folgenden Befehl ein:

     ```
     # cat file > /dev/lptN
     ```

     Ersetzen Sie *`N`* durch die Nummer der parallelen Schnittstelle, an die der Drucker angeschlossen ist (angefangen bei Null). Geben Sie nun das Testprogramm ein, Zeile für Zeile. Kontrollieren Sie jede Zeile, bevor Sie die Eingabetaste drücken: Sie können die Zeile später nicht mehr ändern. Zum Schluss tippen Sie **Ctrl**+**D**. Wenn Sie ein anderes Zeichen nutzen, um das Ende der Datei anzuzeigen, müssen Sie natürlich die entsprechende Tastenkombination für dieses Zeichen betätigen.

     Sie können das Testprogramm auch in einer Datei speichern, und dann diesen Befehl aufrufen:

     ```
     # cat Testprogramm > /dev/lptN
     ```

     Ersetzen Sie *`Testprogramm`* durch den Dateinamen, unter dem Sie das Testprogramm gespeichert haben.

Der Drucker sollte einige Zeilen oder eine Seite drucken. Machen Sie sich keine Sorgen über falsche Formatierungen: die Software-Konfiguration enthält Informationen zum Umformatieren von Druckaufträgen.

10.3.1.4.2. Einen seriellen Drucker prüfen

Dieser Abschnitt führt vor, wie Sie die Kommunikation mit Ihrem Drucker über die parallele Schnittstelle prüfen können.

So testen Sie einen Drucker an einer seriellen Schnittstelle:

1. Werden Sie `root` mithilfe des su(1)-Befehls.
2. Ändern Sie die Datei `/etc/remote`. Fügen Sie den folgenden Eintrag hinzu:

   ```
   printer:dv=/dev/port:br#bps-rate:pa=parity
   ```

   Ersetzen Sie *`Gerät`* durch den Gerätenamen der seriellen Schnittstelle (`ttyu0` für die erste, `ttyu1` für die zweite, usw.), *`Baudrate`* ist die Geschwindigkeit und *`Parität`* die Parität (`even` für gerade, `odd` für ungerade oder `none` für keine), die Sie am Drucker eingestellt haben.

Hier ein Beispieleintrag für einen Drucker, der über die dritte serielle Schnittstelle angeschlossen ist, mit 19.200 Baud kommuniziert und keine Parität verwendet:

```
printer:dv=/dev/ttyu2:br#19200:pa=none
```

3. Verbinden Sie sich mit dem Drucker über den Befehl tip(1):

   ```
   # tip printer
   ```

   Kommt es hierbei zu einer Fehlermeldung, ändern Sie den Eintrag in `/dev/cuaaN` und verwenden Sie `/dev/cuaaN` statt `/dev/ttyuN`.

4. Senden Sie Testdaten an den Drucker.

   - Wenn Ihr Drucker reinen Text direkt drucken kann, verwenden Sie lptest(1):

     ```
     % $lptest
     ```

   - Wenn Ihr Drucker PostScript (oder eine andere Seitenbeschreibungssprache) versteht, senden Sie ein passendes Testprogramm an den Drucker. Geben Sie das Testprogramm ein, Zeile für Zeile. Kontrollieren Sie jede Zeile, bevor Sie die Eingabetaste drücken: Sie können die Zeile später nicht mehr ändern. Zum Schluss tippen Sie `Control-D`.

     Sie können das Testprogramm auch in einer Datei speichern, und dann diesen Befehl aufrufen:

     ```
     % >Testprogramm
     ```

     Ersetzen Sie *`Testprogramm`* durch den Dateinamen, unter dem Sie das Testprogramm gespeichert haben. Nachdem tip(1) die Datei gesendet hat, tippen Sie `Control-D`.

Der Drucker sollte einige Zeilen oder eine Seite drucken. Machen Sie sich keine Sorgen über falsche Formatierungen: die Software-Konfiguration enthält Informationen zum Umformatieren von Druckaufträgen.

10.3.1.5. LPD aktivieren: die `/etc/printcap`-Datei

Nachdem Sie Ihren Drucker angeschlossen haben, Ihren Kernel richtig konfiguriert haben und erfolgreich einen Testausdruck produziert haben, können Sie nun das **LPD**-System konfigurieren.

Sie konfigurieren **LPD**, indem Sie die Datei `/etc/printcap` anpassen. Da **LPD** die Datei jedes Mal liest, wenn eine Aktion durchgeführt wird, werden Änderungen an der Konfiguration sofort aktiv.

Die printcap(5)-Datei ist einfach aufgebaut. Sie können `/etc/printcap` mit Ihrem bevorzugten Texteditor bearbeiten. Sie verwendet dasselbe Format wie auch `/usr/share/misc/termcap` oder `/etc/remote`. Informationen zum Format finden Sie in cgetent(3).

Die Grund-Konfiguration des **LPD**-Systems beinhaltet diese Schritte:

1. Wählen Sie einen Namen (und einige praktische Abkürzungen) für die Druckerwarteschlage, und tragen Sie ihn in die `/etc/printcap`-Datei ein. Abschnitt Einen Namen wählen enthält weitere Informationen.
2.

   Schalten Sie den Druck von Deckblättern aus (dies ist standardmäßig eingeschaltet), indem Sie das Attribut `sh` setzen. Abschnitt Den Druck von Deckblättern ausschalten erklärt, wie Sie dies tun können.
3. Legen Sie ein Pufferverzeichnis für die Warteschlange an, und geben Sie den Pfad mittels des `sd`-Attributs an: siehe Abschnitt Das Pufferverzeichnis anlegen.

4. Geben Sie den Gerätenamen für Ihren Drucker mittels des `lp`-Attributs an: siehe Abschnitt Festlegen der Drucker-Gerätedatei. Ist Ihr Drucker über eine serielle Schnittstelle angeschlossen, benutzen Sie das Attribut `ms#`, wie dies in Abschnitt Festlegen der Kommunikationsparameter beschrieben ist.
5. Installieren Sie einen Filter für reinen Text: siehe Abschnitt Den Textfilter installieren.
6. Testen Sie die Konfiguration, indem Sie etwas mit dem lpr(1)-Befehl drucken. Die Abschnitte Die Konfiguration testen und Fehlersuche und Problembehebung enthalten weitere Informationen.

> **Anmerkung:** Drucker, die eine Seitenbeschreibungssprache wie PostScript verwenden, können keinen reinen Text drucken. Es wird deshalb angenommen, dass Sie nur solche Dateien drucken, die Ihr Drucker verarbeiten kann.

Viele Anwender erwarten, dass sie normalen Text auf allen Druckern drucken können. Viele Programme, die mit **LPD** zusammenarbeiten, gehen ebenfalls von dieser Annahme aus. Wenn Sie einen PostScript-Drucker installieren, und Sie sowohl PostScript- als auch Textdateien drucken möchten, sollten Sie einen weiteren Konfigurationsschritt ausführen und einen Text-zu-PostScript-Filter installieren. Der Abschnitt Drucken von reinen Textdateien auf einem PostScript-Drucker erklärt, wie Sie dies tun können.

10.3.1.5.1. Einen Namen wählen

Der erste einfache Schritt ist, einen Namen für Ihren Drucker zu wählen. Sie können diesen Namen frei wählen, Sie sollten allerdings keine Sonderzeichen oder Umlaute verwenden. Sie können mehrere Alias-Namen vergeben.

Ein Drucker in `/etc/printcap` sollte den Alias `lp` haben. Dieser Name wird standardmäßig von allen Druckbefehlen verwendet, wenn auf der Befehlszeile oder in der `PRINTER`-Umgebungsvariablen kein anderer Drucker angegeben ist.

Ebenso ist es üblich, eine ausführliche Beschreibung des Druckermodells als letzten Alias-Namen zu verwenden.

Sobald Sie einen Namen und einige einfache Alias-Namen ausgewählt haben, tragen Sie sie in die Datei `/etc/printcap` ein. Beginnen Sie die Zeile mit dem Namen des Druckers und fügen Sie alle Alias-Namen an. Trennen Sie die Namen durch den senkrechten Strich `|`. Fügen Sie an das Ende der Zeile einen Doppelpunkt `:` an.

Das folgende Beispiel definiert zwei Drucker, einen Diablo 630 Zeilendrucker, und einen Panasonic KX-P4455 PostScript-Laserdrucker:

```
#
#  /etc/printcap for host rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

Der erste Drucker hat den Namen `rattan`, und hat die Alias-Namen `line`, `diablo`, `lp` und `Diablo 630 Line Printer`. Da er den Alias-Namen `lp` trägt, wird er standardmäßig von den Druckprogrammen verwendet. Der zweite Drucker heißt `bamboo`, und hat die Alias-Namen `ps`, `PS`, `S`, `panasonic` und `Panasonic KX-P4455 PostScript v51.4`.

10.3.1.5.2. Keine Deckblätter drucken

Standardmäßig druckt das **LPD**-System ein *Deckblatt* vor jedem Druckauftrag, die den Namen des Druckauftrags, den Benutzer und den Computer angibt. Während der Einrichtung des Systems und beim Testen stört das Deckblatt allerdings, weshalb Sie sie zunächst deaktivieren sollten.

Um den Druck von Deckblättern zu deaktivieren, fügen Sie das Attribut `sh` zur Druckerdefinition in `/etc/printcap` hinzu. Hier ein Beispiel:

```
#
#  /etc/printcap for host rose - no header pages anywhere
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:
```

Beachten Sie die korrekte Formatierung: die beiden Definitionen beginnen auf einer Zeile; weitere Zeilen der Definition sind mit einem `Tab`-Zeichen eingerückt, und alle Zeilen einer Definition, bis auf die letzte, enden mit dem Backslash `\`.

10.3.1.5.3. Das Pufferverzeichnis anlegen

Der nächste Schritt ist, das *Pufferverzeichnis* anzulegen. In diesem Verzeichnis werden Druckaufträge zwischengespeichert, während sie gedruckt werden. Gleichzeitig werden hier auch einige Verwaltungsdateien des Systems abgelegt.

Da sich die Dateien in diesem Verzeichnis häufig ändern, ist es üblich, das Verzeichnis unter `/var/spool` anzulegen. Es ist nicht notwendig, Sicherungskopien der Dateien herzustellen; das Verzeichnis kann nötigenfalls leicht mit mkdir(1) wieder angelegt werden.

Es ist auch üblich, dem Verzeichnis denselben Namen wie dem Drucker zu geben:

```
# mkdir /var/spool/printer-name
```

Wenn Sie viele Drucker verwenden, ist es am besten, wenn Sie für die Pufferverzeichnisse ein eigenes Unterverzeichnis in `/var/spool` anlegen, wie dies hier für die beiden Beispieldrucker `rattan` und `bamboo` gezeigt wird:

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```

> **Anmerkung:** Um zu verhindern, dass alle Benutzer den Inhalt aller Druckaufträge einsehen können, sollten Sie die Rechte auf den Pufferverzeichnissen einschränken. Die Verzeichnisse sollten dem Benutzer `daemon` und der Gruppe `daemon` gehören, und auch nur vom Benutzer und der Gruppe les-, schreib- und durchsuchbar sein. Für unsere Beispieldrucker:
>
> ```
> # chown daemon:daemon /var/spool/lpd/rattan
> # chown daemon:daemon /var/spool/lpd/bamboo
> # chmod 770 /var/spool/lpd/rattan
> # chmod 770 /var/spool/lpd/bamboo
> ```

Schließlich müssen Sie dem **LPD**-System noch mitteilen, wo Sie die Pufferverzeichnisse angelegt haben. Dazu geben Sie in der Definition das Attribut `sd` an:

```
#
#  /etc/printcap for host rose - added spooling directories
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:
```

Beachten Sie, dass der Druckername in der ersten Spalte beginnt, und dass alle Folgezeilen mit einem `Tab` eingerückt sind.

Wenn Sie das `sd`-Attribut nicht angeben, verwendet das System `/var/spool/lpd` als Verzeichnis.

10.3.1.5.4. Festlegen der Drucker-Gerätedatei

Nachdem Sie die korrekte Gerätedatei für die Schnittstelle im Abschnitt Hardware-Konfiguration identifiziert und angelegt haben, müssen Sie dem **LPD**-System mitteilen, welche Gerätedatei im Verzeichnis `/dev` es für die Datenübertragung zum Drucker verwenden soll.

Geben Sie die Gerätedatei durch das Attribut `lp` in `/etc/printcap` an.

Wenn `rattan` an die erste parallele Schnittstelle angeschlossen ist, und `bamboo` an die sechste serielle, dann sieht `/etc/printcap` so aus:

```
#
#  /etc/printcap for host rose - identified what devices to use
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:
```

Wenn Sie `lp` nicht angeben, versucht **LPD** die Gerätedatei `/dev/lp` zu verwenden. `/dev/lp` ist zurzeit in FreeBSD nicht definiert.

Wenn Ihr Drucker über eine parallele Schnittstelle angeschlossen ist, können Sie mit dem Abschnitt Den Textfilter installieren fortfahren. Verwenden Sie eine serielle Schnittstelle, beachten Sie bitte den folgenden Abschnitt.

10.3.1.5.5. Kommunikationsparameter festlegen

Für seriell angeschlossene Drucker kann **LPD** die Geschwindigkeit, Parität und weitere Kommunikationsparameter einstellen. Dies hat folgende Vorteile:

- Sie können die Parameter einfach in `/etc/printcap` ändern, ohne das Ausgabe-Filterprogramm anpassen zu müssen.
- Dasselbe Ausgabe-Filterprogramm kann für unterschiedliche Drucker verwendet werden, auch wenn diese unterschiedliche Kommunikationseinstellungen benötigen.

Die folgenden Attribute legen die seriellen Kommunikationsparameter fest:

`br#`*`Baudrate`*

Setzt die Übertragungsgeschwindigkeit auf *`Baudrate`*. *`Baudrate`* kann üblicherweise 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600, oder 115200 Bit pro Sekunde betragen.

`ms#`*`stty-Modi`*

Setzt die Eigenschaften für das Gerät, nachdem es geöffnet wurde. Die verfügbaren Eigenschaften sind in stty(1) aufgeführt.

Wenn **LPD** das mit `lp` angegebene Gerät öffnet, setzt es die mit `ms#` angegebenen Eigenschaften. Von besonderem Interesse sind hier die Modi `parenb`, `parodd`, `cs5`, `cs6`, `cs7`, `cs8`, `cstopb`, `crtscts` und `ixon`, die in der stty(1)-Handbuchseite erläutert werden.

Für den über die sechste serielle Schnittstelle angeschlossenen Laserdrucker beträgt die Geschwindigkeit 38.400 Baud, und es werden diese Kommunikationseinstellungen verwendet: keine Parität (`-parenb`), 8-Bit-Zeichen (`cs8`), keine Modemsteuerung (`clocal`) und Hardware-Flusssteuerung (`crtscts`):

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:
```

10.3.1.5.6. Den Textfilter installieren

Ein *Textfilter*, auch als *Eingangsfilter* bezeichnet, ist ein Programm, das von **LPD** aufgerufen wird, wenn ein Druckauftrag verarbeitet wird. Dabei wird die Standardeingabe des Programms mit der zu druckenden Datei verbunden, und die Standardausgabe mit dem im `lp`-Attribut angegebenen Gerät. Das Programm sollte nun die Datei einlesen, alle Übersetzungen durchführen, die für den Drucker notwendig sind, und das Ergebnis über die Standardausgabe an den Drucker senden. Textfilter werden im Abschnitt Filter genauer erläutert.

Um einen einfachen Test durchzuführen, reicht ein kleines Filterprogramm, dass schlicht `/bin/cat` aufruft, um die Daten unverändert und den Drucker zu schicken. FreeBSD verfügt über das Programm `lpf`, das Unterstreichung und Fettdruck für solche Drucker ermöglicht, die ansonsten dazu nicht in der Lage wären. Darüberhinaus gibt es viele andere Filter, die Sie einsetzen können. `lpf` wird im Abschnitt lpf: ein Textfilter ausführlich beschrieben.

Legen Sie zunächst das folgende Shell-Skript als `/usr/local/libexec/if-simple` mit Ihrem bevorzugten Texteditor an:

```
#!/bin/sh
#
# if-simple - Simple text input filter for lpd
# Installed in /usr/local/libexec/if-simple
#
# Simply copies stdin to stdout.  Ignores all filter arguments.

/bin/cat && exit 0
```

```
exit 2
```

Machen Sie die Datei ausführbar:

```
# chmod 555 /usr/local/libexec/if-simple
```

Konfigurieren Sie nun den Textfilter für Ihren Drucker in `/etc/printcap`, indem Sie das `if`-Attribut hinzufügen. Hier die Konfiguration unserer beiden Beispieldrucker:

```
#
#  /etc/printcap for host rose - added text filter
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:\
        :if=/usr/local/libexec/if-simple:
```

Anmerkung: Das Shell-Skript `if-simple` steht im Verzeichnis `/usr/share/examples/printing`.

*10.3.1.5.7. **LPD** aktivieren*

lpd(8) wird von `/etc/rc` gestartet, wenn die rc.conf(5)-Variable `lpd_enable` auf `YES` gesetzt ist. Fügen Sie dazu diese Zeile in `/etc/rc.conf` hinzu:

```
lpd_enable="YES"
```

Starten Sie Ihren Computer neu, oder starten Sie lpd(8) von Hand:

```
# lpd
```

10.3.1.5.8. Die Konfiguration testen

Damit ist die einfache Konfiguration abgeschlossen. Noch muss die Konfiguration aber getestet werden und etwaige Probleme müssen behoben werden. Um die Konfiguration zu testen, sollten Sie einen Probeausdruck mithilfe des Programms lpr(1) produzieren. lpr(1) übergibt Druckaufträge an das **LPD**-System.

Sie können lpr(1) mit lptest(1) kombinieren, um Testdaten zu drucken. lptest(1) wurde im Abschnitt Kommunikation mit den Drucker prüfen vorgestellt.

*So testen Sie die einfache **LPD**-Konfiguration:*

```
# lptest 20 5 | lpr -PDruckername
```

Ersetzen Sie *`Druckername`* durch den Namen des Druckers, den Sie testen möchten. Wenn Sie den Standard-Drucker testen möchten, rufen Sie lpr(1) ohne die Option `-P` auf. Wenn Sie einen Drucker testen möchten,

der nur PostScript versteht, müssen Sie ein PostScript-Testprogramm an lpr(1) übergeben. Ein Testprogramm, das Sie in einer Datei gespeichert haben, können Sie mit dem Befehl `lpr` *`Dateiname`* an das **LPD**-System übergeben.

Bei einem PostScript-Drucker hängt das Ergebnis naturgemäß vom Testprogramm ab. Wenn Sie lptest(1) verwenden, sollte das Ergebnis ungefähr so aussehen:

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

Um sicherzustellen, dass alles richtig funktioniert, sollten Sie jetzt ein größeres PostScript-Programm senden. Mit lptest(1) können Sie größere Datenmengen z.B. mit dem Befehl `lptest 80 60` erzeugen: lptest(1) produziert 60 Zeilen mit je 80 Zeichen.

Wenn Sie nicht erfolgreich drucken können, finden Sie im Abschnitt Fehlersuche und Problembehebung weitere Informationen.

10.4. Erweiterte Drucker-Konfiguration

Übersetzt von Johann Kois.

Warnung: Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttyd`*`N`* in `/dev/ttyu`*`N`* umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

Dieser Abschnitt beschreibt den Einsatz von Filtern für das Drucken speziell formatierter Seiten oder von Deckblättern, das Drucken über ein Netzwerk sowie die Beschränkung und Verrechnung der Druckernutzung.

10.4.1. Filter

Obwohl **LPD** Netzwerkprotokolle, Warteschlangen, Zugriffskontrollen und andere für das Drucken wichtige Aspekte prinzipiell unterstützt, passiert ein Großteil der *wirklichen* Arbeit in den sogenannten *Filtern*. Dabei handelt es sich um Programme, die direkt mit einem Drucker kommunizieren und deren Gerätespezifika und spezielle Anforderungen erfüllen. Im einfachsten Fall installiert man nur einen reinen Textfilter, der mit beinahe allen Druckern funktionieren sollte. (Lesen Sie dazu auch den Abschnitt Den Text-Filter installieren.)

Um die erweiterten Fähigkeiten von Druckern auch einsetzen zu können, sollten Sie verstehen, wie Filter arbeiten, da diese für die Bereitstellung dieser Funktionen zuständig sind. Die schlechte Nachricht ist, dass *Sie* diese Filter bereitstellen müssen. Die gute Nachricht ist allerdings, dass diese in der Regel bereits vorhanden sind. Ist dies nicht der Fall, können Sie einen Filter meist relativ einfach selbst erstellen.

Der Filter `/usr/libexec/lpr/lpf` wird bereits mit FreeBSD geliefert. Er kümmert sich um die korrekte Behandlung von gelöschten Zeichen (das sogenannte *Backspacing*), um im Text enthaltene Tabulatoren, sowie um die Verrechnung von Druckaufträgen. Das ist aber auch alles, was dieser Filter kann. Zusätzliche Filter und für die Funktion von Filtern nötige Komponenten finden sich aber in der FreeBSD Ports-Sammlung.

Dieser Abschnitt behandelt folgende Themen:

- Der Abschnitt Die Funktionsweise von Filtern versucht, einen Überblick über die Rolle von Filtern innerhalb des Druckprozesses zu geben. Sie sollten diesen Abschnitt lesen, damit Sie verstehen, was “unter der Haube” passiert, wenn **LPD** einen Filter verwendet. Dieses Wissen wird Ihnen dabei helfen, Probleme, die bei Installation von Filtern für verschiedene Drucker entstehen können, vorauszusehen und zu beheben.
- **LPD** geht davon aus, dass jeder Drucker in der Lage ist, normalen Text zu drucken. Für PostScript- (oder andere sprachbasierte) Drucker stellt dies allerdings ein Problem dar, da diese nicht in der Lage sind, normalen Text direkt zu drucken. Der Abschnitt Normalen Text auf PostScript-Druckern drucken beschreibt, wie Sie dieses Problem lösen können. Besitzen Sie einen PostScript-Drucker, sollten Sie diesen Abschnitt lesen.
- PostScript ist ein populäres Ausgabeformat, das von vielen Programmen unterstützt wird. Es ist sogar möglich, PostScript-Code direkt zu schreiben. Leider sind PostScript-Drucker in der Regel relativ teuer. Der Abschnitt PostScript auf Nicht-PostScript-Druckern emulieren beschreibt, wie Sie einen Textfilter anpassen müssen, um PostScript-Daten auf einem *nicht-PostScript-fähigen Drucker* auszugeben. Haben Sie keinen PostScript-Drucker, sollten Sie insbesondere diesen Abschnitt lesen.
- Der Abschnitt Konvertierungsfilter beschreibt eine Möglichkeit zur automatischen Konvertierung verschiedener Dateiformate in ein von Ihrem Drucker unterstütztes Format. Nachdem Sie diesen Abschnitt gelesen haben, werden Sie in der Lage sein, Ihren Drucker so zu konfigurieren, dass Sie durch die Eingabe von `lpr -t` troff-Daten, von `lpr -d` TEX-DVI-Daten, oder von `lpr -v` Rasterbilddaten drucken können. Daher sollten Sie diesen Abschnitt auf jeden Fall lesen.
- Im Abschnitt Ausgabefilter wird eine nur selten genutzte Eigenschaft von **LPD**, die sogenannten Ausgabefilter, beschrieben. Wenn Sie keine Deckblätter drucken müssen, können Sie diesen Abschnitt überspringen.
- Der Abschnitt lpf: Ein Textfilter beschreibt `lpf`, einen kompletten, wenn auch einfachen Textfilter für Zeilendrucker (oder auch Laserdrucker, die sich analog verhalten), der bereits mit FreeBSD geliefert wird. Wenn Sie nur am Ausdruck von reinem Text interessiert sind, oder wenn Ihr Drucker nur “Schrott” produziert, wenn er auf Backspace-Zeichen trifft, sollten Sie sich `lpf` näher ansehen.

Anmerkung: Eine Kopie der verschiedenen Skripte finden Sie im Verzeichnis `/usr/share/examples/printing`.

10.4.1.1. Die Funktionsweise von Filtern

Bei einem Filter handelt es sich um ein ausführbares Programm, das von **LPD** gestartet wird, um den geräteabhängigen Teil der Kommunikation mit einem Drucker zu übernehmen.

Wenn **LPD** eine Datei über einen Druckauftrag drucken will, startet es ein Filterprogramm. Danach setzt es die Standardeingabe des Filters auf die zu druckende Datei, die Standardausgabe auf den Drucker und die Standardfehlerausgabe auf `/dev/console` (Voreinstellung) oder auf die über die Option `lf` in `/etc/printcap` festgelegte Datei.

Welcher Filter von **LPD** mit welchen Argumenten geladen wird, wird in der Datei `/etc/printcap` oder durch die Argumente, die der Anwender lpr(1) auf der Kommandozeile übergibt, festgelegt. Gibt der Anwender beispielsweise `lpr -t` ein, startet **LPD** über die `tf`-Fähigkeit den troff-Filter für den gewünschten Drucker. Wollen Sie hingegen normalen Text drucken, wird der `if`-Filter gestartet. (Für Ausnahmen von diesem Vorgehen lesen Sie bitte den Abschnitt Ausgabefilter.)

Es gibt drei Arten von Filtern, die Sie in `/etc/printcap` angeben können:

- *Textfilter* (die in der **LPD**-Dokumentation als *input filter* bezeichnet werden) sind für den Druck von normalem Text zuständig. Es handelt sich dabei um eine Art Standardfilter, da **LPD** von jedem Drucker erwartet, dass er normalen Text drucken kann. Aufgabe des Textfilters ist es, sicherzustellen, dass gelöschte Zeichen (*Backspaces*), Tabulatoren und andere Sonderzeichen Ihren Drucker nicht verwirren. Falls Sie für die Nutzung eines Druckers bezahlen müssen, kann der Textfilter über die Anzahl der gedruckten Zeilen auch die Anzahl der von Ihnen gedruckten Seiten ermitteln. Der Textfilter wird mit folgenden Argumenten gestartet:

  `filter-name` [-c] -w *`width`* -l *`length`* -i *`indent`* -n *`login`* -h *`host acct-file`*

  Die einzelnen Argumente haben folgende Bedeutung:

  `-c`

  Notwendig, wenn `lpr -l` verwendet wird.

  `width`

  Der Wert der in `/etc/printcap` festgelegten Option `pw` (*page width*). In der Voreinstellung ist dieser Wert auf 132 gesetzt.

  `length`

  Der Wert der `pl`-Fähigkeit (*page length*), Voreinstellung 66.

  `indent`

  Der durch `lpr -i` festgelegte Einzug, Voreinstellung 0.

  `login`

  Der Name des Benutzers, der die Datei druckt.

  `host`

  Der Rechner, auf dem der Druckauftrag gestartet wurde.

  `acct-file`

  Der Name der Verrechnungsdatei, in der die Ergebnisse der `af`-Fähigkeit gespeichert werden.

- Ein *Konvertierungsfilter* konvertiert verschiedene Dateiformate in ein Format, das Ihr Drucker auf Papier ausgeben kann. So kann etwa der ditroff-Schriftsatz nicht direkt gedruckt werden, daher müssen Sie einen Konvertierungsfilter installieren, um diese Daten in ein Format zu bringen, das Ihr Drucker verarbeiten und drucken kann. Der Abschnitt Konvertierungsfilter enthält ausführliche Informationen zu diesen Filtern. Konvertierungsfilter können auch zur Verrechnung verwendet werden. Sie werden mit folgenden Argumenten gestartet:

  `filter-name` -x *`pixel-width`* -y *`pixel-height`* -n *`login`* -h *`host acct-file`*

  `pixel-width` ist der Wert der `px`-Fähigkeit (Voreinstellung 0), während *`pixel-height`* dem Wert der `py`-Fähigkeit (Voreinstellung ebenfalls 0) entspricht.

- *Ausgabefilter* werden nur verwendet, wenn keine Textfilter vorhanden sind oder wenn Deckblätter benötigt werden. Der Abschnitt Ausgabefilter enthält weitere Informationen. Ausgabefilter unterstützen nur zwei Argumente:

  `filter-name` -w *`width`* -l *`length`*

  Beide Argumente entsprechen den Optionen `-w` und `-l` der Textfilter.

Alle Filter sollten mit folgenden Rückgabewerten (Exitcodes) *beendet* werden:

exit 0

Der Filter hat die Datei erfolgreich gedruckt.

exit 1

Der Filter war nicht in der Lage, die Datei zu drucken und meldet diesen Exitcode an **LPD**, um die Datei erneut zu drucken. **LPD** startet daraufhin den Filter erneut.

exit 2

Der Filter war nicht in der Lage, die Datei zu drucken. Bei diesem Exitcode soll **LPD** aber nicht versuchen, die Datei erneut zu drucken, sondern den Druckauftrag verwerfen.

`/usr/libexec/lpr/lpf`, der mit FreeBSD gelieferte Textfilter, nutzt die Argumente *page width* und *page length*, um festzulegen, wann ein Seitenumbruch (*form feed*) gesendet werden soll sowie zur Verrechnung von Druckaufträgen. Dazu werden der Benutzername, der für den Druckauftrag verwendete Rechner sowie die Verrechnungsdatei ausgewertet, um die entsprechenden Einträge zu erstellen.

Wenn Sie auf der Suche nach Filtern sind, achten Sie darauf, dass diese LPD-kompatibel sind. Dazu müssen diese die oben beschriebenen Argumente unterstützen. Wenn Sie planen, Ihre Filter selbst zu erstellen, müssen diese ebenfalls die gleichen Argumente und Exitcodes unterstützen.

10.4.1.2. Normalen Text auf PostScript®-Druckern drucken

Sie sind der alleinige Benutzer Ihres Computers und Ihres PostScript-Druckers und Sie sind sich sicher, dass Sie niemals normalen Text an Ihren Drucker senden werden? Außerdem werden Sie niemals ein Programm verwenden, um normalen Text auszudrucken? Nur wenn dies alles zutrifft, können Sie diesen Abschnitt überspringen.

Wollen Sie allerdings sowohl PostScript als auch normalen Text drucken, müssen Sie Ihren Drucker zuvor entsprechend konfigurieren. Dazu muss Ihr Textfilter in der Lage sein, zu unterscheiden, ob es sich bei einem ankommenden Druckauftrag um normalen Text oder um PostScript-Daten handelt. Jeder PostScript-Druckauftrag muss mit den Zeichen `%!` beginnen (sehen Sie in Ihrem Druckerhandbuch nach, ob Ihr Drucker weitere Sprachen unterstützt). Sind dies die beiden ersten Zeichen eines Druckauftrages, so handelt es sich um PostScript-Daten, die direkt gedruckt werden können. Fehlen diese Zeichen allerdings, muss der Textfilter den Inhalt der Datei nach PostScript konvertieren, bevor die Datei gedruckt werden kann.

Wie funktioniert diese Unterscheidung?

Haben Sie einen seriellen Drucker, können Sie `lprps` installieren. `lprps` ist ein PostScript-Druckerfilter, der eine Zweiwegekommunikation mit einem Drucker ermöglicht. Er aktualisiert die Druckerstatusdatei mit Protokollinformationen des Druckers. Dadurch sind Anwender und Administratoren in der Lage, den genauen Zustand des Druckers zu prüfen (durch Meldungen wie `toner low` oder `paper jam`). Wichtiger ist allerdings, dass `lprps` `psif` enthält, ein Programm, das feststellen kann, ob ein ankommender Druckauftrag normalen Text enthält.

Ist dies der Fall, wird `textps` (das ebenfalls mit `lprps` geliefert wird) aufgerufen und die Datei nach PostScript konvertiert. Danach kann `lprps` die Datei an den Drucker senden.

`lprps` ist in der FreeBSD Ports-Sammlung enthalten. Je nach der von Ihnen verwendeten Papiergröße installieren Sie dazu den Port `print/lprps-a4` oder `print/lprps-letter`. Nach der Installation müssen Sie nur noch den Pfad zum Programm `psif` angeben, das als Teil von `lprps` installiert wird. Haben Sie `lprps` über die Ports-Sammlung installiert, fügen Sie folgende Zeile in den Eintrag Ihres PostScript-Druckers in `/etc/printcap` ein:

```
:if=/usr/local/libexec/psif:
```

Zusätzlich sollten Sie die `rw`-Fähigkeit aktivieren, um **LPD** im Schreib- und Lesemodus zu öffnen.

Haben Sie hingegen einen parallelen PostScript-Drucker, was eine Zweiwegekommunikation mit Ihrem Drucker (auf die `lprps` angewiesen ist) unmöglich macht, können Sie das folgende Shell-Skript verwenden:

```
#!/bin/sh
#
#  psif - Drucke PostScript oder normalen Text auf einem PostScript-Drucker
#  Script-Version; das ist NICHT die mit lprps gelieferte Version!
#  Installiert unter:  /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=‘expr "$first_line" : ’\(..\)’‘

if [ "$first_two_chars" = "%!" ]; then
    #
    #  PostScript - einfach drucken.
    #
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
else
    #
    #  Normaler Text - zuerst konvertieren, dann drucken.
    #
    ( echo "$first_line"; cat ) | /usr/local/bin/textps && printf "\004" && exit 0
    exit 2
fi
```

Für dieses Skript wurde `textps` als seperates Programm installiert, um normalen Text nach PostScript zu konvertieren. Sie können aber auch jeden anderen Text-nach-PostScript-Konverter verwenden. Die FreeBSD Ports-Sammlung enthält mit `a2ps` ein umfangreiches Programm zur Konvertierung von normalem Text nach PostScript.

10.4.1.3. PostScript auf Nicht-PostScript-Druckern emulieren

Bei PostScript handelt es sich um den *de facto*-Standard für hochwertigen Schriftsatz und Druck. Leider ist PostScript aber auch ein *teurer* Standard. Glücklicherweise hat Aladdin Enterprises daher eine freie PostScript-ähnliche Implementierung namens **Ghostscript** entwickelt, die auch unter FreeBSD lauffähig ist. **Ghostscript** kann fast jede PostScript-Datei lesen und auf den verschiedensten Geräten ausgeben, darunter auch auf

vielen Nicht-PostScript-Druckern. Durch die Installation von **Ghostscript** und die Nutzung eines speziellen Textfilters erreichen Sie, dass sich Ihr Nicht-PostScript-Drucker wie ein echter PostScript-Drucker verhält.

Ghostscript ist in verschiedenen Versionen in der FreeBSD Ports-Sammlung enhalten, die am häfigsten verwendete Version ist `print/ghostscript-gpl`.

Um PostScript zu emulieren, muss der Textfilter erkennen, ob er eine PostScript-Datei drucken soll. Ist dies nicht der Fall, wird die Datei direkt an den Drucker geschickt. Anderenfalls wird die Datei an **Ghostscript** übergeben, das die Datei in ein Format konvertiert, das Ihr Drucker versteht.

Dazu ein Beispiel. Das folgende Skript ist ein Textfilter für den Drucker DeskJet 500 von Hewlett Packard. Nutzen Sie einen anderen Drucker, müssen Sie die Option `-sDEVICE` beim Aufruf von `gs` (Ghostscript) entsprechend anpassen. Eine Liste der von **Ghostscript** unterstützten Geräte erhalten Sie durch die Eingabe von `gs -h` auf der Kommandozeile.

```
#!/bin/sh
#
#  ifhp - Ghostscript-emuliertes PostScript auf einem HP DeskJet 500 drucken
#  Installiert unter:  /usr/local/libexec/ifhp

#
#  LF als CR+LF behandeln (um einen "Treppeneffekt" auf HP/PCL-Drucker
#  zu vermeiden)
#
printf "\033&k2G" || exit 2

#
#  Lies die ersten zwei Zeichen der Datei
#
IFS="" read -r first_line
first_two_chars=‘expr "$first_line" : ’\(..\)’‘

if [ "$first_two_chars" = "%!" ]; then
    #
    #  Oh.  Es ist PostScript; mit Ghostscript konvertieren, danach drucken.
    #
    /usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
      -sOutputFile=- - && exit 0
else
    #
    #  Normaler Text oder HP/PCL, einfach direkt drucken.  Ans Ende setzen wir
    #  einen Seitenumbruch (also ein Form Feed), damit auch die letzte Seite
    #  ausgeworfen wird.
    #
    echo "$first_line" && cat && printf "\033&l0H" &&
exit 0
fi

exit 2
```

Zuletzt müssen Sie **LPD** noch durch die `if`-Fähigkeit über den neuen Filter informieren:

```
:if=/usr/local/libexec/ifhp:
```

Das ist alles. Ab sofort sollte sowohl ein `lpr` *`normaler.text`* als auch ein `lpr` *`wasauchimmer.ps`* funktionieren und beide Dateien sollten problemlos gedruckt werden.

10.4.1.4. Konvertierungsfilter

Nachdem Sie Ihren Drucker wie unter Einfache Drucker-Konfiguration eingerichtet haben, wollen Sie wahrscheinlich einige Konvertierungsfilter installieren, damit Sie (abgesehen von ASCII-Text) auch Ihre Lieblings-Dateiformate drucken können.

10.4.1.4.1. Warum sollte ich einen Konvertierungsfilter installieren?

Konvertierungsfilter erleichtern das Drucken von verschiedenen Dateiformaten. Nehmen wir beispielsweise an, dass Sie sehr viel mit dem TEX-Satzsystem arbeiten und über einen PostScript-Drucker verfügen. Eine vom TEX-System erzeugte DVI-Datei kann erst dann gedruckt werden, nachdem diese nach PostScript konvertiert wurde. Dazu geben Sie Folgendes ein:

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

Haben Sie einen Konvertierungsfilter für DVI-Dateien installiert, können Sie die manuelle Konvertierung überspringen, da dies nun **LPD** für Sie erledigt. Wollen Sie eine DVI-Datei drucken, geben Sie nur noch den folgenden Befehl ein:

```
% lpr -d seaweed-analysis.dvi
```

Durch die Verwendung der Option `-d` wurde **LPD** angewiesen, unsere DVI-Datei vor dem Druck zu konvertieren. Der Abschnitt Formatierungs- und Konvertierungsoptionen beschreibt die dabei möglichen Optionen.

Für jede Konvertierungsoption, die Ihr Drucker unterstützen soll, müssen Sie einen eigenen *Konvertierungsfilter* installieren und dessen Pfad in der Datei `/etc/printcap` angeben. Ein Konvertierungsfilter verhält sich im Prinzip wie ein Textfilter bei einer einfachen Druckerkonfiguration (lesen Sie dazu auch den Abschnitt Den Textfilter installieren), allerdings konvertiert er die Datei in ein Format, das Ihr Drucker versteht, anstatt normalen Text zu drucken.

10.4.1.4.2. Welche Konvertierungsfilter sollte ich installieren?

Sie sollten nur Filter installieren, die Sie auch benötigen. Wenn Sie sehr viele DVI-Dateien drucken, sollten Sie auch einen DVI-Konvertierungsfilter installieren. Müssen Sie viele troff-Daten drucken, ist ein troff-Filter hilfreich.

Die folgende Tabelle listet die von **LPD** unterstützten Filter sowie die Einträge in `/etc/printcap` auf, mit denen Sie diese Fähigkeiten aktivieren. Zusätzlich wird angegeben, wie Sie `lpr` jeweils aufrufen müssen:

Dateityp	`/etc/printcap`-Fähigkeit	`lpr`-Option
cifplot	`cf`	`-c`
DVI	`df`	`-d`
plot	`gf`	`-g`
ditroff	`nf`	`-n`
FORTRAN-Text	`rf`	`-f`
troff	`tf`	`-f`

Dateityp	/etc/printcap-Fähigkeit	lpr-Option
Rasterdaten	vf	-v
Normaler Text	if	keine, -p, or -l

Wollen Sie also lpr -d verwenden, muss die df-Fähigkeit in /etc/printcap aktiviert sein.

Obwohl manche Leute etwas anderes behaupten, sind Formate wie FORTRAN-Text und -Plot inzwischen nahezu obsolet. Wenn Sie diese Formate dennoch benötigen, installieren Sie einfach einen angepassten Filter. Wollen Sie beispielsweise zwar Printerleaf-Dateien (also Dateien des Desktop Publishing-Programms von Interleaf), aber keine Plotdateien drucken, so können Sie einen Printerleaf-Konvertierungsfilter installieren, der es durch die Aktivierung der gf-Fähigkeit erlaubt, diese Dateien direkt zu drucken. Nun müssen Sie Ihren Mitarbeitern nur noch mitteilen, dass lpr -g nun für “drucke Printerleaf-Dateien” steht.

10.4.1.4.3. Konvertierungsfilter installieren

Da Konvertierungsfilter nicht zum Basissystem von FreeBSD gehören, sollten diese unter /usr/local installiert werden. Häufig wird das Verzeichnis /usr/local/libexec verwendet, da es sich bei Konvertierungsfiltern um spezielle Programme handelt, die nur von **LPD**, aber nicht von einem normalen Benutzer gestartet werden.

Um einen Konvertierungsfilter zu aktivieren, müssen Sie dessen Pfad zusätzlich zur benötigten Fähigkeit in der Datei /etc/printcap eintragen.

In unserem Beispiel wollen wir einen DVI-Konvertierungsfilter für den Drucker bamboo installieren. Unsere bereits bekannte /etc/printcap wurde allerdings um die df-Fähigkeit für den Drucker bamboo erweitert:

```
#
#  /etc/printcap des Rechners rose - neuer df-Filter für bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Beim DVI-Filter handelt es sich um ein Shell-Skript namens /usr/local/libexec/psdf:

```
#!/bin/sh
#
#  psdf - DVI-nach-PostScript Druckerfilter
#  Installiert unter:  /usr/local/libexec/psdf
#
#  Wird von lpd aktiviert, wenn der Nutzer lpr -d eingibt.
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

Dieses Skript startet dvips im Filtermodus (durch das Argument -f wird der Druckauftrag über die Standardeingabe entgegengenommen). Danach wird der PostScript-Druckerfilter lprps (lesen Sie dazu auch den Abschnitt

Drucken von reinen Textdateien auf einem PostScript-Drucker) mit den von **LPD** übergebenen Argumenten gestartet. Das `lprps`-Werkzeug wiederum nutzt diese Argumente, um die gedruckten Seiten zu verrechnen.

10.4.1.4.4. Beispiele für Konvertierungsfilter

Da es keine verbindliche Prozedur zur Installation eines Druckerfilters gibt, folgen nun weitere Beispiele in diesem Abschnitt. Verwenden Sie diese, um Ihre eigenen Filter zu erstellen. Falls ein Filter Ihren Anforderungen bereits entspricht, können Sie ihn auch direkt verwenden.

Das erste Beispiel beschreibt einen Konvertierungsfilter für GIF-Dateien für den Drucker LaserJet III-Si von Hewlett Packard:

```
#!/bin/sh
#
#  hpvf - Konvertiert GIF-Dateien nach HP/PCL, danach wird gedruckt.
#  Installiert unter:  /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | ppmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

Dieser Filter konvertiert eine GIF-Datei in eine portable Anymap, diese in ein portables Graustufenbild, dieses wiederum in eine portable Bitmap, die schließlich in ein LaserJet/PCL-kompatibles Format umgewandelt wird.

`/etc/printcap` muss für einen Drucker, der diesen Filter nutzen will, folgenden Eintrag enthalten:

```
#
#  /etc/printcap des Rechners orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:
```

Das folgende Skript ist ein Konvertierungsfilter, der das Drucken von troff-Daten des groff-Textsatzsystems auf dem PostScript-Drucker `bamboo` ermöglicht:

```
#!/bin/sh
#
#  pstf - Konvertiert groff's troff-Daten nach PS, dann wird gedruckt.
#  Installiert unter:  /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

Dieses Skript nutzt wiederum `lprps`, um mit dem Drucker zu kommunizieren. Wäre der Drucker an einem parallelen Port angeschlossen, würde das Skript so aussehen:

```
#!/bin/sh
#
#  pstf - Konvertiert groff's troff-Daten nach PS, danach wird gedruckt.
#  Installiert unter:  /usr/local/libexec/pstf
#
```

```
exec grops
```

Das ist alles. Um den Filter verwenden zu können, müssen Sie ihn allerdings noch in `/etc/printcap` aktivieren:

```
:tf=/usr/local/libexec/pstf:
```

Das nächste Skript ist ein FORTRAN-Textfilter für jeden Drucker, der normalen Text direkt drucken kann und der hier für den Drucker `teak` installiert wird:

```
#!/bin/sh
#
# hprf - FORTRAN-Textfilter für den Drucker LaserJet 3si:
# Installiert unter:  /usr/local/libexec/hprf
#

printf "\033&k2G" && fpr && printf "\033&l0H" &&
 exit 0
exit 2
```

Zusätzlich benötigen wir wiederum einen Eintrag in `/etc/printcap`, um diesen Filter für den Drucker `teak` zu aktivieren:

```
:rf=/usr/local/libexec/hprf:
```

Das letzte Beispiel ist etwas komplexer. Es soll ein DVI-Filter für den bereits erwähnten LaserJet-Drucker `teak` installiert werden. Der erste Teil ist einfach: Sie müssen den Pfad des DVI-Filters in `/etc/printcap` eintragen:

```
:df=/usr/local/libexec/hpdf:
```

Nun kommt der schwierige Teil: Sie müssen den Filter funktionsfähig machen. Dazu benötigen Sie einen DVI-nach-LaserJet/PCL-Konverter. Glücklicherweise enthält die FreeBSD Ports-Sammlung mit `print/dvi2xx` ein solches Programm. Nach der Installation des Pakets verfügen wir über das Programm `dvilj2p`, das zur Konvertierung von DVI-Daten in zu den Druckern LaserJet IIp, LaserJet III, sowie LaserJet 2000 kompatible Codes benötigt wird.

Durch den Einsatz von `dvilj2p` wird der Filter `hpdf` relativ komplex, da `dvilj2p` nicht von der Standardeingabe lesen kann, sondern als Eingabe einen Dateinamen erwartet. Zusätzlich muss der Dateiname auf `.dvi` enden, daher ist die Verwendung von `/dev/fd/0` als Standardeingabe problematisch. Wir können diese Problem aber umgehen, indem wir einen temporären Dateinamen symbolisch nach `/dev/fd/0` linken. Dadurch wird `dvilj2p` gezwungen, dennoch von der Standardeingabe zu lesen.

Das letzte Problem, das wir noch lösen müssen, ist, dass wir `/tmp` nicht als temporären Link verwenden können. Symbolische Links gehören dem User sowie der Gruppe `bin`. Der Filter läuft aber als User `daemon`. Außerdem ist `/tmp` durch ein Sticky-Bit gesichert. Daher kann der Filter den Link zwar erzeugen, ein Aufräumen ist aber nicht mehr möglich, weil sich die Eigentümer des Filters und des temporären Verzeichnisses unterscheiden.

Daher legt der Filter den symbolischen Link im Arbeitsverzeichnis an, das gleichzeitig als Spooling-Verzeichnis dient (festgelegt durch die Aktivierung der `sd`-Fähigkeit in `/etc/printcap`). Das Arbeitsverzeichnis ist ein idealer Ort für den Filter, insbesondere da dieses (manchmal) sogar über mehr freien Speicherplatz als `/tmp` verfügt.

Mit diesen Informationen sind wir nun in der Lage, den Filter zu entwickeln:

```
#!/bin/sh
#
```

```
#  hpdf - DVI-Daten auf einen HP/PCL-Drucker drucken
#  Installiert unter:  /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
#  Eine Funktion zum Aufräumen unserer temporären Dateien.
#  Diese finden sich im Arbeitsverzeichnis, das wir auch als
#  Spooling-Verzeichnis für unseren Drucker verwenden werden.
#
cleanup() {
   rm -f hpdf$$.dvi
}

#
#  Eine Funktion, um fatale Fehler zu behandeln.  Dazu die Meldung
#  ausgeben, danach ein exit 2.  Dadurch weiß LPD, dass es
#  den Auftrag nicht noch einmal drucken soll.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
}

#
#  Wenn ein Anwender den Auftrag entfernt, sendet LPD ein SIGINT, daher
#  wollen wir SIGINT und einige andere Signale abfangen (trappen), um
#  nach der Konvertierung aufräumen zu können.
#
trap cleanup 1 2 15

#
#  Bevor wir anfangen, räumen wir noch auf.  Sicher ist sicher.
#
cleanup

#
#  Die DVI-Eingabedatei auf die Standardeingabe linken (die zu druckende
#  Datei).
#
ln -s /dev/fd/0 hpdf$$.dvi || fatal "Konnte Symlink nicht anlegen!"

#
#  Umwandeln: LF = CR+LF
#
printf "\033&k2G" || fatal "Konnte Drucker nicht initialisieren!"

#
#  Konvertieren und drucken.  Da der Rückgabewert von dvilj2p
#  unzuverlässig ist, ignorieren wir ihn einfach.
#
dvilj2p -M1 -q -e- dfhp$$.dvi
```

```
#
#  Aufräumen und beenden.
#
cleanup
exit 0
```

10.4.1.4.5. Automatische Konvertierung: Eine Alternative zu Konvertierungsfiltern

Alle in diesem Abschnitt besprochenen Konvertierungsfilter sind zwar sehr hilfreich, allerdings müssen Sie nach wie vor bei jedem Aufruf von lpr(1) angeben, welchen Filter sie verwenden wollen, was mit der Zeit sicher nervend wird. Schlimmer ist allerdings, dass die Auswahl eines unpassenden Filters dazu führen kann, dass Sie Hunderte Seiten Papier ausdrucken.

Statt also Konvertierungsfilter zu installieren, könnten Sie den Textfilter (der ohnehin der Standardfilter ist) verwenden, um den zu druckenden Dateityp zu erkennen und anschließend den korrekten Konvertierungsfilter auszuwählen. Um den Dateityp zu bestimmen, können Sie beispielsweise `file` verwenden. Leider ist es bei *einigen* Dateitypen problematisch, diese zu unterscheiden. Daher könnten Sie für diese Dateitypen dennoch einen Konvertierungsfilter installieren.

Die FreeBSD Ports-Sammlung enthält mit `apsfilter` (`print/apsfilter`) einen Textfilter, der diese automatische Konvertierung durchführen kann. Er ist in der Lage, normalen Text, PostScript, DVI und beinahe jede Art von Datei zu erkennen, diese zu konvertieren und auf Ihren Drucker auszugeben.

10.4.1.5. Ausgabefilter

LPD unterstützt noch eine weitere Filterart, die sogenannten Ausgabefilter. Diese sind – analog zu einem Textfilter – für den Druck von normalem Text ausgelegt, allerdings verfügen sie im Vergleich zu diesen nur über sehr eingeschränkte Fähigkeiten. Wenn Sie einen Ausgabefilter (aber keinen Textfilter) verwenden, dann

- startet **LPD** nur einen Ausgabefilter für den kompletten Druckauftrag, statt für jede Datei des Auftrags einen eigenen Filter zu starten.
- kümmert sich **LPD** nicht darum, den Beginn oder das Ende einer Datei innerhalb des Druckauftrages zu finden.
- übergibt **LPD** weder den Benutzer- noch den Rechnernamen desjenigen, der den Druckauftrag erteilt hat, an den Ausgabefilter, was eine Verrechnung von Druckaufträgen unmöglich macht. Ausgabefilter unterstützen insgesamt nur zwei Argumente:

  `filter-name` -w *`width`* -l *`length`*

  `width` basiert auf der `pw`-Fähigkeit, *`length`* hingegen auf der `pl`-Fähigkeit des gewählten Druckers.

Lassen Sie sich von dieser angeblichen Einfachheit eines Ausgabefilters nicht täuschen. Ausgabefilter sind beispielsweise *nicht dazu in der Lage*, jede Datei eines Druckauftrages auf einer neuen Seite zu drucken. Dazu benötigen Sie einen Textfilter (die im Abschnitt Den Textfilter installieren beschrieben werden). Außerdem sind Ausgabefilter in Wirklichkeit *komplexer*, da sie den gesendeten Bytestrom nicht nur auf Sonderzeichen hin untersuchen müssen, sondern auch die Übertragung von Signalen für **LPD** übernehmen müssen.

Sie *benötigen* Ausgabefilter aber dann, wenn Sie Deckblätter drucken wollen, da dazu Escape-Sequenzen und Initialisierungsstrings erforderlich sind. (Es ist allerdings *nicht möglich*, den Druck dieser Deckblätter zu verrechnen, da **LPD** keine Benutzer- oder Rechnerinformationen an den Ausgabefilter übergibt.)

LPD kann für den gleichen Drucker sowohl Ausgabefilter als auch Textfilter verwenden. In solchen Fällen verwendet **LPD** den Ausgabefilter nur für den Druck von Deckblättern (die im Abschnitt Deckblätter näher beschrieben werden). Nach dem Druck des Deckblattes erwartet **LPD**, dass sich der Ausgabefilter *selbst beendet*. Dazu werden zwei Bytes an den Ausgabefilter gesendet: ASCII 031, gefolgt von ASCII 001. Wenn ein Ausgabefilter diese zwei Bytes (031, 001) empfängt, sendet er das Signal `SIGSTOP` an sich selbst. Nachdem **LPD** den Rest des Druckauftrages erledigt hat, wird der Ausgabefilter erneut gestartet, indem ein `SIGCONT` an den Ausgabefilter gesendet wird.

Haben Sie nur einen Ausgabefilter, aber *keinen* Textfilter installiert, dann verwendet **LPD** den Ausgabefilter auch für den Druck von normalem Text. Wie bereits erwähnt, werden dabei allerdings alle Dateien des Druckauftrags unmittelbar hintereinander gedruckt, Seitenumbrüche oder ein zusätzlicher Papiervorschub sind also nicht möglich. Da dieses Verhalten von Ihnen wahrscheinlich *nicht* gewünscht wird, werden Sie in fast allen Fällen einen zusätzlichen Textfilter benötigen.

Der weiter oben beschriebene Textfilter `lpf` kann auch als Ausgabefilter verwendet werden. Wenn Sie nur einen funktionierenden Ausgabefilter benötigen, aber nicht den dafür benötigten Code (zur Zeichenerkennung und zum Senden von Signalen) schreiben wollen, sollten Sie sich `lpf` näher ansehen. Sie können `lpf` auch in ein Shell-Skript einbinden, um von Ihrem Drucker benötigte Initialisierungscodes zu verarbeiten.

10.4.1.6. `lpf`: Ein Textfilter

Der Textfilter (Eingabefilter) `/usr/libexec/lpr/lpf` wird bereits mit FreeBSD geliefert. Er erlaubt das Einrücken der Ausgabe (über `lpr -i`), die Übergabe von Zeichen-Literalen (über `lpr -l`), das Anpassen der Druckposition bei gelöschten Zeichen (*Backspaces*) oder Tabulatoren, sowie die Verrechnung gedruckter Seiten. Zusätzlich kann dieser Textfilter auch als Ausgabefilter arbeiten.

`lpf` ist für viele verschiedene Druckumgebungen geeignet. Zwar ist dieser Textfilter nicht in der Lage, Initialisierungssequenzen an einen Drucker zu senden, dieses Problem kann allerdings durch das Schreiben und Ausführen eines Shell-Skripts (das diese Funktion übernimmt) und das anschließende Aufrufen von `lpf` gelöst werden.

Damit Sie `lpf` für die Verrechnung von Druckaufträgen einsetzen können, müssen Sie die korrekten Werte für die `pw`- und `pl`-Fähigkeiten in `/etc/printcap` eintragen. `lpf` verwendet diese Werte, um festzustellen, wieviel Text auf eine Seite passt und wieviele Seiten im Druckauftrag enthalten sind. Weitere Informationen zur Verrechnung der Druckernutzung enthält der Abschnitt Die Druckernutzung verrechnen.

10.4.2. Deckblätter

Wenn Sie *viele* Benutzer mit verschiedenen Druckern verwalten müssen, sollten Sie *Deckblätter* als notwendiges Übel akzeptieren.

Deckblätter (manchmal auch als *Bannerseiten* oder *burst pages* bezeichnet) geben an, wem die Ausgabe eines Druckauftrags gehört. Sie werden normalerweise in großen fetten Buchstaben gedruckt, manchmal sogar mit zusätzlicher Umrandung, damit man sie leichter von den tatsächlichen Seiten eines Druckauftrages unterscheiden kann. Der Nachteil von Deckblättern ist allerdings, dass es sich dabei um eine zusätzliche zu druckende Seite handelt, die in der Regel bereits nach wenigen Minuten wieder im Papierkorb landet. Da aber für jeden Druckauftrag nur ein einziges Deckblatt gedruckt wird, ist der Papierverbrauch in den meisten Fällen tolerierbar.

Das **LPD**-System kann Deckblätter automatisch erzeugen, *wenn* Ihr Drucker normalen Text direkt drucken kann. Haben Sie hingegen einen PostScript-Drucker, benötigen Sie ein externes Programm, um die Deckblätter zu generieren (Lesen Sie dazu auch den Abschnitt Deckblätter auf PostScript-Druckern erzeugen.).

10.4.2.1. Deckblätter aktivieren

Im Abschnitt Einfache Drucker-Konfiguration haben wir die Ausgabe von Deckblättern durch die die Angabe der Option `sh` (*suppress header*) in `/etc/printcap` deaktiviert. Um die Ausgabe von Deckblättern wieder zu aktivieren, müssen Sie daher die `sh`-Fähigkeit wieder entfernen.

Das klingt zu einfach? Wo ist der Haken?

Sie haben recht. Es ist *möglich*, dass Sie einen Ausgabefilter verwenden müssen, um die nötigen Initialisierungsstrings an den Drucker zu senden. Das folgende Beispiel beschreibt einen Ausgabefilter für PCL-kompatible Drucker von Hewlett Packard:

```
#!/bin/sh
#
#  hpof - Ausgabefilter für PCL-kompatible Drucker von Hewlett Packard
#  Installiert unter:  /usr/local/libexec/hpof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

Geben Sie den Pfad des Ausgabefilters über die `of`-Fähigkeit an. Weitere Informationen finden Sie im Abschnitt Ausgabefilter.

Das nächste Beispiel beschreibt die Datei `/etc/printcap` des bereits erwähnten Druckers `teak`. Allerdings sind nun die Ausgabe von Deckblättern sowie der vorhin beschriebene Ausgabefilter enthalten:

```
#
#  /etc/printcap für den Rechner orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/hpif:\
        :vf=/usr/local/libexec/hpvf:\
        :of=/usr/local/libexec/hpof:
```

Wenn ein Anwender nun einen Druckauftrag an den Drucker `teak` schickt, wird für jeden Druckauftrag ein Deckblatt erstellt. Benötigt ein Anwender keine Deckblätter, kann er die Ausgabe dieser Seiten durch die Verwendung von `lpr -h` unterdrücken. Weitere, für die Ausgabe von Deckblättern interessante lpr(1)-Optionen finden Sie im Abschnitt Deckblattoptionen.

> **Anmerkung: LPD** verwendet ein *Form Feed*, um das Deckblatt abzuschließen. Wenn Ihr Drucker ein anderes Zeichen verwendet, um eine Seite auszuwerfen, geben Sie dieses über die `ff`-Fähigkeit in `/etc/printcap` an.

10.4.2.2. Deckblätter kontrollieren

Haben Sie die Ausgabe von Deckblättern aktiviert, gibt **LPD** eine ganze Seite in großen Buchstaben aus, die den Anwender, den verwendeten Rechner sowie den Druckauftrag beschreiben. Das folgende Beispiel ist ein Deckblatt für den Druckauftrag “outline”, der von `kelly` auf dem Rechner `rose` erstellt wurde:

```
k                  ll       ll
k                   l        l
k                   l        l
k   k     eeee      l        l     y    y
k  k     e    e     l        l     y    y
k k      eeeeee     l        l     y    y
kk k     e          l        l     y    y
k   k    e    e     l        l     y   yy
k    k    eeee     lll      lll     yyy y
                                        y
                                       y    y
                                        yyyy


                              ll
                   t          l        i
                   t          l
 oooo    u    u   ttttt      l       ii     n nnn     eeee
o    o   u    u     t        l        i     nn   n   e    e
o    o   u    u     t        l        i     n    n   eeeeee
o    o   u    u     t        l        i     n    n   e
o    o   u   uu     t  t     l        i     n    n   e    e
 oooo     uuu u      tt      lll      iii    n    n    eeee


r rrr     oooo     ssss     eeee
rr   r   o    o   s    s   e    e
r        o    o    ss      eeeeee
r        o    o      ss    e
r        o    o   s    s   e    e
r         oooo     ssss     eeee


                                   Job:  outline
                                   Date: Sun Sep 17 11:04:58 1995
```

LPD fügt ein *Form Feed* an diesen Text an, damit der eigentliche Druckauftrag auf einer neuen Seite gestartet wird (es sei denn, Sie haben die `sf`-Fähigkeit (*suppress form feeds*) des jeweiligen Druckers in `/etc/printcap` aktiviert).

Wenn Sie dies wüschen, kann **LPD** auch nur ein *kurzes Deckblatt* ausgeben. Dazu verwenden Sie die Option `sb` (*short banner*) in `/etc/printcap`. Dadurch erhalten Sie ein Deckblatt ähnlich dem folgenden:

```
rose:kelly  Job: outline  Date: Sun Sep 17 11:07:51 1995
```

In der Voreinstellung druckt **LPD** zuerst das Deckblatt und danach den eigentlichen Druckauftrag. Um diese Reihenfolge umzukehren, geben Sie die Option `hl` (*header last*) in `/etc/printcap` ans.

10.4.2.3. Deckblätter verrechnen

Wenn Sie die in **LPD** eingebaute Funktion zur Erstellung von Deckblättern verwenden, werden Sie auf folgendes Paradigma stoßen: Deckblätter müssen *kostenlos* sein.

Warum ist das so?

Weil der Ausgabefilter das einzige externe Programm ist, das zum Zeitpunkt der Erstellung des Deckblatts eine Verrechnung durchführen könnte. Da Ausgabefilter aber weder über *Benutzer- noch über Rechnerinformationen* verfügen, ist es nicht möglich, einen Druckauftrag einem bestimmten Benutzer zuzuordnen. Da ein Benutzer die Ausgabe von Deckblättern über `lpr -h` unterdrücken kann, ist es auch nicht möglich, die Vorgabe “verrechne eine zusätzliche Seite” in den Text- oder Konvertierungsfilter (die über die zur Verrechnung nötigen Benutzer- und Rechnerinformationen verfügen) aufzunehmen, weil Benutzer sonst für Deckblätter bezahlen müssten, die sie nicht gedruckt haben.

Es ist *ebenfalls nicht ausreichend*, jeden Filter eigene Deckblätter erzeugen zu lassen (und sie dadurch verrechnen zu können). Wollte ein Benutzer durch ein `lpr -h` die Ausgabe eines Deckblattes unterdrücken, würde dieses nun trotzdem verrechnet werden, da **LPD** keine Informationen über die Verwendung der Option `-h` an einen Filter weitergibt.

Welche Möglichkeiten habe ich nun?

Sie können:

- Das Paradigma von **LPD** einfach akzeptieren und die Deckblätter gratis abgeben.
- Eine alternatives Drucksystem wie **LPRng** installieren. Der Abschnitt Alternativen zum Standard-Drucksystem beschreibt verschiedene Drucksysteme, die **LPD** ersetzen können.
- Schreiben Sie einen *intelligenten* Ausgabefilter. Normalerweise kümmert sich ein Ausgabefilter nur um die Initialisierung des Druckers oder um eine einfache Zeichenkonvertierung. Außerdem eignet er sich für die Ausgabe von Deckblättern und normalem Text, wenn Sie keinen Text- oder Eingabefilter installiert haben. Haben Sie allerdings einen Textfilter installiert, verwendet **LPD** Ausgabefilter nur für die Ausgabe von Deckblättern. Ein Ausgabefilter kann den Text des von **LPD** erzeugten Deckblattes untersuchen, um festzustellen, welcher Benutzer und welcher Rechner den Druckauftrag gestartet hat. Leider weiß der Ausgabefilter auch mit dieser Methode nicht, welche Datei er zur Verrechnung verwenden soll (da der Name dieser Datei durch die `af`-Fähigkeit übergeben wird). Wenn Sie eine Standard-Verrechnungsdatei verwenden, können Sie diese in den Ausgabefilter einbauen. Um den Text des Deckblattes zu untersuchen, verwenden Sie die `sh`-Fähigkeit (*short header*) in `/etc/printcap`. Falls Ihnen das zuviel Aufwand ist, freuen sich Ihre Benutzer sicher darüber, wenn Sie ihnen den kostenlosen Druck von Deckblättern erlauben.

10.4.2.4. Deckblätter auf PostScript-Druckern ausgeben

In der Regel erzeugt **LPD** ein Deckblatt mit normalem Text, das für viele verschiedene Drucker geeignet ist. Da PostScript-Drucker normalen Text aber nicht drucken können, ist die **LPD**-Funktion zur Erstellung von Deckblättern auf diesen Drucker relativ sinnlos.

Es sei denn, jeder Text- und Konvertierungsfilter erzeugt über den Benutzer- und Rechnernamen sein eigenes, für den jeweiligen Drucker geeignetes Deckblatt. Das Problem dieser Methode ist allerdings, dass ein Anwender auch dann ein Deckblatt erhält, wenn er dies über `lpr -h` verhindern wollte.

Das folgende Skript benötigt drei Argumente (den Loginnamen des Benutzers, den Rechnernamen und den Namen des Druckauftrages), um daraus ein einfaches PostScript-Deckblatt zu erzeugen:

```
#!/bin/sh
#
#  make-ps-header - ein PostScript-Deckblatt auf stdout ausgeben
#  Installiert unter:  /sr/local/libexec/make-ps-header
#

#
#  Die folgenden Werte sind PostScript-Einheiten (72 pro Zoll).
#  Passen Sie diese Werte für A4 oder die von Ihnen verwendete
#  Papiergröße an:
#
page_width=612
page_height=792
border=72

#
#  Argumente prüfen
#
if [ $# -ne 3 ]; then
    echo "Usage: ‘basename $0‘ <user> <host> <job>" 1>&2
    exit 1
fi

#
#  Diese Werte in Variablen speichern, damit der PostScript-Code
#  übersichtlicher wird.
#
user=$1
host=$2
job=$3
date=‘date‘

#
#  Sende den PostScript-Code an stdout.
#
exec cat <<EOF
%!PS

%
%  Sicherstellen, dass es keine unerwünschten Wechselwirkungen mit
%  dem folgenden Druckauftrag gibt.
```

```
%
save

%
%  Ziehe eine fette Umrandung.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
%  Zeige den Benutzernamen groß und fett an.
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub moveto
($user) show

%
%  Und nun zeige noch die Einzelheiten an.
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) ] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) ] {
        270 y moveto show /y y 18 sub def
} forall

%
% Das wars.
%
restore
showpage
EOF
```

Nun kann jeder Konvertierungs- oder Textfilter dieses Skript aufrufen, um zuerst das Deckblatt zu erzeugen und danach den Druckauftrag zu drucken. Das nächste Beispiel enthält den bereits beschriebenen DVI-Konvertierungsfilter, der hier um die Funktion zur Erzeugung eines Deckblatts erweitert wurde:

```
#!/bin/sh
#
#  psdf - DVI-nach-PostScript - Druckerfilter
#  Installiert unter:  /usr/local/libexec/psdf
#
#  Wird von lpd aufgerufen, wenn der Benutzer lpr -d verwendet.
#
```

```
orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do
    case $option in
        x|y)  ;; # Ignore
        n)    login=$OPTARG ;;
        h)    host=$OPTARG ;;
        *)    echo "LPD started ‘basename $0‘ wrong." 1>&2
              exit 2
              ;;
    esac
done

[ "$login" ] || fail "No login name"
[ "$host" ] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps $orig_args
```

Beachten Sie, dass der Filter die Liste der Argumente überprüft, um den Benutzer- und den Rechnernamen zu ermitteln. Dieser Vorgang ist prinzipiell für alle Filter identisch. Der Textfilter benötigt allerdings etwas andere Argumente, die im Abschnitt Die Funktionsweise von Filtern beschrieben werden.

Wie bereits erwähnt, deaktiviert diese Methode leider die “suppress header page”-Option (also die Option `-h`) von `lpr`. Benutzer können danach den Ausdruck eines Deckblattes nicht mehr verhindern, da der angepasste Filter zu jedem Druckauftrag automatisch ein Deckblatt erstellt.

Damit ein Benutzer bei Bedarf den Ausdruck eines Deckblatts dennoch unterbinden kann, müssen Sie auch hier den im Abschnitt Deckblätter verrechnen beschriebenen Trick einsetzen: Schreiben Sie einen Ausgabefilter, der das von LPD erzeugte Deckblatt untersucht und daraus eine PostScript-Version erzeugt. Wenn der Benutzer den Druckauftrag mit `lpr -h` verschickt, erzeugt **LPD** kein Deckblatt, was in weiterer Folge auch für Ihren Ausgabefilter gilt. Soll hingegen ein Deckblatt erzeugt werden, liest der Ausgabefilter den von **LPD** übergebenen Text und erzeugt daraus ein für Ihren PostScript-Drucker geeignetes Deckblatt.

Haben Sie Ihren PostScript-Drucker über eine serielle Verbindung angeschlossen, können Sie auch `lprps` verwenden. In diesem Paket ist mit `psof` auch ein Ausgabefilter enthalten, der die eben beschriebenen Funktionen übernehmen kann. Beachten Sie aber, dass Sie mit `psof` keine Deckblätter verrechnen können.

10.4.3. Drucken über ein Netzwerk

FreeBSD unterstützt das Drucken über ein Netzwerk, also den Versand von Druckaufträgen an einen entfernten Drucker. Man unterscheidet dabei zwei Möglichkeiten:

- Den Zugriff auf einen an einem entfernten Rechner angeschlossenen Drucker. Sie konfigurieren dabei auf Ihrem System einen Drucker, der über eine konventionelle serielle oder parallele Verbindung an einem anderen Rechner

angeschlossen ist. Danach richten Sie **LPD** auf dem entfernten System so ein, dass andere Drucker über das Netzwerk auf diesen Drucker zugreifen können. Der Abschnitt Auf entfernten Rechnern installierte Drucker beschreibt, wie Sie dazu vorgehen müssen.

- Den Zugriff auf einen direkt an ein Netzwerk angeschlossenen Drucker. Ein solcher Drucker verfügt anstelle (oder zusätzlich zu) einer parallelen oder seriellen Schnittstelle über eine Netzwerkschnittstelle. Ein solcher Drucker kann sich auf zwei Arten verhalten:
  - Er kann das **LPD**-Protokoll direkt unterstützen und sogar Druckjobs von entfernten Rechner verwalten. In diesem Fall verhält sich der Drucker wie ein normaler Rechner, auf dem **LPD** läuft. Lesen Sie den Abschnitt Auf entfernten Rechnern installierte Drucker, um einen solchen Drucker einzurichten.
  - Er könnte Verbindungen über ein Netzwerk unterstützen. In diesem Fall "verbinden" Sie den Drucker mit einem Rechner Ihres Netzwerks, der danach für die Verwaltung von Druckaufträgen sowie den tatsächlichen Druck verantwortlich ist. Der Abschnitt Drucker mit direkter TCP-Schnittstelle enthält Hinweise zur Installation derartiger Drucker.

10.4.3.1. Auf entfernten Rechnern installierte Drucker

Das **LPD**-Drucksystem unterstützt den Versand von Druckaufträgen an andere Rechner, auf denen entweder **LPD** läuft oder die zu **LPD** kompatibel sind. Dadurch können Sie einen Drucker auf einem Rechner installieren und von anderen Rechnern des Netzwerks darauf zugreifen. Außerdem werden Drucker mit direkter TCP-Schnittstelle unterstützt, wenn diese das **LPD**-Protokoll unterstützen.

Um diese Art des Druckens über ein Netzwerk zu aktivieren, installieren Sie zuerst Ihren Drucker auf einem Rechner Ihres Netzwerks, dem sogenannten *printer host*. Die dazu nötigen Schritte werden im Abschnitt Einfache Drucker-Konfiguration beschrieben. Falls Sie eine erweiterte Druckerkonfiguration benötigen, sollten Sie auch den Abschnitt Erweiterte Drucker-Konfiguration lesen. Danach testen Sie, ob der Drucker alle von Ihnen aktivierten **LPD**-Fähigkeiten unterstützt. Stellen Sie auch sicher, dass Ihr *lokales System* berechtigt ist, den **LPD**-Dienst auf dem *entfernten System* zu nutzen (lesen Sie dazu den Abschnitt Druckaufträge auf entfernten Druckern beschränken).

Wenn Sie einen Drucker mit einer zu **LPD** kompatiblen Netzwerkschnittstelle verwenden, handelt es sich beim *printer host* um den Drucker selbst, und der *Druckername* ist der von Ihnen für diesen Drucker vorgegebene Name. Lesen Sie die Dokumentation Ihres Druckers und/oder der Netzwerkschnittstelle Ihres Druckers, um dies zu klären.

> **Tipp:** Wenn Sie einen Hewlett Packard Laserjet-Drucker verwenden, sorgt der Druckername `text` für eine automatische LF-zu-CRLF-Konvertierung. In diesem Fall wird das `hpif`-Skript nicht benötigt.

Danach müssen Sie auf jedem Rechner, der auf diesen Drucker zugreifen soll, einen entsprechenden Eintrag in deren `/etc/printcap` aufnehmen. Dazu werden folgende Informationen benötigt:

1. Der Name des Eintrags. Entspricht in der Regel dem Eintrag auf dem *printer host*.
2. Lassen Sie den Eintrag für die `lp`-Fähigkeit leer, schreiben Sie also `:lp=:`.
3. Erzeugen Sie ein Spooling-Verzeichnis und geben Sie dessen Pfad über die `sd`-Fähigkeit an. **LPD** speichert Ihre Druckaufträge in diesem Verzeichnis, bevor sie an den Drucker geschickt werden.
4. Geben Sie den Namen des *printer hosts* über die `rm`-Fähigkeit an.

5. Geben Sie den Namen des Druckers (auf dem *printer host*) über die `rp`-Fähigkeit an.

Das ist alles. Sie benötigen weder Konvertierungsfilter, noch Seitengrößen oder sonstige Angaben in Ihrer lokalen `/etc/printcap`.

Dazu ein Beispiel. Der Rechner `rose` verfügt über zwei Drucker, `bamboo` und `rattan`. Wir wollen nun allen Benutzern des Rechners `orchid` erlauben, diese Drucker zu verwenden. Es folgt nun wieder die bereits aus dem Abschnitt Deckblätter verwenden bekannte `/etc/printcap` für den Rechner `orchid`. Diese enthielt bereits einen Eintrag für den Drucker `teak`. Zusätzlich tragen wir nun die zwei Drucker des Rechners `rose` ein:

```
#
#  /etc/printcap für den Rechner orchid - mit zusätzlichen
#  Einträgen für die (entfernten) Drucker auf dem Rechner rose
#

#
#  teak ist ein lokaler Drucker und direkt mit orchid verbunden:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

#
#  rattan ist mit rose verbunden, Druckaufträge für rattan gehen daher
#  an den Rechner rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
#  bamboo ist ebenfalls mit rose verbunden:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:
```

Nun müssen wir nur noch die Spooling-Verzeichnisse auf dem Rechner `orchid` erzeugen:

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo
```

Damit können Benutzer des Rechners `orchid` die Drucker `rattan` und `bamboo` verwenden. Ein Benutzer gibt auf `orchid` beispielsweise ein:

```
% lpr -P bamboo -d sushi-review.dvi
```

Die Anwendung **LPD** auf dem Rechner `orchid` kopiert daraufhin den Druckauftrag in das Spooling-Verzeichnis `/var/spool/lpd/bamboo` und stellt fest, dass es sich um einen DVI-Auftrag handelt. Sobald `rose` über genug freien Platz im `bamboo`-Spooling-Verzeichnis verfügt, würden die beiden **LPD** die Datei auf den Rechner `rose` transferieren. Diese Datei verbleibt danach in der Druckerwarteschlange des Rechners `rose`, bis der Ausdruck der

Datei abgeschlossen ist. Vor dem Ausdruck würde die Datei noch von DVI nach PostScript konvertiert werden, da es sich bei `bamboo` um einen an den Rechner `rose` angeschlossenen PostScript-Drucker handelt.

10.4.3.2. Drucker mit direkter TCP-Schnittstelle

Wenn Sie eine Netzwerkkarte für Ihren Drucker kaufen, können Sie zwei verschiedene Versionen wählen: Eine Version, die ein Drucksystem emuliert (die teure Version), oder eine Version, die sich verhält, als wäre der Drucker an eine serielle oder parallele Schnittstelle angeschlossen (die billige Version). Dieser Abschnitt beschreibt die billige Variante. Bevorzugen Sie die teure Variante, sollten Sie den Abschnitt
Auf entfernten Rechnern installierte Drucker nochmals lesen.

Das Format der Datei `/etc/printcap` erlaubt es Ihnen, anzugeben, welche serielle oder parallele Schnittstelle verwendet werden soll und (falls Sie eine serielle Schnittstelle verwenden) welche Parameter (Baudrate, Flußkontrolle, Behandlung von Tabulatoren, Konvertierung von neuen Zeilen und andere mehr) Sie verwenden wollen. Es gibt allerdings keine Möglichkeit, eine Verbindung zu einem Drucker zu definieren, der einen TCP/IP- oder einem anderem Netzwerkport auf Druckaufträge hin abfragt.

Um Daten an einen Netzwerkdrucker zu schicken, müssen Sie daher ein Kommunikationsprogramm entwickeln, das von Text- und Konvertierungsfiltern aufgerufen werden kann. Dazu ein Beispiel. Das Skript `netprint` übernimmt alle Daten von der Standardeingabe und schickt sie an einen Netzwerkdrucker. `netprint` erwartet zwei Argumente: Als erstes Argument wird der Hostname des Druckers und als zweites Argument der Port, über den die Verbindung erfolgen soll, übergeben. Dabei handelt sich allerdings um eine Ein-Wege-Kommunikation (von FreeBSD zum Drucker). Viele Netzwerkdrucker unterstützen aber auch eine Zwei-Wege-Kommunikation, deren Vorteile (Abfrage des Druckerstatus, die Verrechnung von Druckaufträgen und andere mehr) Sie vielleicht nutzen wollen.

```
#!/usr/bin/perl
#
#  netprint - Textfilter für einen Netzwerkdrucker
#  Installiert unter:  /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

Dieses Skript kann für verschiedene Filter eingesetzt werden. Das folgende Beispiel verwendet den an ein Netzwerk angeschlossenen Zeilendrucker Diablo 750-N. Dieser Drucker empfängt zu druckende Daten auf dem Port 5100. Der Hostname des Druckers lautet `scrivener`. Daher sieht der Textfilter für diesen Drucker wie folgt aus:

```
#!/bin/sh
#
#  diablo-if-net - Textfilter für den Diablo-Drucker ‘scrivener’.
#  Drucker lauscht auf Port 5100.
#  Installiert unter:  /usr/local/libexec/diablo-if-net
#
exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint scrivener 5100
```

10.4.4. Den Druckerzugriff beschränken

Dieser Abschnitt beschreibt, wie Sie den Druckerzugriff beschränken können. Das **LPD**-Drucksystem erlaubt Ihnen die Kontrolle darüber, wer lokal oder über ein Netzwerk auf einen Drucker zugreifen darf, ob mehrere Kopien erstellt werden dürfen und wie groß Druckaufträge und Druckerwarteschlangen werden dürfen.

10.4.4.1. Den Ausdruck von mehreren Kopien verhindern

Das **LPD**-System macht es dem einzelnen Benutzer einfach, mehrere Kopien einer Datei zu drucken. So werden mit `lpr -#5` beispielsweise fünf Kopien jeder Datei des Druckauftrags erstellt. Ob dies gut oder schlecht ist, müssen Sie selbst entscheiden.

Wenn Sie der Meinung sind, dass multiple Kopien eine unnötige Beanspruchung Ihres Druckers darstellen, sollten Sie die `-#`-Opion von lpr(1) deaktivieren, indem Sie die `sc`-Fähigkeit in Ihre `/etc/printcap` aufnehmen. Verwendet ein Benutzer dennoch die Option `-#`, erhält er daraufhin folgende Meldung:

```
lpr: multiple copies are not allowed
```

Wenn Sie den Zugriff auf einen entfernten Drucker (wie in Abschnitt Auf entfernten Rechnern installierte Drucker beschrieben) konfiguriert haben, müssen Sie die `sc`-Fähigkeit auch auf den entfernten Rechnern, die auf Ihren Drucker zugreifen dürfen, in `/etc/printcap` eintragen, damit Benutzer diese Vorgabe nicht durch den Wechsel auf einen anderen Rechner umgehen können.

Dazu ein Beispiel. Es handelt sich dabei um die Datei `/etc/printcap` auf dem Rechner `rose`. Der Drucker `rattan` soll multiple Kopien zulassen, auf dem Laserdrucker `bamboo` sollen multiple Kopien hingegen nicht erlaubt sein, daher müssen wir für diesen Drucker die `sc`-Fähigkeit aktivieren:

```
#
#  /etc/printcap für den Rechner rose - multiple Kopien auf bamboo verbieten
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
```

```
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Außerdem müssen wir noch die `sc`-Fähigkeit in der Datei `/etc/printcap` des Rechners `orchid` aktivieren. Parallel dazu untersagen wir das Erstellen von multiplen Kopien auf dem Drucker `teak`:

```
#
#  /etc/printcap für den Rechner orchid - lokal machen wir keine multiplen Kopien
#  Lokaler Drucker teak oder entfernter Drucker bamboo:
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
        :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
        :if=/usr/local/libexec/ifhp:\
        :vf=/usr/local/libexec/vfhp:\
        :of=/usr/local/libexec/ofhp:

rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

Durch die Verwendung der `sc`-Fähigkeit ist zwar die Verwendung von `lpr -#` nicht mehr möglich, ein Benutzer kann aber weiterhin lpr(1) mehrmals hintereinander aufrufen oder eine Datei mehrfach in den gleichen Druckauftrag aufnehmen:

```
% lpr forsale.sign forsale.sign forsale.sign forsale.sign forsale.sign
```

Auch dieser Mißbrauch Ihres Druckers kann verhindert werden, falls Sie dies wünschen. Diese Maßnahmen werden in diesem Abschnitt allerdings nicht behandelt.

10.4.4.2. Den Zugriff auf bestimmte Drucker beschränken

Sie können angeben, wer auf welchem Drucker drucken darf, wenn Sie den Gruppenmechanismus von UNIX in Kombination mit der `rg`-Fähigkeit von `/etc/printcap` einsetzen. Weisen Sie dazu alle Benutzer, die auf einen Drucker zugreifen dürfen, einer gemeinsamen Gruppe zu und geben Sie diese Gruppe über die `rg`-Fähigkeit an.

Wenn Benutzer, die dieser Gruppe nicht angehören (dies gilt auch für `root`), werden diese durch die Meldung begrüsst, wenn Sie diesen Drucker verwenden wollen.

```
lpr: Not a member of the restricted group
```

Analog zur `sc`-Fähigkeit (*suppress multiple copies*) müssen Sie die `rg`-Fähigkeit auch auf allen entfernten Rechnern aktivieren, die auf Ihren Drucker zugreifen dürfen (lesen Sie dazu auch den Abschnitt
Auf entfernten Rechnern installierte Drucker).

Wollen wir beispielsweise allen Benutzern die Verwendung des Druckers `rattan`, aber nur Mitgliedern der Gruppe `artists` die Verwendung des Druckers `bamboo` erlauben, passen wir die bereits bekannte `/etc/printcap` des Rechners `rose` entsprechend an:

```
#
#  /etc/printcap des Rechners rose - Zugriffsbeschränkung für bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
```

```
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Die Datei `/etc/printcap` des Rechners `orchid` wird dadurch nicht beeinflusst. Jeder Benutzer des Rechners `orchid` kann also weiterhin den Drucker `bamboo` verwenden.

> **Anmerkung:** Für jeden Drucker kann nur eine einzige priviligierte Gruppe erstellt werden.

10.4.4.3. Die Größe von Druckaufträgen kontrollieren

Wenn Sie viele Benutzer haben, die Ihre Drucker verwenden dürfen, werden Sie wahrscheinlich eine Obergrenze für Dateien angeben wollen, die Benutzer an Ihren Drucker senden dürfen. Dies ist sinnvoll, weil Speicherplatz für Spooling-Verzeichnisse nur begrenzt verfügbar ist und Sie stets sicherstellen müssen, dass auch die Druckaufträge anderer Benutzer verarbeitet werden können.

LPD verwendet die `mx`-Fähigkeit, um die maximal erlaubte Größe von Dateien eines Druckauftrags anzugeben. Dieser Wert wird in 1.024 Bytes großen `BUFSIZ`-Blöcken angegeben. Setzen Sie diesen Wert auf Null, gibt es keine Größenbeschränkung. Existiert die `mx`-Fähigkeit hingegen überhaupt nicht, so gilt ein Limit von 1.000 Blöcken.

> **Anmerkung:** Diese Limits gelten nur für die Größe von *Dateien* innerhalb eines Druckauftrages, *nicht aber* für die Gesamtgröße des Druckauftrags.

LPD lehnt eine Datei auch dann nicht ab, wenn sie das Limit des Druckers überschreitet. Vielmehr wird die Datei bis zum Erreichen des Limits in die Warteschlange geladen, danach wird der Druck gestartet. Der das Limit überschreitende Rest wird hingegen verworfen und nicht gedruckt!

Mit diesem Wissen können wir nun Limits für die Drucker `rattan` und `bamboo` definieren. Da PostScript-Dateien der Gruppe `artists` in der Regel sehr groß sind, setzen wir ein Limit von fünf Megabytes. Für den Druck von normalen Text (auf dem Drucker `rattan`) setzen wir hingegen kein Limit:

```
#
#  /etc/printcap für den Rechner rose
#

#
#  Kein Größenlimit:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:mx#0:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:
```

```
#
#  Ein Limit von 5 Megabyte:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Auch diese Limits gelten nur für lokale Benutzer. Wenn Sie den Zugriff auf Ihren Drucker auch über ein Netzwerk erlauben wollen, unterliegen die Benutzer dieser Rechner diesen Limits nicht. Daher müssen Sie diese Limits über die `mx`-Fähigkeit auch in der `/etc/printcap` jedes Rechners definieren, der Ihren Drucker verwenden darf. Der Abschnitt Auf entfernten Rechnern installierte Drucker enthält weitere Informationen zum Drucken über ein Netzwerk.

Es gibt eine weitere Möglichkeit, um die Größe von Druckaufträgen von entfernten Rechnern zu beschränken. Lesen Sie dazu den Abschnitt Druckaufträge von entfernten Rechnern beschränken.

10.4.4.4. Druckaufträge von entfernten Rechnern beschränken

Das **LPD**-System bietet mehrere Möglichkeiten, um Druckaufträge zu beschränken, die auf entfernten Rechnern gestartet wurden:

Rechner beschränken

Sie können festlegen, von welchen entfernten Rechnern ein lokaler **LPD** Druckaufträge annimmt, indem Sie die Dateien `/etc/hosts.equiv` sowie `/etc/hosts.lpd` entsprechend anpassen. **LPD** überprüft diese Dateien, um festzustellen, ob ein Druckauftrag von einem Rechner stammt, der in einer dieser Dateien aufgeführt ist. Ist dies nicht der Fall, lehnt **LPD** den Druckauftrag ab.

Der Aufbau dieser Datei ist sehr einfach: Jede Zeile enthält einen einzigen Rechnernamen. Beachten Sie aber, dass `/etc/hosts.equiv` auch vom ruserok(3)-Protokoll benötigt wird und Änderungen dieser Datei auch Programme wie rsh(1) und rcp(1) beeinflussen können.

Das folgende Beispiel beschreibt die Datei `/etc/hosts.lpd` auf dem Rechner `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

Durch diese Vorgaben akzeptiert `rose` nur noch Druckaufträge von den Rechnern `orchid`, `violet`, und `madrigal.fishbaum.de`. Versucht ein anderer Rechner, auf den **LPD** von `rose` zuzugreifen, wird dieser Druckauftrag abgelehnt werden.

Größenbeschränkungen

Sie können festlegen, wieviel Speicherplatz auf dem Dateisystem, in dem das Spooling-Verzeichnis liegt, mindestens frei sein muss. Dazu erzeugen Sie im Spooling-Verzeichnis Ihres lokalen Druckers die Datei `minfree`. In dieser Datei geben Sie an, wieviele 512 Byte große Blöcke auf Ihrer Platte frei sein müssen, damit ein Druckauftrag von einem entfernten Rechner akzeptiert wird.

Durch diese Vorgabe können Sie sicherstellen, dass Benutzer von entfernten Rechnern Ihr Dateisystem nicht "zumüllen". Außerdem können Sie damit lokale Benutzer bevorzugen, da diese auch dann noch Druckaufträge erteilen dürfen, wenn der verfügbare Plattenplatz unter das in der Datei `minfree` definierte Limit gefallen ist.

Legen wir nun die Datei `minfree` für den Drucker `bamboo` an. Zuerst untersuchen wir `/etc/printcap`, um das Spooling-Verzeichnis für diesen Drucker zu finden. Das folgende Beispiel zeigt den Eintrag für den Drucker `bamboo`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
        :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
        :lp=/dev/ttyu5:ms#-parenb cs8 clocal crtscts:rw:mx#5000:\
        :if=/usr/local/libexec/psif:\
        :df=/usr/local/libexec/psdf:
```

Das Spooling-Verzeichnis wird über die `sd`-Fähigkeit festgelegt. Wir wollen, dass mindestens drei Megabyte (also 6144 Blöcke) freier Plattenplatz vorhanden sein müssen, damit **LPD** einen Druckauftrag von einem entfernten Rechner akzeptiert:

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

Benutzer beschränken

Sie können auch festlegen, welche entfernten Benutzer Ihren lokalen Drucker verwenden dürfen, indem Sie die `rs`-Fähigkeit in `/etc/printcap` definieren. Wenn für den Eintrag eines lokalen Druckers die `rs`-Fähigkeit definiert ist, akzeptiert **LPD** Druckaufträge von entfernten Rechnern nur dann, *wenn* der Benutzer, der den Druckauftrag gesendet hat, auch über ein gleichnamiges Benutzerkonto auf dem lokalen Rechner verfügt. Ist dies nicht der Fall, lehnt **LPD** den Druckauftrag ab.

Diese Fähigkeit ist besonders in Umgebungen nützlich, in denen beispielsweise verschiedene Abteilungen ein gemeinsames Netzwerk teilen, wobei einige Benutzer zu mehreren Abteilungen gehören. Haben diese Benutzer auch ein Benutzerkonto auf Ihrem System, so können sie Ihren Drucker auch von ihrer eigenen Abteilung aus nutzen. Wollen Sie zwar den Zugriff auf Ihren Drucker, *nicht aber* den Zugriff auf Ihre übrigen Ressourcen erlauben, können Sie für diese Benutzer einen sogenannten "Token-Account" ohne Heimatverzeichnis und mit einer nutzlosen Shell wie `/usr/bin/false` erstellen.

10.4.5. Die Druckernutzung verrechnen

Sie wollen die Nutzung Ihrer Drucker kostenpflichtig machen? Warum auch nicht? Papier und Tinte kosten Geld. Auch eine regelmäßige Wartung muss bezahlt werden. Nachdem Sie einen Preis festgelegt haben, den Sie für jede gedruckte Seite verrechnen wollen, stellt sich die Frage, wie Sie die Verrechnung der Druckkosten technisch umsetzen können.

Die schlechte Nachricht ist, dass das **LPD**-System dabei wenig hilfreich ist. Die Verrechnung von Druckaufträgen hängt stark vom verwendeten Drucker, den zu druckenden Dateiformaten und *Ihren* Anforderungen an die Verrechnung der Druckernutzung ab.

Um die Verrechnung der Druckernutzung zu implementieren, müssen Sie sowohl Ihre Textfilter (um den Druck von normalem Text abzurechnen) als auch Ihre Konvertierungsfilter (um den Druck sonstiger Formate abzurechnen) entsprechend anpassen, damit diese die Zahl der gedruckten Seiten ermitteln können. Leider können Sie dazu nicht einen einfachen Ausgabefilter verwenden, da diese die Verrechnung von Druckaufträgen nicht unterstützen. Weitere Informationen zu den verschiedenen Filterarten finden Sie im Abschnitt Filter.

Prinzipiell gibt es zwei Möglichkeiten, wie Sie diese Verrechnung umsetzen können:

- Die *periodische Verrechnung* wird häufiger verwendet, da sie einfacher zu implementieren ist. Wenn ein Druckauftrag ausgeführt wird, schreibt der Filter den Benutzer, den verwendeten Rechner sowie die Anzahl der gedruckten Seiten in eine Verrechnungsdatei. Nach einem zu definierenden Zeitraum werden diese Dateien ausgewertet, die Gesamtzahl der von einem Benutzer gedruckten Seiten bestimmt und dem jeweiligen Benutzer verrechnet. Danach werden alle Protokolldateien zurückgesetzt, und die Protokollierung beginnt von Neuem.
- Die *unmittelbare Verrechnung* wird nur selten eingesetzt, das sie schwieriger zu implementieren ist. Bei dieser Methode wird der Druckauftrag verrechnet, sobald der Drucker verwendet wird. Dadurch können Sie beispielsweise verhindern, dass ein Benutzer seine erlaubte "Druckquote" überschreitet. Zusätzlich können Sie es Ihren Benutzern erlauben, deren Druckquote abzufragen oder anzupassen. Allerdings benötigen Sie eine Datenbank, um Benutzer und deren Quoten verwalten zu können.

Das **LPD**-Drucksystem unterstützt beide Methoden. Allerdings müssen Sie die benötigen Filter sowie den zur Verrechnung nötigen Code selbst bereitstellen. Der Vorteil dabei ist allerdings, dass Sie in der Wahl Ihrer Verrechnungsmethode äußerst flexibel sind. So können Sie sich etwa für die periodische oder die unmittelbare Verrechnung entscheiden. Sie können festlegen, welche Informationen Sie erfassen wollen: Benutzernamen, Rechnernamen, die Art der Druckaufträge, die Anzahl der gedruckten Seiten, den Papierverbrauch, den Zeitaufwand für die Bearbeitung eines Druckauftrages und viele andere mehr. Dazu müssen Sie Ihre Filter entsprechend anpassen, damit diese Informationen erfassst und gespeichert werden.

10.4.5.1. Kurzanleitung für die Implementierung der Druckerverrechnung

FreeBSD bietet Ihnen zwei Programme, um eine periodische Verrechnung rasch zu implementieren. Dabei handelt es sich um den im Abschnitt lpf: Ein Textfilter behandelten Textfilter sowie um pac(8), ein Programm, mit dem Sie Einträge aus Verrechnungsdateien auslesen und aufsummieren können.

Wie bereits im Abschnitt Filter erwähnt, startet **LPD** den Text- oder Konvertierungsfilter mit dem Namen der Verrechnungsdatei als Argument. Dadurch weiß der Filter, in welche Datei er einen Verrechnungseintrag schreiben soll. Der Name dieser Datei wird über die `af`-Fähigkeit in `/etc/printcap` festgelegt. Falls die Datei nicht über einen absoluten Pfad angegeben wird, handelt es sich um einen Pfad relativ zum Spooling-Verzeichnis.

LPD startet `lpf` mit den Argumenten *page width und page length*, die über die `pw`- und `pl`-Fähigkeit definiert werden. Das Kommando `lpf` verwendet diese Argumente danach, um den Papierverbrauch zu bestimmen. Nachdem die Datei an den Drucker geschickt wurde, wird ein Verrechnungseintrag in die Verrechnungsdatei geschrieben. Ein solcher Eintrag sieht dabei ähnlich den folgenden aus:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

Sie sollten für jeden Drucker eine eigene Verrechnungsdatei verwenden, da `lpf` die Verrechnungsdatei nicht sperren kann. Sind also gleichzeitig zwei `lpf`-Instanzen aktiv, kann es dazu kommen, dass Ihre Verrechnungsdatei zerstört wird, wenn beide Instanzen gleichzeitig in die gleiche Datei schreiben. Damit für jeden Drucker eine eigene Verrechnungsdatei angelegt wird, fügen Sie den Eintrag `af=acct` in `/etc/printcap` ein. Dadurch wird für jeden Drucker eine separate Verrechnungsdatei mit dem Namen `acct` im Spooling-Verzeichnis des jeweiligen Druckers erzeugt.

Wenn Sie Ihre Daten erfasst haben und die entstandenen Kosten Ihren Benutzern verrechnen wollen, starten Sie pac(8). Dazu wechseln Sie in das Spooling-Verzeichnis des auszuwertenden Druckers und geben `pac` ein. Dadurch erhalten Sie eine Ausgabe ähnlich der folgenden:

```
  Login               pages/feet   runs    price
orchid:kelly               5.00    1   $  0.10
orchid:mary               31.00    3   $  0.62
orchid:zhang               9.00    1   $  0.18
rose:andy                  2.00    1   $  0.04
rose:kelly               177.00  104   $  3.54
rose:mary                 87.00   32   $  1.74
rose:root                 26.00   12   $  0.52

total                    337.00  154   $  6.74
```

Folgende Argumente können an pac(8) übergeben werden:

`-P`*`Drucker`*

Gibt an, welcher *`Drucker`* ausgewertet werden soll. Diese Option setzt voraus, dass für die `af`-Fähigkeit in `/etc/printcap` ein absoluter Pfad angegeben wurde.

`-c`

Sortiert die Ausgabe nach den verursachten Kosten anstelle einer alphabetischen Sortierung der Benutzernamen.

`-m`

Ignoriert den Rechnernamen in Verrechnungsdateien. Ist diese Option gesetzt, ist der Benutzer `smith` auf dem Rechner `alpha` mit dem Benutzer `smith` auf dem Rechner `gamma` identisch. Ist diese Option nicht gesetzt, handelt es sich um unterschiedliche Benutzer.

`-p`*`Preis`*

Berechnet die entstandenen Kosten aus dem *`Preis`* in Dollar pro Seite statt aus dem über die `pc`-Fähigkeit in `/etc/printcap` definierten Preis. In der Voreinstellung sind dies zwei Cent pro Seite. Sie können aber auch einen eigenen *`Preis`* in Form einer Gleitkommazahl angeben.

`-r`

Die Sortierreihenfolge umkehren.

`-s`

Die Verrechnungsdatei in einer neuen Datei aufsummieren und die originale Verrechnungsdatei zurücksetzen.

`name ...`

Verrechnungsinformationen nur für die angegebenen Benutzernamen ausgeben.

In der Voreinstellung gibt pac(8) aus, wieviele Seiten von welchem Benutzer auf welchem Rechner gedruckt wurden. Wenn Rechnernamen für Sie uninteressant sind (weil sich Benutzer beispielsweise auf jedem Rechner anmelden können), sollten Sie `pac -m` verwenden, um die folgende Ausgabe zu erhalten:

```
  Login               pages/feet   runs    price
andy                       2.00    1   $  0.04
kelly                    182.00  105   $  3.64
mary                     118.00   35   $  2.36
root                      26.00   12   $  0.52
zhang                      9.00    1   $  0.18
```

```
total                    337.00  154   $  6.74
```

Um den zu verrechnenden Betrag zu ermitteln, verwendet pac(8) die `pc`-Fähigkeit von `/etc/printcap` (Voreinstellung 200, dieser Wert entspricht 2 Cents). Geben Sie hier (als Hundertfaches des tatsächlichen Wertes) den Preis pro Seite an, den Sie verrechnen wollen. Sie können diesen Wert überschreiben, wenn Sie pac(8) mit der Option `-p` ausführen. Beachten Sie dabei aber, dass Sie in diesem Fall die Einheiten in Dollar angeben, und nicht als Hundertfaches des tatsächlichen Cent-Betrages. So steht

```
# pac -p1.50
```

beispielsweise für einen Preis von einem Dollar und fünfzig Cent pro Seite.

Der Aufruf von `pac -s` führt schließlich dazu, dass die aufsummierten Informationen in einer eigenen Auswertedatei gespeichert werden. Diese hat den gleichen Namen wie die Verrechnungsdatei, es wird lediglich ein `_sum` an den Dateinamen angehängt. Danach wird die Verrechnungsdatei zurückgesetzt. Wenn Sie pac(8) erneut aufrufen, wird die Auswertedatei eingelesen, um die Startbeträge zu erhalten, alle weiteren Informationen stammen danach aus der normalen Verrechnungsdatei.

10.4.5.2. Wie kann man die Anzahl der gedruckten Seiten ermitteln?

Um die Druckernutzung auch nur annähernd genau verrechnen zu können, müssen Sie ermitteln, wieviel Papier ein Druckauftrag verbraucht. Die Bestimmung dieses Wertes ist das zentrale Problem, das Sie lösen müssen, wenn Sie Druckaufträge kostenpflichtig machen wollen.

Normaler Text stellt in der Regel kein Problem dar: Sie zählen dazu nur die Zeilen des Druckauftrages und dividieren diesen Wert durch die Anzahl der Zeilen pro Seite, die Ihr Drucker bietet. Allerdings dürfen Sie dabei nicht vergessen, dass gelöschte Zeichen (*Backspaces*) Zeilen überschreiben. Außerdem können sich lange logische Zeilen (im Druckauftrag) über mehrere physikalische Zeilen (am Ausdruck) erstrecken.

Der im Abschnitt lpf: Ein Textfilter vorgestellte Textfilter `lpf` berücksichtigt diese Besonderheiten. Wenn Sie einen eigenen Textfilter für die Verrechnung der Druckernutzung schreiben wollen, sollten Sie sich daher den Quellcode von `lpf` näher ansehen.

Aber was ist mit anderen Dateiformaten?

Für die DVI-nach-LaserJet- oder für die DVI-nach-PostScript-Konvertierung können Sie die Protokolldateien von `dvilj` oder `dvips` auslesen, um festzustellen, wieviele Seiten konvertiert wurden. Die gleiche Methode könnte auch mit anderen Dateitypen funktionieren.

Alle diese Methoden haben aber das Problem, dass ein Drucker möglicherweise nicht alle Seiten des Druckauftrages drucken kann. So könnte es etwa zu einem Papierstau kommen, der Toner könnte zu Ende gehen oder es könnte ein Druckerdefekt auftreten – trotzdem würden alle Seiten des Druckauftrages verrechnet werden.

Was kann man dagegen tun?

Es gibt nur eine einzige *sichere* Methode, um die Druckernutzung *exakt* zu bestimmen. Besorgen Sie sich einen Drucker, der das verbrauchte Papier protokolliert und verbinden Sie ihn über eine serielle oder eine Netzwerkverbindung. Nahezu alle PostScript-Drucker, aber auch viele andere Modelle und Druckertypen (beispielsweise Laserdrucker von Imagen) sind dazu in der Lage. Passen Sie die Filter für diese Drucker entsprechend an, damit diese nach jedem Druckauftrag die Anzahl der gedruckten Seiten ermitteln und verrechnen Sie Druckaufträge *ausschließlich* über diesen Wert. Danach müssen Sie sich um die Anzahl der gedruckten Zeilen oder um mögliche Druckerprobleme nie mehr kümmern.

Sie können aber auch großzügig sein und alle Ausdrucke kostenlos abgeben.

10.5. Drucker verwenden

Übersetzt von Johann Kois.

Dieser Abschnitt beschreibt, wie Sie einen unter FreeBSD konfigurierten Drucker verwenden können. Die folgende Liste bietet einen Überblick über wichtige Anwenderbefehle:

lpr(1)

Einen Druckauftrag drucken

lpq(1)

Eine Druckerwarteschlange prüfen

lprm(1)

Einen Druckauftrag aus einer Warteschlange entfernen (stornieren)

Zusätzlich existiert mit lpc(8) ein Befehl zur zur Steuerung von Druckern und Druckerwarteschlangen, der im Abschnitt Drucker verwalten näher beschrieben wird.

Jeder der drei Befehle lpr(1), lprm(1), sowie lpq(1) akzeptiert die Option `-P` *`printer-name`*, mit der Sie den zu verwendenden Drucker (der dazu in `/etc/printcap` definiert sein muss) festlegen. Dadurch sind Sie in der Lage, Druckaufträge zu erstellen, zu stornieren, oder den Status Ihrer Druckaufträge zu überprüfen. Verwenden Sie die Option `-P` nicht, wird der in der Umgebungsvariable `PRINTER` definierte Drucker verwendet. Existiert diese Variable nicht, greifen diese Befehle auf den Drucker `lp` zurück.

Im Folgenden steht der Begriff *Standarddrucker* daher für den über die Umgebungsvariable `PRINTER` definierten Drucker, oder, falls diese Variable nicht existiert, für den Drucker `lp`.

10.5.1. Druckaufträge erstellen

Um eine Datei zu drucken, geben Sie folgenden Befehl ein:

```
% lpr filename ...
```

Dadurch wird jede angegebene Datei an den Standarddrucker geschickt. Wenn Sie keine Datei angeben, liest lpr(1) die zu druckenden Daten von der Standardeingabe. Um beispielsweise einige wichtige Systemdateien zu drucken, geben Sie folgenden Befehl ein:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

Um einen bestimmten Drucker auszuwählen, verwenden Sie:

```
% lpr -P printer-name filename ...
```

Das folgende Beispiel gibt eine ausführliche Liste aller im Arbeitsverzeichnis enthaltenen Dateien auf den Drucker `rattan` aus:

```
% ls -l | lpr -P rattan
```

Da keine Dateien an lpr(1) übergeben werden, liest `lpr` die zu druckenden Daten von der Standardeingabe, in unserem Fall also die Ausgabe des Befehls `ls -l`.

lpr(1) akzeptiert auch verschiedene Optionen zur Formatierung und Konvertierung von Dateien, zur Erzeugung von multiplen Ausdrucken und so weiter. Lesen Sie dazu den Abschnitt Druckoptionen.

10.5.2. Druckaufträge verwalten

Wenn Sie lpr(1) verwenden, werden alle zu druckenden Daten in ein Paket, den sogenannten “Druckauftrag”, gepackt und an **LPD** geschickt. Jeder Drucker verfügt über eine Druckerwarteschlange, in der Ihre Druckaufträge gemeinsam mit denen anderer Benutzer verbleiben, bis sie gedruckt werden können. Zuerst eintreffende Druckaufträge werden dabei auch zuerst gedruckt.

Um die Druckerwarteschlange des Standarddruckers anzuzeigen, verwenden Sie lpq(1). Wollen Sie einen anderen Drucker abfragen, müssen Sie die Option `-P` verwenden. Der Befehl

```
% lpq -P bamboo
```

zeigt so die Druckerwarteschlange des Druckers `bamboo` an. Dieser Befehl liefert eine Ausgabe ähnlich der folgenden:

```
bamboo is ready and printing
Rank   Owner    Job  Files                              Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv   88 bytes
2nd    kelly    10   (standard input)                   1635 bytes
3rd    mary     11   ...                                78519 bytes
```

Derzeit enthält die Warteschlange von `bamboo` drei Druckaufträge. Dem ersten Auftrag, der vom Benutzer kelly erstellt wurde, wurde die “Auftragsnummer (job number)” 9 zugewiesen. Analog erhält jeder Druckerauftrag eine eindeutige Nummer zugewiesen. Diese Nummern sind nur dann von Bedeutung, wenn Sie einen Druckauftrag stornieren wollen. Der Abschnitt Druckaufträge stornieren beschreibt, wie Sie dazu vorgehen.

Der Auftrag mit der Nummer 9 besteht aus zwei Dateien, mehrere an lpr(1) übergebene Dateien werden also als Teil eines (gemeinsamen) Druckauftrags betrachtet. Dieser Druckauftrag ist derzeit aktiv (beachten Sie den Status `active` in der Spalte “Rank”), wird also gerade gedruckt. Der zweite Auftrag besteht aus Daten, die von der Standardeingabe an lpr(1) übergeben wurden. Der dritte Auftrag wurde vom Benutzer `mary` erstellt. Er ist sehr viel größer als die anderen Aufträge. Da der Pfad der zu druckenden Datei aufgrund seiner Länge nicht in der Spalte “Files” Platz hat, werden von lpq(1) nur drei Punkte angezeigt.

Die erste Zeile der Ausgabe von lpq(1) ist ebenfalls sehr nützlich: Sie beschreibt den momentanen Druckerstatus (oder zumindest, was **LPD** denkt, dass der Drucker gerade macht).

lpq(1) unterstützt auch die Option `-l` zur Erstellung einer ausführlicheren Ausgabe. Die Eingabe von `lpq -l` erzeugt für unser obiges Beispiel die folgende Ausgabe:

```
waiting for bamboo to become ready (offline ?)
kelly: 1st                              [job 009rose]
       /etc/host.conf                    73 bytes
       /etc/hosts.equiv                  15 bytes

kelly: 2nd                              [job 010rose]
```

```
       (standard input)                 1635 bytes

mary: 3rd                               [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes
```

10.5.3. Druckaufträge stornieren

Mit lprm(1) können Sie einen Druckauftrag stornieren. Häufig ist lprm(1) auch noch in der Lage, einen bereits aktiven Auftrag abzubrechen, allerdings wird dabei in der Regel trotzdem ein Teil des Auftrages oder der gesamte Auftrag gedruckt.

Um einen Druckauftrag auf dem Standarddrucker zu stornieren, müssen Sie zuerst die Auftragsnummer über lpq(1) ermitteln. Danach geben Sie Folgendes ein:

```
% lprm Job-Nummer
```

Um einen Druckauftrag eines anderen Druckers zu stornieren, benötigen Sie wiederum die Option `-P`. Der folgende Befehl entfernt den Druckauftrag mit der Nummer 10 aus der Warteschlange des Druckers `bamboo`:

```
% lprm -P bamboo 10
```

lprm(1) unterstützt verschiedene Kurzbefehle:

lprm -

> Entfernt alle Druckaufträge (des Standarddruckers), die von Ihnen erstellt wurden.

lprm *user*

> Entfernt alle Druckaufträge (des Standarddruckers), die vom Benutzer *user* erstellt wurden. Der Superuser kann im Gegensatz zu einem normalen Benutzer auch Aufträge anderer Benutzer entfernen.

lprm

> Wenn Sie weder eine Auftragsnummer, einen Benutzernamen, noch die Option `-` angeben, entfernt lprm(1) den aktiven Druckauftrag auf dem Standarddrucker, falls dieser Auftrag von Ihnen erstellt wurde. Der Superuser kann hingegen jeden aktiven Druckauftrag abbrechen.

Verwenden Sie zusätzlich die Option `-P` zu den eben beschriebenen Kurzbefehlen, wenn Sie diese auf einen anderen Drucker als den Standarddrucker anwenden wollen. So entfernt der folgende Befehl beispielsweise alle Druckaufträge des aktuellen Benutzers aus der Druckerwarteschlange des Druckers `rattan`:

```
% lprm -P rattan -
```

> **Anmerkung:** Wenn Sie in einer Netzwerkumgebung arbeiten, erlaubt es lprm(1) Ihnen nur, Druckaufträge auf dem Rechner zu stornieren, auf dem sie erstellt wurden. Dies gilt selbst dann, wenn der gleiche Drucker auch auf anderen Rechnern des Netzwerks verfügbar ist. Die folgende Befehlsfolge veranschaulicht diesen Umstand:
>
> ```
> % lpr -P rattan myfile
> % rlogin orchid
> % lpq -P rattan
> Rank   Owner      Job  Files                          Total Size
> active seeyan     12    ...                           49123 bytes
> ```

```
2nd    kelly      13   myfile                          12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfA013rose dequeued
cfA013rose dequeued
```

10.5.4. Abseits von normalem Text: Druckoptionen

lpr(1) unterstützt verschiedene Optionen zur Formatierung von Text, zur Konvertierung von Grafik- und anderen Dateiformaten, zur Erzeugung von multiplen Kopien, zur Verwaltung von Druckaufträgen und andere mehr. Dieser Abschnitt beschreibt einige dieser Optionen.

10.5.4.1. Formatierungs- und Konvertierungsoptionen

Die folgenden lpr(1)-Optionen kontrollieren die Formatierung von in einem Druckauftrag enthaltenen Dateien. Verwenden Sie diese Optionen, wenn Ihr Druckauftrag keinen normalen Text enthält, oder wenn Sie normalen Text mit pr(1) formatieren wollen.

Der folgende Befehl druckt so beispielsweise eine DVI-Datei (des TEX-Satzsystems) namens *`fish-report.dvi`* auf dem Drucker `bamboo`:

```
% lpr -P bamboo -d fish-report.dvi
```

Diese Optionen gelten für jede Datei des Druckauftrags, daher ist es nicht möglich beispielsweise DVI- und ditroff-Dateien über den gleichen Druckauftrag zu drucken. Sie müssen diese Dateien vielmehr über getrennte Druckaufträge drucken, wobei Sie jeweils geeignete Konvertierungsoptionen verwenden.

> **Anmerkung:** Alle Optionen mit Ausnahme von `-p` und `-T` setzen einen installierten und für den jeweiligen Drucker konfigurierten Konvertierungsfilter voraus. So benötigt die Option `-d` den DVI-Konvertierungsfilter. Diese Filter werden im Abschnitt Konvertierungsfilter ausführlich beschrieben.

`-c`

Druckt cifplot-Dateien.

`-d`

Druckt DVI-Dateien.

`-f`

Druckt FORTRAN-Textdateien.

`-g`

Druckt Plot-Daten.

`-i` *`anzahl`*

Rückt die Ausgabe um *`anzahl`* Spalten ein, lassen Sie *`anzahl`* weg, wird der Text um 8 Spalten eingerückt. Beachten Sie aber, dass diese Option nicht mit allen Konvertierungsfiltern funktioniert.

Anmerkung: Zwischen der Option `-i` und der der Zahl darf dabei kein Leerzeichen stehen.

`-l`

Druckt Text inklusive vorhandener Steuerzeichen.

`-n`

Druckt ditroff-Dateien (geräteunabhängiges troff).

`-p`

Formatiert normalen Text mit pr(1), bevor der Ausdruck erfolgt.

`-T` *`titel`*

Verwende *`titel`* auf dem pr(1)-Deckblatt anstelle des Dateinamens. Diese Option ist nur wirksam, wenn sie gemeinsam mit der Option `-p` verwendet.

`-t`

Druckt troff-Daten.

`-v`

Druckt Rasterdaten.

Dazu ein Beispiel. Der folgende Befehl druckt eine formatierte Version der Manualpage zu ls(1) auf den Standarddrucker:

```
% zcat /usr/share/man/man1/ls.1.gz | troff -t -man | lpr -t
```

zcat(1) dekomprimiert den Quellcode der Manualpage ls(1) und reicht ihn an troff(1) weiter, das ihn formatiert und daraus GNU troff-Daten erzeugt. Diese werden wiederum an lpr(1) weitergereicht, das den Druckauftrag schließlich an **LPD** übergibt. Da die Option `-t` von lpr(1) verwendet wurde, konvertiert das Drucksystem die GNU troff-Daten zuvor in ein Format, das der Standarddrucker verstehen und ausgeben kann.

10.5.4.2. Druckaufträge verwalten

Die folgenden Optionen von lpr(1) weisen **LPD** an, den Druckauftrag auf verschiedene Art und Weise zu behandeln:

`-#` *`anzahl`*

Erzeugt *`anzahl`* Ausdrucke jeder im Druckauftrag enthaltenen Datei anstelle eines einzigen Exemplars. Diese Option kann von einem Administrator deaktiviert werden, um die Beanspruchung des Druckers zu verringern. Lesen Sie den Abschnitt Den Ausdruck von mehreren Kopien verhindern, wenn Sie diese Funktion benötigen.

Das folgende Beispiel druckt drei Kopien der Datei *parser.c*, gefolgt von drei Kopien von *parser.h* auf den Standarddrucker:

```
% lpr -#3 parser.c parser.h
```

-m

Verschickt eine E-Mail, nachdem der Druckauftrag beendet wurde. Verwenden Sie diese Option, sendet **LPD** Ihnen eine E-Mail, wenn es die Bearbeitung Ihres Druckauftrages abgeschlossen hat. Diese Nachricht enthält Informationen darüber, ob Ihr Auftrag erfolgreich erledigt wurde oder ob ein Fehler auftrat. Ist dies der Fall, wird meist noch angegeben, welcher Fehler auftrat.

-s

Kopiert die Dateien nicht in das Spooling-Verzeichnis, sondern verlinkt stattdessen symbolisch auf diese Dateien.

Wenn Sie einen umfangreichen Druckauftrag erstellen, werden Sie diese Option wahrscheinlich verwenden wollen. Einerseits sparen Sie dadurch Speicherplatz im Spooling-Verzeichnis (im schlimmsten Fall könnte Ihr Druckauftrag ansonsten das Dateisystem des Spooling-Verzeichnis zum Überlaufen bringen), andererseits sparen Sie dadurch auch Zeit, weil **LPD** die in Ihrem Druckauftrag enthaltenen Dateien nicht in das Spooling-Verzeichnis kopieren muss.

Da **LPD** in diesem Fall die Originaldateien verwendet, muss sichergestellt sein, dass diese nicht verändert werden, bevor der Ausdruck abgeschlossen ist.

> **Anmerkung:** Wenn Sie auf einen entfernten Drucker drucken, muss **LPD** die Dateien dennoch vom lokalen auf den entfernten Rechner kopieren. In diesem Fall spart die Option `-s` Speicherplatz lediglich im lokalen Spooling-Verzeichnis, nicht aber im entfernten. Dennoch ist diese Option auch in diesem Fall nützlich.

-r

Löscht die im Druckauftrag enthaltenen Dateien, nachdem sie in das Spooling-Verzeichnis kopiert oder unter Verwendung der Option `-s` gedruckt werden. Verwenden Sie diese Option daher nur mit äußerster Vorsicht!

10.5.4.3. Deckblatt-Optionen

Die folgenden lpr(1)-Optionen passen den Text an, der auf einem Deckblatt eines Druckauftrages ausgegeben wird. Wird die Ausgabe von Deckblättern auf dem Zieldrucker unterdrückt, bleiben diese Optionen wirkungslos. Lesen Sie den Abschnitt Deckblätter, wenn Sie diese Funktion benötigen.

-C *text*

Ersetzt den Rechnernamen auf dem Deckblatt durch *text*. Der Rechnername ist dabei in der Regel der Name des Rechners, auf dem der Druckauftrag erstellt wurde.

-J *text*

Ersetzt den Namen des Druckauftrages auf dem Deckblatt durch *text*. Der Name des Druckauftrages entspricht in der Regel dem Namen der ersten Datei des Druckauftrages oder `stdin`, wenn Sie die Standardeingabe an den Drucker weiterleiten.

-h

Verhindert den Ausdruck von Deckblättern.

> **Anmerkung:** Ob diese Option funktioniert, hängt von der Art und Weise ab, wie Deckblätter auf Ihrem System erzeugt werden. Lesen Sie den Abschnitt Deckblätter für weitere Informationen.

10.5.5. Drucker verwalten

Als Administrator Ihres Systems ist es Ihre Aufgabe, Drucker zu installieren, zu konfigurieren und zu testen. Um mit Ihrem Drucker zu kommunizieren, können Sie lpc(8) verwenden. Dadurch sind Sie in der Lage,

- Ihre Drucker zu starten und zu beenden.
- Die Warteschlangen Ihrer Drucker zu aktivieren und zu deaktivieren.
- Die Reihenfolge der Druckaufträge zu ändern.

Am Anfang dieses Abschnitts steht die Erklärung einiger Begriffe. Wenn ein Drucker *beendet* ist, wird der Inhalt seiner Warteschlange nicht gedruckt. Druckaufträge können zwar weiterhin erstellt werden, diese verbleiben aber solange in der Warteschlange, bis der Drucker wieder *gestartet* oder die Warteschlange gelöscht wird.

Ist eine Warteschlange *deaktiviert*, kann (mit Ausnahme von `root`) kein Benutzer mehr einen Druckauftrag erteilen. Ist die Warteschlange hingegen *aktiviert*, können Druckaufträge erteilt werden. Ist ein Drucker zwar *gestartet*, die Warteschlange hingegen *deaktiviert*, werden dennoch alle noch in der Warteschlange vorhandenen Druckaufträge gedruckt.

Im Allgemeinen benötigen Sie `root`-Rechte, um lpc(8) einsetzen zu können. Als normaler Benutzer erlaubt es Ihnen lpc(8) lediglich, den Druckstatus abzufragen und einen hängenden Drucker neu zu starten.

Es folgt nun eine Zusammenfassung der Befehle von lpc(8). Die meisten dieser Befehle benötigen das Argument *printer-name*, mit dem Sie angeben, auf welchen Drucker der Befehl angewendet werden soll. Wenn Sie für *printer-name* `all` angeben, wird der Befehl auf alle in `/etc/printcap` definierten Drucker angewendet.

`abort` *printer-name*

Bricht den aktuellen Druckauftrag ab und beendet den Drucker. Solange die Warteschlange aktiviert ist, können allerdings weiterhin Druckaufträge erteilt werden.

`clean` *printer-name*

Entfernt veraltete Dateien aus dem Spooling-Verzeichnis des Druckers, da diese manchmal nicht vollständig von **LPD** entfernt werden können. Dies ist insbesondere dann der Fall, wenn während der Bearbeitung des

Druckauftrages Fehler auftraten. Dieser Befehl sucht dabei nach Dateien, die nicht in das Spooling-Verzeichnis gehören und entfernt diese.

`disable` *`printer-name`*

Deaktiviert die Annahme neuer Druckaufträge. Solange der Drucker nicht beendet wird, werden weiterhin alle in der Warteschlange enthaltenen Auftrage bearbeitet und gedruckt. `root` kann jederzeit Druckaufträge erstellen, selbst dann, wenn die Druckerwarteschlange deaktiviert ist.

Dieser Befehl ist besonders nützlich, wenn Sie einen neuen Drucker testen müssen oder einen neuen Filter installiert haben. Dazu deaktivieren Sie die Warteschlange des Druckers und erstellen Ihre Druckaufträge als `root`. Andere Benutzer können erst dann einen Druckauftrag erstellen, wenn Sie Ihre Tests abgeschlossen haben und die Druckerwarteschlange mit `enable` wieder reaktivieren.

`down` *`printer-name nachricht`*

Beendet einen Drucker. Äquivalent zu `disable`, gefolgt von `stop`. Die von Ihnen definierte *`nachricht`* wird als Druckerstatus angezeigt, wenn ein Benutzer die Warteschlange des Druckers mit lpq(1) oder mit `lpc status` abfragt.

`enable` *`printer-name`*

Aktiviert die Warteschlange eines Druckers. Erteilte Druckaufträge können zwar erteilt werden, diese werden aber nur dann gedruckt, wenn der Drucker auch gestartet ist.

`help` *`command-name`*

Ausgaben von hilfreichen Informationen zu *`command-name`*. Wird kein *`command-name`* angegeben, wird die Liste der verfügbaren Befehle ausgegeben.

`restart` *`printer-name`*

Startet den Drucker. Normale Benutzer können diesen Befehl verwenden, um einen hängenden **LPD** zu reaktivieren, sie sind allerdings nicht berechtigt, einen Drucker zu starten, der mit `stop` oder `down` beendet wurde. Dieser Befehl ist äquivalent zu `abort`, gefolgt von `start`.

`start` *`printer-name`*

Startet den Drucker, um die in der Warteschlange enthaltenen Aufträge zu drucken.

`stop` *`printer-name`*

Beendet den Drucker. Der Drucker beendet den aktiven Druckauftrag noch, danach wird kein weiterer in der Warteschlange enthaltener Auftrag gedruckt. Obwohl der Drucker beendet wurde, können weiterhin Druckaufträge erteilt werden, solange die Warteschlange nicht deaktiviert wurde.

`topq` *`printer-name job-or-username`*

Sortiert die Druckerwarteschlange des Druckers *`printer-name`* um, wobei der Auftrag mit der angegebenen *`Auftragsnummer`*, oder Druckaufträge, die von *`username`* erstellt wurden, an den Beginn der Warteschlange gesetzt werden. Für diesen Befehl kann die Option `all` nicht als *`printer-name`* verwendet werden.

`up` *`printer-name`*

Startet einen Drucker. Das Gegenstück zu `down`. Äquivalent zu `start`, gefolgt von `enable`.

lpc(8) akzeptiert diese Befehle direkt auf der Kommandozeile. Geben Sie keinen Befehl ein, wird lpc(8) im interaktiven Modus gestartet. In diesem Modus können Sie solange Befehle eingeben, bis Sie `exit` oder `quit` eingeben.

10.6. Alternativen zum LPD-Drucksystem

Wenn Sie dieses Kapitel bis hierher gelesen haben, wissen Sie so gut wie alles über **LPD**, das Standarddrucksystem von FreeBSD. Wahrscheinlich sind Ihnen bereits einige Unzulänglichkeiten dieses Systems aufgefallen, und Sie fragen sich nun, welche anderen Drucksysteme es für FreeBSD gibt.

LPRng

> **LPRng** steht für "LPR: the Next Generation". Dabei handelt es sich um eine von Grund auf neu geschriebene Version von PLP. LPRng wurde von Patrick Powell und Justin Mason, dem Hauptmaintainer von PLP, entwickelt. Die offizielle Webseite von **LPRng** ist unter http://www.lprng.org/ zu finden.

CUPS

> **CUPS**, das *Common UNIX Printing System*, stellt eine portable Abstraktionsschicht dar, die das Drucken auf allen UNIX-artigen Betriebsystemen ermöglicht. **CUPS** wurde von Easy Software entwickelt, um UNIX-Herstellern und -Benutzern eine einheitliche Standardlösung für den Druck von Dokumenten zu bieten.
>
> **CUPS** verwendet das Internet Printing Protocol (IPP), um Druckaufträge und -warteschlangen zu verwalten. Zusätzlich werden die Protokolle *Line Printer Daemon* (LPD), *Server Message Block* (SMB), und *AppSocket/JetDirect*), unterstützt, wenn auch nur mit eingeschränkter Funktionalität. Ausserdem ermöglicht **CUPS** das Auffinden von Netzwerkdruckern sowie die Verwendung auf *PostScript Printer Description* (PPD) basierender Druckoptionen.
>
> Die offizielle Webseite von **CUPS** ist http://www.cups.org/.

HPLIP

> **HPLIP**, das HP Linux Imaging and Printing System, ist eine von HP entwickelte Sammlung von Programmen, die Unterstützung für das drucken, scannen und faxen bei HP-Geräten bieten. Diese Programm-Sammlung verwendet **CUPS** als Grundlage für einige seiner Druck-Eigenschaften.
>
> Die Hauptseite für **HPLIP** ist http://hplipopensource.com/hplip-web/index.html.

10.7. Problembehandlung

Wenn Sie eine einfache Testseite mit lptest(1) gedruckt haben, könnte eines der folgenden Probleme aufgetreten sein:

Der Druck hat erst mit einer gewissen Verzögerung geklappt oder das bedruckte Blatt verblieb im Drucker, so als wäre der Druckvorgang noch nicht abgeschlossen.

Die Testseite wurde zwar gedruckt, danach tat sich allerdings nichts mehr. Vielleicht mussten Sie sogar eine Taste Ihres Druckers, etwa PRINT REMAINING oder FORM FEED drücken, damit der Druckvorgang fortgesetzt wurde.

Wenn das der Fall ist, hat der Drucker vermutlich vor dem eigentlichen Drucken gewartet, ob noch weitere Daten für Ihren Druckauftrag gesendet werden. Um dieses Problem zu beheben, können Sie den Textfilter anweisen, ein *Form Feed* -Zeichen (oder ein anderes entsprechendes Zeichen) an den Drucker zu senden. Dies reicht für gewöhnlich aus, um den Drucker zum Druck des noch im internen Puffer verbliebenen Textes zu bewegen. Dadurch kann auch sichergestellt werden, dass jeder neue Druckauftrag auf einer neuen Seite beginnt.

Der folgende Ersatz für das Shell-Skript `/usr/local/libexec/if-simple` gibt ein “Form Feed” aus, nachdem der Auftrag an den Drucker geschickt wurde:

```
#!/bin/sh
#
# if-simple - Einfacher Eingabefilter für lpd
# Installiert unter /usr/local/libexec/if-simple
#
# Kopiert stdin einfach nach stdout. Ignoriert alle Filter-Argumente.
# Schreibt ein Form-Feed-Zeichen (\f) nach dem Ende des Druckauftrages.

/bin/cat && printf "\f" && exit 0
exit 2
```

Der Drucker erzeugte einen “Treppeneffekt” (*staircase effect*).

Sie haben einen Ausdruck ähnlich dem folgenden erhalten:

```
!"#$%&'()*+,-./01234
                "#$%&'()*+,-./012345
                              #$%&'()*+,-./0123456
```

Sie sind zu einem weiteren Opfer des *Treppeneffekts* geworden. Verursacht wird dieser Effekt durch unterschiedliche Ansichten darüber, welche Zeichen den Beginn einer neuen Zeile anzeigen sollen. UNIX-ähnliche Betriebssysteme verwenden dafür ein einzelnes Zeichen: ASCII-Code 10, auch als *Line Feed* (LF) bekannt. MS-DOS, OS/2® und andere Betriebssysteme verwenden stattdessen ein Zeichenpaar: ASCII-Code 10 *und* ASCII-Code 13, *Carriage Return* (CR). Viele Drucker verwenden in der Voreinstellung die Konvention von MS-DOS, um Zeilenumbrüche darzustellen.

Wenn Sie unter FreeBSD drucken, wird nur das Zeichen *Line Feed* verwendet. Der Drucker erkennt dieses Zeichen und erweitert den Druckbereich um eine Zeile, verbleibt zum Druck des nächsten Zeichens aber in derselben horizontalen Position. Das ist der Grund für die Verwendung des *Carriage Return*: Es setzt die Position für das folgende Zeichen auf den linken Rand der Seite.

FreeBSD erwartet von einem Drucker das folgende Verhalten:

Drucker empfängt CR	Drucker druckt CR
Drucker empfängt LF	Drucker druckt CR + LF

Es gibt mehrere Möglichkeiten, dieses Verhalten zu erreichen:

- Verändern Sie die Konfiguration Ihres Druckers, um die Interpretation dieser Zeichen zu verändern. Lesen

Sie Ihr Druckerhandbuch, wenn Sie nicht wissen, was Sie dazu tun müssen.

> **Anmerkung:** Wenn Sie auf Ihrem Rechner neben FreeBSD noch andere Betriebssysteme verwenden, müssen Sie Ihren Drucker möglicherweise anschließend *erneut konfigurieren*, damit die Zeichen CR und LF unter diesen Systemen korrekt interpretiert werden. Ist dies bei Ihnen der Fall, werden Sie wohl eine der folgenden Lösungen bevorzugen.

- Lassen Sie LF durch den Treiber der seriellen Schnittstelle automatisch in CR+LF konvertieren. Selbstverständlich funktioniert dies nur mit Druckern, die an einer seriellen Schnittstelle angeschlossen sind. Um diese Möglichkeit zu nutzen, müssen Sie die `ms#`-Fähigkeit verwenden und in `/etc/printcap` den `onlcr`-Modus für den Drucker aktivieren.
- Senden Sie eine *Escape-Sequenz* an den Drucker, damit das Zeichen LF zeitweilig anders behandelt wird. Suchen Sie im Handbuch Ihres Druckers nach den von Ihrem Drucker unterstützten Escape-Sequenzen. Wenn Sie eine entsprechenden Escape-Sequenz finden, müssen Sie den Textfilter so anpassen, dass zuerst die Escape-Sequenz und anschließend der Druckauftrag gesendet wird.

  Es folgt nun ein Bespieltextfilter für einen Drucker, der die Hewlett Packard PCL Escape-Sequenzen versteht. Dieser Filter veranlasst den Drucker, LF-Zeichen als Folgen von LF+CR aufzufassen. Anschließend wird der Druckauftrag gesendet. Als Abschluss wird ein *Form Feed* gesendet, um die letzte Seite des Druckauftrags auszuwerfen. Dieses Beispiel sollte mit nahezu allen Druckern von Hewlett Packard funktionieren.

  ```
  #!/bin/sh
  #
  # hpif - Einfacher Text-Eingabefilter für lpd für auf HP-PCL basierende Drucker
  # Installiert unter /usr/local/libexec/hpif
  #
  # Kopiert stdin einfach nach stdout.  Ignoriert alle Filterargumente.
  # Weist den Drucker an LF als CR+LF zu interpretieren.
  # Wirft die Seite nach dem Drucken aus.

  printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
  exit 2
  ```

  Das nächste Beispiel aus `/etc/printcap` beschreibt den Rechner `orchid`, an dessen Parallelport ein Drucker angeschlossen ist. Es handelt sich dabei um einen Hewlett Packard LaserJet 3Si, der den Namen `teak` verwendet. Als Textfilter wird das Skript aus dem letzten Beispiel verwendet:

  ```
  #
  #  /etc/printcap für den Rechner orchid
  #
  teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
          :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
          :if=/usr/local/libexec/hpif:
  ```

Alle Zeilen wurden in die gleiche Zeile gedruckt.

Der Drucker hat niemals eine neue Zeile begonnen. Alle Zeilen des Textes wurden in eine einzige Zeile gedruckt.

Dieses Problem ist das “Gegenteil” des oben beschriebenen Treppeneffekts und kommt wesentlich seltener vor. Die von FreeBSD zum Abschluss einer Zeile benutzten LF-Zeichen werden als CR-Zeichen interpretiert. Dadurch wird die Druckposition zwar auf den linken Rand der Seite, aber nicht um eine Zeile nach unten gesetzt.

Konfigurieren Sie Ihren Drucker, um die folgende Interpretation der Zeichen LF und CR zu erzwingen:

Drucker empfängt	Drucker druckt
CR	CR
LF	CR + LF

Manche Zeichen wurden nicht gedruckt.

Der Drucker hat in jeder Zeile einige Zeichen nicht gedruckt. Vielleicht ist das Problem auch während des Druckens schlimmer geworden, und der Drucker hat immer mehr Zeichen nicht gedruckt.

Dieses Problem entsteht, weil der Drucker mit der Geschwindigkeit, mit der die Daten über die serielle Schnittstelle (an einer parallelen Schnittstelle sollte das Problem nicht auftreten) eintreffen, nicht mithalten kann. Es gibt zwei Möglichkeiten, dieses Problem zu lösen:

- Wenn der Drucker die Flusskontrolle mit XON/XOFF unterstützt, können Sie in der `ms#`-Fähigkeit den `ixon`-Modus aktivieren.
- Unterstützt der Drucker die Anfrage zum Senden/Löschen des Sende-Hardware-Handshakes (allgemein bekannt als `RTS/CTS`, dann sollten Sie den `crtscts`-Modus in der `ms#`-Fähigkeit aktivieren. Stellen Sie aber sicher, dass das verwendete Druckerkabel auch für die Hardware-Flusskontrolle geeignet ist.

Es wurden nur wirre Zeichen gedruckt.

Anstelle des gewünschten Textes wurden nur zufällige Zeichen gedruckt.

Dieses Problem wird ebenfalls durch falsche Konfigurationsparameter im Zusammenhang mit einem seriellen Drucker verursacht. Kontrollieren Sie die bps-Rate in der `br`-Fähigkeit und die Paritätseinstellung (*Parity*) in der `ms#`-Fähigkeit. Überprüfen Sie außerdem, ob der Drucker auch tatsächlich die gleichen Einstellungen verwendet, die in `/etc/printcap` definiert wurden.

Der Drucker hat überhaupt nicht reagiert.

Wenn gar nichts passiert ist, dann liegt das vermutlich an FreeBSD und nicht am Drucker. Aktivieren Sie die Protokollierung (`lf`-Fähigkeit) für den entsprechenden Drucker in der Datei `/etc/printcap`. Es folgt nun ein Beispieleintrag für den Drucker `rattan`, bei dem die `lf`-Fähigkeit aktiviert wurde.

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
        :sh:sd=/var/spool/lpd/rattan:\
        :lp=/dev/lpt0:\
        :if=/usr/local/libexec/if-simple:\
        :lf=/var/log/rattan.log
```

Versuchen Sie jetzt noch einmal zu drucken. Überprüfen Sie die Protokolldatei (in unserem Beispiel `/var/log/rattan.log`) auf etwaige Fehlermeldungen. Versuchen Sie aufgrund dieser Meldungen, das Problem zu beheben.

Wenn Sie keine Protokolldatei festlegen, verwendet **LPD** in der Voreinstellung `/dev/console` für die Ausgabe der Fehlermeldungen.

Kapitel 11. Linux-Binärkompatibilität

Restrukturiert und teilweise aktualisiert von Jim Mock. Beigetragen von Brian N. Handy und Rich Murphey. Übersetzt von Johann Kois.

11.1. Übersicht

FreeBSD bietet Binärkompatibilität zu verschiedenen anderen UNIX Betriebssystemen, darunter auch Linux. Nun könnten Sie sich fragen, warum FreeBSD in der Lage sein muss, Linux-Binärprogramme auszuführen? Die Antwort auf diese Frage ist sehr einfach. Viele Unternehmen und Entwickler programmieren bzw. entwickeln nur für Linux, da es “das Neueste und Beste” in der Computerwelt ist. Für uns FreeBSD-Anwender heißt dies, genau diese Unternehmen und Entwickler zu bitten, FreeBSD-Versionen ihrer Programme herauszubringen. Das Problem dabei ist nur, dass die meisten dieser Firmen trotzdem nicht erkennen, wie viele zusätzliche Anwender ihre Produkte benutzen würden, wenn es auch FreeBSD-Versionen gäbe, und daher weiterhin ausschließlich für Linux entwickeln. Was also kann ein FreeBSD-Anwender tun? Genau an diesem Punkt kommt die Linux- Binärkompatibilität ins Spiel.

Um es auf den Punkt zu bringen, genau diese Kompatibilität erlaubt es FreeBSD-Anwendern, etwa 90 % aller Linux-Anwendungen ohne Code-Änderungen zu verwenden. Dies schließt solche Anwendungen wie **StarOffice**, **Open Office**, die Linux-Versionen von **Netscape**, **Adobe Acrobat**, **RealPlayer**, **Oracle**, **WordPerfect®**, **Doom**, **Quake** und viele andere ein. Es wird sogar berichtet, dass diese Linux-Anwendungen in manchen Fällen unter FreeBSD eine bessere Leistung als unter Linux aufweisen.

Allerdings gibt es nach wie vor einige Linux-spezifische Betriebssystem-Eigenschaften, die unter FreeBSD nicht unterstützt werden. Linux-Anwendungen, die i386-spezifische Aufrufe (wie die Aktivierung des virtuellen 8086-Modus) verwenden, funktionieren unter FreeBSD leider nicht.

Nach dem Lesen dieses Kapitels werden Sie

- wissen, wie Sie die Linux-Binärkompatibilität installieren bzw. aktivieren.
- Wissen, wie man zusätzliche Linux-Systembibliotheken unter FreeBSD installiert.
- Linux-Anwendungen unter FreeBSD installieren können.
- Wissen, wie die Linux-Binärkompatibilität unter FreeBSD verwirklicht wurde.

Bevor Sie dieses Kapitel lesen, sollten Sie

- wissen, wie man Software Dritter installiert (Kapitel 5).

11.2. Installation

Die Linux-Binärkompatibilität ist per Voreinstellung nicht aktiviert. Der einfachste Weg, dies zu tun, ist das `Linux` KLD (“Kernel LoaDable object”) zu laden. Dies erreichen Sie durch die Eingabe des folgenden Befehls:

```
# kldload linux
```

Wollen Sie die Linux-Binärkompatibilität dauerhaft aktivieren, sollten Sie die folgende Zeile in `/etc/rc.conf` einfügen:

```
linux_enable="YES"
```

Der kldstat(8)-Befehl kann benutzt werden, um festzustellen, ob KLD geladen wurde:

```
% kldstat
Id Refs Address    Size     Name
 1    2 0xc0100000 16bdb8   kernel
 7    1 0xc24db000 d000     linux.ko
```

Wenn Sie das KLD nicht laden können oder wollen, besteht auch die Möglichkeit, die Linux-Binärkompatibiltät statisch in den Kernel einzubinden. Dazu fügen Sie Ihrer Kernelkonfigurationsdatei den Eintrag `options COMPAT_LINUX` hinzu. Anschließend installieren Sie Ihren neuen Kernel wie in Kapitel 9 beschrieben.

11.2.1. Linux-Laufzeitbibliotheken installieren

Dies kann auf zwei Arten geschehen, entweder über den linux_base-Port oder durch manuelle Installation der Bibliotheken.

11.2.1.1. Installation unter Verwendung des linux_base-Ports

Dies ist die einfachste Methode, um die Laufzeitbibliotheken zu installieren. Sie funktioniert genauso wie die Installation eines beliebigen anderen Ports aus der Ports-Sammlung (/usr/ports/). Dazu machen Sie einfach folgendes:

```
# cd /usr/ports/emulators/linux_base-f10
# make install distclean
```

> **Anmerkung:** Bei FreeBSD-Systemen vor FreeBSD 8.0 müssen Sie den Port `emulators/linux_base-fc4` anstatt `emulators/linux_base-f10` installieren.

Sie sollten nun über eine funktionierende Linux-Binärkompatibilität verfügen. Einige Programme könnten sich zwar über falsche Unterversionsnummern der Systembibliotheken beschweren, dies ist im Allgemeinen aber kein Problem.

> **Anmerkung:** Unter Umständen gibt es mehrere Versionen des Ports `emulators/linux_base`. Die Ports entsprechen unterschiedlichen Versionen verschiedener Linux-Distributionen Sie sollten den Port installieren, der am besten die Anforderungen der Linux-Anwendung erfüllt.

11.2.1.2. Manuelle Installation der Bibliotheken

Wenn Sie die “Ports”-Sammlung nicht installiert haben, können Sie die Bibliotheken auch manuell installieren. Dazu brauchen Sie die jeweiligen Linux-Systembibliotheken, die das zu installierende Programm verwendet sowie den Laufzeit-Linker. Zusätzlich müssen Sie auf Ihrem FreeBSD-System einen “virtuellen” Verzeichnisbaum für die Linux-Bibliotheken einrichten. Alle unter FreeBSD gestarteten Linux-Programme suchen zuerst in diesem Verzeichnisbaum nach Systembibliotheken. Wenn also ein Linuxprogramm beispielsweise `/lib/libc.so` lädt, versucht FreeBSD zuerst, `/compat/linux/lib/libc.so` laden. Ist diese Datei nicht vorhanden, wird

`/lib/libc.so` geladen. Systembibliotheken sollten daher besser in den “virtuellen” Verzeichnisbaum `/compat/linux/lib` als in den vom Linux-`ld.so` vorgeschlagenen installiert werden.

Im Allgemeinen müssen Sie nur zu Beginn nach den Systembibliotheken suchen, die von Linuxprogrammen benötigt werden. Nach den ersten Installationen von Linuxprogrammen auf Ihrem FreeBSD-System verfügen Sie über eine Sammlung von Linux-Systembibliotheken, die es Ihnen ermöglichen wird, neue Linuxprogramme ohne Zusatzarbeit zu installieren.

11.2.1.3. Installation zusätzlicher Systembibliotheken

Was passiert, wenn Sie den `linux_base`-Port installieren, und Ihr Programm beschwert sich trotzdem über fehlende Systembibliotheken? Woher wissen Sie, welche Systembibliotheken von Linux-Binärprogrammen benötigt werden, und wo Sie diese finden? Grundsätzlich gibt es dafür zwei Möglichkeiten (um dieser Anleitung zu folgen, müssen Sie unter FreeBSD als Benutzer `root` angemeldet sein):

Wenn Sie Zugriff auf ein Linux-System haben, können Sie dort nachsehen, welche Systembibliotheken eine Anwendung benötigt, und diese auf Ihr FreeBSD-System kopieren. Dazu folgendes Beispiel:

Nehmen wir an, Sie haben FTP verwendet, um die Linux-Binärversion von **Doom** zu bekommen und haben sie auf Ihrem Linux-System installiert. Nun können Sie überprüfen, welche Systembibliotheken das Programm benötigt, indem Sie `ldd linuxdoom` eingeben. Das Resultat sieht dann so aus:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

Sie müssten nun alle Dateien aus der letzten Spalte kopieren und sie unter `/compat/linux` speichern, wobei die Namen der ersten Spalte als symbolische Links auf diese Dateien zeigen. Damit haben Sie schließlich folgende Dateien auf Ihrem FreeBSD-System:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

> **Anmerkung:** Beachten Sie, dass wenn Sie bereits eine Linux-Systembibliothek einer zur ersten Spalte passenden Hauptversionsnummer (laut `ldd`-Ausgabe) besitzen, Sie die Datei aus der zweiten Spalte nicht mehr kopieren müssen, da die bereits vorhandene Version funktionieren sollte. Hat die Systembibliothek jedoch eine neuere Versionsnummer, sollten Sie sie dennoch kopieren. Sie können die alte Version löschen, solange Sie einen symbolischen Link auf die neue Version anlegen. Wenn Sie also folgende Bibliotheken auf Ihrem System installiert haben:
>
> ```
> /compat/linux/lib/libc.so.4.6.27
> /compat/linux/lib/libc.so.4 -> libc.so.4.6.27
> ```
>
> und Sie haben eine neue Binärdatei, die laut `ldd` eine neuere Bibliothek benötigt:
>
> ```
> libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
> ```
>
> Wenn diese sich nur um ein oder zwei Stellen in der Unterversionsnummer unterscheiden, müssen Sie `/lib/libc.so.4.6.29` nicht auf Ihr System kopieren, da das Programm auch mit der etwas älteren Version

ohne Probleme funktionieren sollte. Wenn Sie wollen, können Sie `libc.so` aber dennoch ersetzen (das heißt aktualisieren), was dann zu folgender Ausgabe führt:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Anmerkung: Der Mechanismus der symbolischen Links wird *nur* für Linux-Binärdateien benötigt. Der FreeBSD-Laufzeitlinker sucht sich die passenden Hauptversionsnummern selbst, das heißt Sie müssen sich nicht darum kümmern.

11.2.2. Linux ELF-Binärdateien installieren

ELF-Binärdateien benötigen manchmal eine zusätzliche "Kennzeichnung". Wenn Sie versuchen, eine nicht gekennzeichnete ELF-Binärdatei auszuführen, werden Sie eine Fehlermeldung ähnlich der folgenden erhalten:

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

Damit der FreeBSD-Kernel eine Linux-ELF-Datei von einer FreeBSD-ELF-Datei unterscheiden kann, gibt es das Werkzeug brandelf(1).

```
% brandelf -t Linux my-linux-elf-binary
```

Die GNU Werkzeuge schreiben nun automatisch die passende Kennzeichnungsinformation in die ELF-Binärdateien, so dass Sie diesen Schritt in Zukunft nur noch selten benötigen werden.

11.2.3. Installieren einer beliebigen RPM-basierten Linuxanwendung

FreeBSD besitzt seine eigene Paketdatenbank und diese wird dazu verwendet, um alle Ports (auch Linux-Ports) zu verfolgen. Deshalb wird die Linux RPM-Datenbank nicht benutzt (fehlende Unterstützung).

Falls Sie jedoch eine beliebige RPM-basierte Linux-Anwendung installieren wollen, erreichen Sie das mittels:

```
# cd /compat/linux
# rpm2cpio -q < /path/to/linux.archive.rpm | cpio -id
```

Benutzen Sie dann brandelf auf die installierten ELF-Binärdateien (nicht die Bibliotheken!). Sie werden keine saubere Deinstallation hinbekommen, aber evtl. helfen ein paar Tests weiter.

11.2.4. Namensauflösung konfigurieren

Wenn DNS nicht funktioniert, oder Sie folgende Fehlermeldung erhalten:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

müssen sie `/compat/linux/etc/host.conf` wie folgt anlegen:

```
order hosts, bind
multi on
```

Diese Reihenfolge legt fest, dass zuerst `/etc/hosts` und anschließend DNS durchsucht werden. Wenn `/compat/linux/etc/host.conf` nicht vorhanden ist, finden Linux-Anwendungen FreeBSD's `/etc/host.conf` und beschweren sich über die inkompatible FreeBSD-Syntax. Wenn Sie keinen Nameserver (in `/etc/resolv.conf`) konfiguriert haben, sollten Sie den Eintrag `bind` entfernen.

11.3. Mathematica® installieren

Für Mathematica 5.x aktualisiert von Boris Hollas.

Dieses Dokument beschreibt die Installation der Linux-Version von **Mathematica 5.x** auf einem FreeBSD-System.

Die Linux-Version von **Mathematica** oder **Mathematica für Studenten** kann direkt von Wolfram unter http://www.wolfram.com/ bestellt werden.

11.3.1. Den Mathematica-Installer starten

Zuerst müssen Sie FreeBSD mitteilen, dass die Linux-Binärversion von **Mathematica** die Linux-ABI verwendet. Dies erreichen Sie am einfachsten, indem Sie die Standard-ELF-Kennzeichnung für alle ungekennzeichneten Binärdateien auf Linux festlegen:

```
# sysctl kern.fallback_elf_brand=3
```

Danach wird FreeBSD annehmen, dass alle ungekennzeichneten ELF-Binärdateien die Linux-ABI verwenden und es wäre nun möglich, das Installationsprogramm direkt von der CD-ROM zu starten.

Unter FreeBSD müssen allerdings die Datei `MathInstaller` in ein lokales Verzeichnis Ihrer Festplatte kopieren:

```
# mount /cdrom
# cp /cdrom/Unix/Installers/Linux/MathInstaller /LokalesVerzeichnis/
```

In dieser Datei ersetzen Sie in der ersten Zeile den Wert `/bin/sh` durch `/compat/linux/bin/sh`. Dadurch wird sichergestellt, dass der Installer von der Linux-Version von sh(1) aufgerufen wird. Danach ersetzen Sie durch das im nächsten Abschnitt zu findende Skript oder über einen Texteditor alle Vorkommen von `Linux)` durch `FreeBSD)`. Dadurch ist es dem **Mathematica**-Installer möglich, durch den Einsatz von `uname -s` das Betriebssystem zu bestimmen. FreeBSD wird dabei als Linux-artiges Betriebssystem behandelt. Durch den Aufruf von `MathInstaller` kann **Mathematica** anschließend installiert werden.

11.3.2. Die Mathematica-Programmdateien anpassen

Das von **Mathematica** während der Installation erzeugte Shell-Skript muss angepasst werden, bevor Sie es einsetzen können. Wenn Sie die **Mathematica**-Programmdateien unter `/usr/local/bin` installieren, finden Sie in diesem

Verzeichnis die symbolische Links `math`, `mathematica`, `Mathematica`, sowie `MathKernel`. In jeder dieser Dateien müssen Sie jedes Vorkommen von `Linux)` durch `FreeBSD)` ersetzen (entweder über einen Texteditor oder durch das folgende Shellskript):

```
#!/bin/sh
cd /usr/local/bin
for i in math mathematica Mathematica MathKernel
  do sed 's/Linux)/FreeBSD)/g' $i > $i.tmp
  sed 's/\/bin\/sh/\/compat\/linux\/bin\/sh/g' $i.tmp > $i
  rm $i.tmp
  chmod a+x $i
done
```

11.3.3. Ihr Mathematica-Passwort anfordern

Wenn Sie **Mathematica** das erste Mal starten, werden Sie nach einem Passwort gefragt. Haben Sie noch kein Passwort von Wolfram erhalten, müssen Sie zuerst im Installationsverzeichnis `mathinfo` aufrufen, um Ihre "Rechner-ID" zu bestimmen. Diese Rechner-ID basiert ausschließlich auf der MAC-Adresse Ihrer ersten Netzwerkkarte. Daher ist es nicht möglich, Ihre **Mathematica**-Kopie auf verschiedenen Rechnern zu installieren.

Wenn Sie sich bei Wolfram registrieren (durch E-Mail, Telefon oder Fax), teilen Sie Ihre "Rechner-ID" mit und erhalten dafür ein aus Zahlengruppen bestehendes Passwort.

11.3.4. Das Mathematica-Frontend über ein Netzwerk ausführen

Mathematica verwendet einige spezielle Schriftarten, um Zeichen anzuzeigen, die in den Standardzeichensätzen nicht vorhanden sind (z.B. Integrale, Summen, griechische Buchstaben). Das X-Protokoll verlangt allerdings, dass diese Schriftarten *lokal* installiert sind. Das bedeutet, dass Sie diese Schriftarten von der CD-ROM oder von einem Rechner, auf dem **Mathematica** installiert ist, auf Ihren Rechner kopieren müssen. Diese Schriftarten befinden sich normalerweise in `/cdrom/Unix/Files/SystemFiles/Fonts` (Mathematica-CD) oder in `/usr/local/mathematica/SystemFiles/Fonts` (Festplatte). Die aktuellen Schriftarten befinden sich dabei in den Unterverzeichnissen `Type1` und `X`. Um diese Schriftarten zu verwenden, gibt es mehrere Möglichkeiten, die nun beschrieben werden:

Die erste Möglichkeit besteht darin, die Schriftarten in eins der bereits existierenden Schriftartenverzeichnisse unter `/usr/X11R6/lib/X11/fonts` zu kopieren. Dies bedeutet, dass Sie `fonts.dir` editieren müssen, indem Sie die Schriftnamen hinzufügen und die Anzahl der Schriftarten in der ersten Zeile ändern. Alternativ ist es auch möglich, im Verzeichnis, in das Sie die Schriftarten kopiert haben, das Kommando mkfontdir(1) auszuführen.

Die zweite Möglichkeit, besteht darin, die Verzeichnisse nach `/usr/X11R6/lib/X11/fonts` zu kopieren:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
```

```
# mkfontdir
```

Nun fügen Sie die neuen Schriftartenverzeichnisse in Ihren Pfad ein:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

Wenn Sie den Xorg-Server verwenden, können Sie die Schriftarten-Verzeichnisse automatisch laden lassen, wenn Sie sie in Ihrer `xorg.conf` angeben.

Wenn Sie *noch kein* `/usr/X11R6/lib/X11/fonts/Type1`-Verzeichnis haben, können Sie das `MathType1`-Verzeichnis im vorherigen Beispiel in `Type1` umbenennen.

11.4. Maple™ installieren

Beigetragen von Aaron Kaplan. Mit Unterstützung durch Robert Getschmann.

Maple™ ist ein mit **Mathematica** vergleichbares kommerzielles Mathematikprogramm. Sie können dieses Programm unter http://www.maplesoft.com/ kaufen und sich anschließend registrieren, um eine Lizenz zu erhalten. Um dieses Programm unter FreeBSD zu installieren, gehen Sie wie folgt vor:

1. Führen Sie das `INSTALL`-Shell-Skript der Softwaredistribution aus. Wählen Sie die “RedHat”-Option aus, wenn Sie das Installationsprogramm danach fragt. Ein typisches Installationsverzeichnis wäre z.B. `/usr/local/maple`.
2. Wenn Sie dies noch nicht gemacht haben, besorgen Sie sich nun eine **Maple**-Lizenz von Maple Waterloo Software (http://register.maplesoft.com) und kopieren Sie diese nach `/usr/local/maple/license/license.dat`.
3. Installieren Sie den **FLEXlm**-Lizenz-Manager, indem Sie das `INSTALL_LIC`-Installations-Shellskript ausführen, das mit **Maple** ausgeliefert wird. Geben Sie Ihren primären Rechnernamen für den Lizenz-Server an.
4. Verändern Sie `/usr/local/maple/bin/maple.system.type` wie folgt:

```
   ----- snip ------------------
*** maple.system.type.orig      Sun Jul  8 16:35:33 2001
--- maple.system.type   Sun Jul  8 16:35:51 2001
***************
*** 72,77 ****
--- 72,78 ----
          # the IBM RS/6000 AIX case
          MAPLE_BIN="bin.IBM_RISC_UNIX"
          ;;
+     "FreeBSD"|\
      "Linux")
          # the Linux/x86 case
        # We have two Linux implementations, one for Red Hat and
   ----- snip end of patch -----
```

   Bitte beachten Sie, dass nach `"FreeBSD"|\` kein anderes Zeichen eingefügt werden darf.

Dieser Patch weist **Maple** an, FreeBSD als “eine Art von Linux-System” zu erkennen. Das Shell-Skript `bin/maple` ruft das Shell-Skript `bin/maple.system.type` auf, welches wiederum `uname -a` verwendet, um den Namen des Betriebssystems herauszufinden. Abhängig vom Betriebssystem weiß das System nun, welche Binärdateien verwendet werden sollen.

5. Starten Sie den Lizenz-Server.

   Das folgende, als `/usr/local/etc/rc.d/lmgrd.sh` installierte Shell-Skript ist ein komfortabler Weg, um `lmgrd` zu starten:

```
   ----- snip ------------

#! /bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
LOG=/var/log/lmgrd.log

case "$1" in
start)
       lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
       echo -n " lmgrd"
       ;;
stop)
       lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
       ;;
*)
       echo "Usage: ‘basename $0‘ {start|stop}" 1>&2
       exit 64
       ;;
esac

exit 0
   ----- snip ------------
```

6. Versuchen Sie, **Maple** zu starten:

```
% cd /usr/local/maple/bin
% ./xmaple
```

   Nun sollte das Programm laufen und alles funktionieren. Falls ja, vergessen Sie nicht, an Maplesoft zu schreiben und sie wissen zu lassen, dass Sie gerne eine native FreeBSD-Version hätten.

11.4.1. Häufige Fehlerquellen

- Der **FLEXlm**-Lizenzmanager kann schwierig zu bedienen sein. Zusätzliche Dokumentation zu diesem Thema finden Sie unter http://www.globetrotter.com/.
- Es ist bekannt, dass `lmgrd` sehr pingelig ist, wenn es um die Lizenzdatei geht. Gibt es Probleme, führt dies zu einem Speicherauszug (*core dump*). Ein korrekte Lizenzdatei sollte ähnlich der folgenden aussehen:

```
# ====================================================
# License File for UNIX Installations ("Pointer File")
```

```
# ================================================
SERVER chillig ANY
#USE_SERVER
VENDOR maplelmg

FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXX \
         PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." \
         ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" \
         SN=XXXXXXXXX
```

Anmerkung: Seriennummer und Schlüssel wurden durch mehrere `X` unkenntlich gemacht. `chillig` ist ein Rechnername.

Veränderungen an der Lizenzdatei sind möglich, solange Sie die `FEATURE`-Zeile nicht verändern (diese ist durch den Lizenzschlüssel geschützt).

11.5. MATLAB® installieren

Beigesteuert von Dan Pelleg.

Im Folgenden wird die Installation der Linux-Anwendung **MATLAB®** Version 6.5 auf FreeBSD beschrieben. Mit Ausnahme der **Java Virtual Machine™** (siehe Abschnitt 11.5.3) läuft die Anwendung auch ganz gut.

Die Linux-Version von **MATLAB** können Sie direkt bei The MathWorks (http://www.mathworks.com) bestellen. Vergewissern Sie sich, dass Sie die Lizenz-Datei oder eine Anleitung zum Erstellen der Lizenz-Datei erhalten haben. Wenn Sie mit MathWorks in Kontakt stehen, weisen Sie bitte auf die fehlende FreeBSD-Version der Software hin.

11.5.1. Das MATLAB-Installationsskript

Um **MATLAB** zu installieren, gehen Sie wie folgt vor:

1. Hängen Sie die Installations-CD ein und wechseln Sie zu `root`, wie im Installations-Skript gefordert. Starten Sie die Installation mit dem folgenden Kommando:

   ```
   # /compat/linux/bin/sh /cdrom/install
   ```

   Tipp: Die Installation erfordert eine graphische Benutzeroberfläche. Wenn Sie die Fehlermeldung erhalten, dass das Display nicht geöffnet werden konnte, führen Sie das folgende Kommando aus:

   ```
   # setenv HOME ~USER
   ```

   Für *`USER`* setzen Sie den Benutzer ein, von dem aus Sie `root` geworden sind.

2. Beantworten Sie die Frage nach dem **MATLAB**-Root-Verzeichnis mit: **`/compat/linux/usr/local/matlab`**.

   Tipp: Den langen Pfad werden Sie noch öfter brauchen. Die Tipparbeit können Sie sich mit dem folgenden Befehl erleichtern:

```
# set MATLAB=/compat/linux/usr/local/matlab
```

3. Editieren Sie die Lizenz-Datei entsprechend der Anweisung, die Sie beim Erwerb der Lizenz erhalten haben.

   Tipp: Sie können die Datei schon vorher mit Ihrem Lieblingseditor bearbeiten. Kopieren Sie die Lizenz-Datei nach `$MATLAB/license.dat` bevor das Installationsprogramm Sie auffordert, die Datei zu editieren.

4. Schließen Sie die Installation ab.

Die **MATLAB**-Installation ist jetzt abgeschlossen. Die folgenden Schritte passen **MATLAB** an FreeBSD an.

11.5.2. Den Lizenzmanager starten

1. Erstellen Sie symbolische Links zu den Startskripten des Lizenzmanagers:

   ```
   # ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
   # ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
   ```

2. Erstellen Sie das Startskript `/usr/local/etc/rc.d/flexlm.sh`. Das folgende Beispiel ist eine geänderte Version des mitgelieferten Skripts `$MATLAB/etc/rc.lm.glnx86`. Angepasst wurden die Pfade zu den Dateien und der Start des Lizenzmanagers unter der Linux-Emulation.

   ```
   #!/bin/sh
   case "$1" in
     start)
           if [ -f /usr/local/etc/lmboot_TMW ]; then
                 /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u username && echo 'MATLAB_lmgrd'
           fi
           ;;
     stop)
          if [ -f /usr/local/etc/lmdown_TMW ]; then
               /compat/linux/bin/sh /usr/local/etc/lmdown_TMW  > /dev/null 2>&1
          fi
           ;;
     *)
          echo "Usage: $0 {start|stop}"
          exit 1
          ;;
   esac

   exit 0
   ```

   Wichtig: Machen Sie Datei ausführbar:

   ```
   # chmod +x /usr/local/etc/rc.d/flexlm.sh
   ```

   Ersetzen Sie im Skript *username* durch einen existierenden Benutzer Ihres Systems (bitte keinesfalls `root`).

3. Starten Sie den Lizenzmanager:

   ```
   # /usr/local/etc/rc.d/flexlm.sh start
   ```

11.5.3. Einrichten der Java-Laufzeitumgebung

Erstellen Sie einen symbolischen Link auf eine unter FreeBSD laufende Java-Laufzeitumgebung (JRE):

```
# cd $MATLAB/sys/java/jre/glnx86/
# unlink jre; ln -s ./jre1.1.8 ./jre
```

11.5.4. Ein MATLAB-Startskript erstellen

1. Kopieren Sie das folgende Skript nach `/usr/local/bin/matlab`:

   ```
   #!/bin/sh
   /compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
   ```

2. Machen Sie das Skript ausführbar:

   ```
   # chmod +x /usr/local/bin/matlab
   ```

   Tipp: Abhängig von der Version des Ports `emulators/linux_base` kann das Skript auf Fehler laufen. Die Fehler können Sie vermeiden, indem Sie die Datei `/compat/linux/usr/local/matlab/bin/matlab` editieren. Ändern Sie die nachstehende Zeile

   ```
   if [ ‘expr "$lscmd" : ’.*->.*’‘ -ne 0 ]; then
   ```

   (mit Version 13.0.1 in der Zeile 410) in die folgende um:

   ```
   if test -L $newbase; then
   ```

11.5.5. Stopp-Skript für MATLAB erstellen

Das nachstehende Skript beendet MATLAB ordnungsgemäß.

1. Erstellen Sie die Datei `$MATLAB/toolbox/local/finish.m` mit dem nachstehenden Inhalt:

   ```
   ! $MATLAB/bin/finish.sh
   ```

   Anmerkung: Übernehmen Sie die Zeichenkette `$MATLAB` unverändert.

   Tipp: Im selben Verzeichnis befinden sich die Dateien `finishsav.m` und `finishdlg.m`. Die Dateien sichern die Einstellungen der Arbeitsfläche bevor MATLAB beendet wird. Wenn Sie eine der beiden Dateien benutzen, fügen Sie die obige Zeile unmittelbar nach dem `save`-Kommando ein.

2. Erstellen Sie die Datei `$MATLAB/bin/finish.sh` mit nachstehendem Inhalt:

   ```
   #!/compat/linux/bin/sh
   (sleep 5; killall -1 matlab_helper) &
   ```

```
exit 0
```

3. Machen Sie die Datei ausführbar:

```
# chmod +x $MATLAB/bin/finish.sh
```

11.5.6. MATLAB benutzen

Jetzt können Sie **MATLAB** mit dem `matlab` starten.

11.6. Oracle® installieren

Beigetragen von Marcel Moolenaar.

11.6.1. Übersicht

Dieses Dokument beschreibt die Installation von **Oracle 8.0.5** und **Oracle 8.0.5.1 Enterprise Edition** für Linux auf einem FreeBSD-Rechner.

11.6.2. Installation der Linux-Umgebung

Stellen Sie sicher, dass Sie sowohl `emulators/linux_base` und `devel/linux_devtools` aus der Ports-Sammlung installiert haben. Wenn Sie mit diesen Ports Schwierigkeiten haben, müssen Sie vielleicht ältere Versionen der Linux-Umgebung aus der Ports-Sammlung installieren.

Wenn Sie den Intelligent-Agent verwenden wollen, müssen Sie zusätzlich das RedHat Tcl-Paket installieren: `tcl-8.0.3-20.i386.rpm`. Zur Installation von RPM-Paketen wir der Port `archivers/rpm` benötigt. Ist der Port installiert, lassen sich RPM-Pakete anschließend mit dem nachstehenden Befehl installieren:

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm package
```

Die Installation der RPM-Pakete sollte ohne Fehlermeldung ablaufen.

11.6.3. Die Oracle-Umgebung erzeugen

Bevor Sie **Oracle** installieren können, müssen Sie eine entsprechende Umgebung erzeugen. Dieses Dokument beschreibt nur, was Sie *im Speziellen* tun müssen, um die Linux-Version von **Oracle** unter FreeBSD zu installieren, nicht aber, was bereits in der Installationsanleitung von **Oracle** beschrieben wird.

11.6.3.1. Kernel-Tuning

Wie in der Installationsanleitung von **Oracle** beschrieben, müssen Sie die maximale Shared-Memory Größe festlegen. Verwenden Sie `SHMMAX` nicht unter FreeBSD. `SHMMAX` wird lediglich aus `SHMMAXPGS` und `PGSIZE` berechnet. Definieren Sie stattdessen `SHMMAXPGS`. Alle anderen Optionen können wie in der Anleitung beschrieben verwendet werden. Zum Beispiel:

```
options SHMMAXPGS=10000
```

```
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Passen Sie diese Optionen entsprechend dem von Ihnen gewünschten Einsatzzweck von **Oracle** an.

Stellen Sie außerdem sicher, dass Sie folgende Optionen in Ihren Kernel kompilieren:

```
options SYSVSHM #SysV shared memory
options SYSVSEM #SysV semaphores
options SYSVMSG #SysV interprocess communication
```

11.6.3.2. Oracle-Benutzer anlegen

Legen Sie den Account `oracle` an. Der Account unterschiedet sich von normalen Accounts dadurch, dass er eine Linux-Shell zugeordnet bekommen muss. Fügen Sie `/compat/linux/bin/bash` in die Datei `/etc/shells` ein und setzen Sie die Shell für den `oracle`-Account auf `/compat/linux/bin/bash`.

11.6.3.3. Umgebung

Neben den normalen **Oracle**-Variablen, wie z.B. `ORACLE_HOME` und `ORACLE_SID` müssen Sie die folgenden Variablen setzen:

Variable	Wert
`LD_LIBRARY_PATH`	`$ORACLE_HOME/lib`
`CLASSPATH`	`$ORACLE_HOME/jdbc/lib/classes111.zip`
`PATH`	`/compat/linux/bin /compat/linux/sbin /compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin /usr/bin /usr/sbin /usr/local/bin $ORACLE_HOME/bin`

Es ist empfehlenswert, alle Variablen in der Datei `.profile` zu setzen. Ein komplettes Beispiel sieht folgendermaßen aus:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

11.6.4. Oracle installieren

Auf Grund einer kleinen Unregelmäßigkeit im Linux-Emulator müssen Sie das Verzeichnis `.oracle` unter `/var/tmp` erzeugen, bevor Sie das Installationsprogramm starten. Das Verzeichnis muss dem Account `oracle` gehören. Sie sollten **Oracle** nun ohne Probleme installieren können. Treten dennoch Probleme auf, überprüfen Sie zuerst Ihre **Oracle**-Distribution und Ihre Konfiguration. Nachdem Sie **Oracle** erfolgreich installiert haben, installieren Sie die Patches wie in den zwei folgenden Abschnitten beschrieben:

Ein häufiges Problem ist, dass der TCP Protokoll-Adapter nicht korrekt installiert wird. Daraus folgt, dass Sie keine TCP-Listener starten können. Dieses Problem kann durch folgende Schritte behoben werden:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

Vergessen Sie nicht, `root.sh` nochmals auszuführen!

11.6.4.1. root.sh patchen

Während der **Oracle**-Installation werden einige Aktionen, die als `root` ausgeführt werden müssen, in ein Shell-Skript mit dem Namen `root.sh` gespeichert. Dieses Skript befindet sich im Verzeichnis `orainst`. Verwenden Sie folgenden Patch für `root.sh`, damit es das richtige `chown` Kommando verwendet, oder lassen Sie das Skript alternativ unter einer Linux-Shell ablaufen:

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
***************
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

Wenn Sie **Oracle** nicht von CD-ROM installieren, können Sie Quelldatei für `root.sh` verändern. Sie heißt `rthd.sh` und befindet sich im `orainst`-Verzeichnis des Quellcodebaums.

11.6.4.2. genclntsh patchen

Das Skript `genclntsh` wird verwendet, um eine Shared-Library für Clients zu erzeugen. Diese wird bei der Erzeugung der Demos verwendet. Verwenden Sie folgenden Patch, um die Definition von `PATH` auszukommentieren:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
***************
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

11.6.5. Oracle starten

Wenn Sie den Anweisungen gefolgt sind, sollten Sie nun in der Lage sein, **Oracle** zu starten, genau so, wie Sie dies auch unter Linux tun würden.

11.7. Weiterführende Themen

Wenn Sie sich fragen, wie die Linux-Binärkompatibilität unter FreeBSD realisiert wurde, sollten Sie diesen Abschnitt lesen. Der Großteil der folgenden Informationen stammt aus einer E-Mail, die von Terry Lambert (<`tlambert@primenet.com`>) an die FreeBSD-Chat-Mailingliste (<`freebsd-chat@FreeBSD.org`>) geschrieben wurde (Message ID: <`199906020108.SAA07001@usr09.primenet.com`>).

11.7.1. Wie funktioniert es?

FreeBSD verfügt über eine "execution class loader" genannte Abstraktion. Dabei handelt es sich um einen Eingriff in den execve(2) Systemaufruf.

FreeBSD verfügt über eine Liste von Ladern, anstelle eines einzigen, auf `#!` zurückgreifenden Laders, um Shell-Interpreter oder Shell-Skripte auszuführen.

Historisch gesehen untersuchte der einzige, auf UNIX-Plattformen vorhandene Lader die "magische Zahl" (in der Regel die ersten 4 oder 8 Bytes der Datei), um festzustellen, ob der Binärtyp dem System bekannt war. War dies der Fall, wurde der Binärlader aufgerufen.

Wenn es sich nicht um den zum System gehörigen Binärtyp handelte, gab execve(2) einen Fehler zurück, und die Shell versuchte stattdessen, die Datei als Shell-Befehl auszuführen.

Dabei wurde als Standardeinstellung “was auch immer die aktuelle Shell ist” festgelegt.

Später wurde ein Hack in sh(1) eingefügt, der die zwei ersten Zeichen untersuchte. Wenn diese `:\n` entsprachen, wurde stattdessen die csh(1)-Shell aufgerufen (wir glauben, dass dies zuerst von SCO umgesetzt wurde).

FreeBSD versucht heute eine Liste von Ladern, unter denen sich ein allgemeiner Lader für Interpreter befindet. Der auszuführende Interpreter wird im ersten, durch Leerzeichen getrennten Feld, der `#!`-Zeile angegeben. Lässt sich der Interpreter nicht ermitteln, wird auf `/bin/sh` zurückgegriffen.

Für die Linux ABI-Unterstützung erkennt FreeBSD die magische Zahl als ELF-Binärdatei (Zu diesem Zeitpunkt wird nicht zwischen FreeBSD, Solaris, Linux oder anderen Systemen unterschieden, die über ELF-Binärdateien verfügen.).

Der ELF-Lader sucht nach einer speziellen *Kennzeichnung*, die aus einem Kommentarabschnitt in der ELF-Datei besteht, und die in SVR4/Solaris ELF Binärdateien nicht vorhanden ist.

Damit Linux-Binärdateien (unter FreeBSD) funktionieren, müssen sie als `Linux` *gekennzeichnet* werden, und zwar durch brandelf(1):

```
# brandelf -t Linux file
```

Nachdem dies geschehen ist, erkennt der ELF-Lader die `Linux`-Kennzeichnung der Datei.

Wenn der ELF-Lader die `Linux`-Kennzeichnung sieht, wird ein Zeiger in der `proc`-Struktur ersetzt. Alle Systemaufrufe werden durch diesen Zeiger indiziert (in einem traditionellen UNIX System wäre das ein `sysent[]`-Strukturfeld, das die Systemaufrufe enthält). Der Prozess wird weiterhin speziell gekennzeichnet, so dass der Trap-vector im Signal-trampoline-code eine spezielle Behandlung erfährt und das Linux-Kernelmodul verschiedene kleinere Korrekturen vornehmen kann.

Der Linux-Systemaufrufvektor enthält neben anderen Dingen eine Liste der `sysent[]`-Einträge, deren Adressen sich im Kernelmodul befinden.

Wenn ein Linux-Programm einen Systemaufruf ausführt, dereferenziert die Trap-Behandlungsroutine den Zeiger auf die Eintrittspunkte für die Systemaufrufe und erhält damit die Linux-Eintrittspunkte und nicht die FreeBSD-Eintrittspunkte.

Zusätzlich *verändert* der Linuxmodus die Systempfade dynamisch; genauso, wie dies die Option `union` beim Einbinden von Dateisystemen macht (Achtung: *nicht* das Dateisystem unionfs!). Zuerst wird die Datei im Verzeichnis `/compat/linux/`*`Originalpfad`* gesucht, *danach*, wenn sie dort nicht gefunden wurde, wird sie im FreeBSD-Verzeichnis *`/Originalpfad`* gesucht. Dadurch wird sichergestellt, dass Binärdateien, die zur Ausführung andere Binärdateien benötigen, ausgeführt werden können (so dass alle Linux-Werkzeuge unter der ABI laufen). Dies bedeutet auch, dass Linux-Binärdateien FreeBSD-Binärdateien laden und ausführen können, wenn keine passenden Linux-Binärdateien vorhanden sind. Ein in `/compat/linux` plaziertes uname(1) kann damit Linux-Programmen vorgaukeln, dass sie auf einem Linux-System laufen.

Im Endeffekt gibt es einen Linux-Kernel innerhalb des FreeBSD-Kernels. Die Sprungtabellen für Linux- beziehungsweise FreeBSD-Systemaufrufe verweisen allerdings auf dieselben Funktionen, die Kerneldienste wie Dateisystemoperationen, Operationen für den virtuellen Speicher, Signalübermittlung und System V IPC bereitstellen, Der einzige Unterschied ist, dass Binärdateien unter FreeBSD FreeBSD-*glue*-Funktionen verwenden. Linux-Binärdateien hingegen verwenden die Linux-*glue*-Funktionen. Die meisten älteren Betriebssysteme hatten ihre eigenen *glue*-Funktionen: Funktionsadressen in einem globalen, statischen `sysent[]` Strukturfeld an Stelle von Funktionsadressen, die durch einen dynamisch initialisierten Zeiger aus der `proc` Struktur, die den Aufruf gemacht hatte, dereferenziert wurden.

Welche ist die echte FreeBSD-ABI? Das spielt keine Rolle. Grundsätzlich ist der einzige Unterschied (zurzeit ist das so; dies könnte sich in zukünftigen Versionen leicht ändern und wird sich wahrscheinlich auch ändern), dass die FreeBSD-*glue*-Funktionen statisch in den Kernel gelinkt sind, und dass die Linux-*glue*-Funktionen statisch gelinkt oder über ein Modul eingebunden werden können.

Ja, aber ist das wirkliche eine Emulation? Nein. Es ist eine Implementierung eines ABIs, keine Emulation. Es ist kein Emulator (oder Simulator, um der nächsten Frage zuvorzukommen) beteiligt.

Warum wird es manchmal “Linux-Emulation” genannt? Um es schwerer zu machen, FreeBSD zu verkaufen. Wirklich, das kommt daher, weil dies zu einer Zeit implemtiert wurde, in der es kein anderes Wort (als Emulation) gab, das beschrieb, was vor sich ging. Wenn der Kernel nicht entsprechend konfiguriert wurde oder das Modul geladen wurde, war es falsch zu behaupten, FreeBSD würde Linux-Binärprogramme ausführen. Man benötigte ein Wort, das beschrieb, was da geladen wurde – daher “Der Linux-Emulator”.

III. Systemadministration

Die restlichen Kapitel behandeln alle Aspekte der FreeBSD Systemadministration. Am Anfang jedes Kapitels finden Sie eine Zusammenfassung, die beschreibt, was Sie nach dem Durcharbeiten des Kapitels gelernt haben. Weiterhin werden die Voraussetzungen beschrieben, die für das Durcharbeiten des Kapitels erforderlich sind.

Diese Kapitel sollten Sie lesen, wenn Sie die Informationen darin benötigen. Sie brauchen Sie nicht in einer bestimmten Reihenfolge zu lesen, noch müssen Sie die Kapitel lesen, bevor Sie anfangen, FreeBSD zu benutzen.

Kapitel 12. Konfiguration und Tuning

Geschrieben von Chern Lee. Nach einem Tutorium von Mike Smith. Basiert ebenfalls auf tuning(7) von Matt Dillon. Übersetzt von Martin Heinen.

12.1. Übersicht

Ein korrekt konfiguriertes System kann die Arbeit, die bei der zukünftigen Pflege und bei Migrationen des Systems entsteht, erheblich reduzieren. Dieses Kapitel beschreibt die Konfiguration von FreeBSD sowie Maßnahmen zur Leistungssteigerung von FreeBSD-Systemen.

Nachdem Sie dieses Kapitel durchgearbeitet haben, werden Sie Folgendes wissen:

- Wie Sie effizient Dateisysteme und Swap-Partitionen auf Ihrer Festplatte einrichten.
- Die Grundlagen der Konfiguration mit `rc.conf` und des Systems zum Starten von Anwendungen in `/usr/local/etc/rc.d`.
- Wie Sie Netzwerkkarten konfigurieren und testen.
- Wie Sie virtuelle Hosts und Netzwerkgeräte konfigurieren.
- Wie Sie die verschiedenen Konfigurationsdateien in `/etc` benutzen.
- Wie Sie mit `sysctl`-Variablen FreeBSD einstellen können.
- Wie Sie die Platten-Performance einstellen und Kernel-Parameter modifizieren können.

Bevor Sie dieses Kapitel lesen, sollten Sie

- die Grundlagen von UNIX und FreeBSD (Kapitel 4) verstehen.
- Damit vertraut sein, wie Sie einen Kernel konfigurieren und kompilieren (Kapitel 9).

12.2. Vorbereitende Konfiguration

12.2.1. Layout von Partitionen

12.2.1.1. Partitionen

Wenn Sie Dateisysteme mit bsdlabel(8) oder sysinstall(8) anlegen, sollten Sie beachten, dass Festplatten auf Daten in den äußeren Spuren schneller zugreifen können als auf Daten in den inneren Spuren. Daher sollten die kleineren oft benutzten Dateisysteme, wie das Root-Dateisystem oder die Swap-Partition, an den äußeren Rand der Platte gelegt werden. Die größeren Partitionen wie `/usr` sollten in die inneren Bereiche gelegt werden. Es empfiehlt sich, die Partitionen in einer ähnlichen Reihenfolge wie Root-Partition, Swap, `/var` und `/usr` anzulegen.

Die Größe der `/var`-Partition ist abhängig vom Zweck der Maschine. Das `/var`-Dateisystem enthält hauptsächlich Postfächer, den Spoolbereich zum Drucken und Logdateien. Abhängig von der Anzahl der Systembenutzer und der Aufbewahrungszeit für Logdateien, können gerade die Postfächer und Logdateien zu ungeahnten Größen wachsen. Die meisten Benutzer werden selten mehr als etwa ein Gigabyte in `/var` benötigen.

> **Anmerkung:** Ein paar Mal wird es vorkommen, dass viel Festplattenspeicher in `/var/tmp` gebraucht wird. Wenn neue Software mit pkg_add(1) installiert wird, extrahieren die Paketwerkzeuge eine vorübergehende Kopie der Pakete unter `/var/tmp`. Die Installation grosser Softwarepakete wie **Firefox**, **Openoffice** oder **LibreOffice** kann sich wegen zu wenig Speicherplatz in `/var/tmp` als trickreich herausstellen.

Die `/usr`-Partition enthält viele der Hauptbestandteile des Systems, dazu gehöhren die ports(7)-Sammlung (empfohlen) und die Quellen (optional). Sowohl die Ports als auch die Quellen des Basissystems sind zum Zeitpunkt der Installation optional, trotzdem sollten Sie mindestens zwei Gigabyte für diese Partition vorsehen.

Wenn Sie die Größe der Partitionen festlegen, beachten Sie bitte das Wachstum Ihres Systems. Wenn Sie den Platz auf einer Partition vollständig aufgebraucht haben, eine andere Partition aber kaum benutzen, kann die Handhabung des Systems schwierig werden.

> **Anmerkung:** Die automatische Partitionierung von sysinstall(8) mit `Auto-defaults` legt manchmal zu kleine `/` und `/var`-Partition an. Partitionieren Sie weise und großzügig.

12.2.1.2. Swap Partition

Als Daumenregel sollten Sie doppelt soviel Speicher für die Swap-Partition vorsehen, als Sie Hauptspeicher haben. Verfügt die Maschine beispielsweise über 128 Megabyte Hauptspeicher, sollten Sie 256 Megabyte für den Swap-Bereich vorsehen. Systeme mit weniger Speicher werden wahrscheinlich mit viel mehr Swap mehr leisten. Es wird nicht empfohlen, weniger als 256 Megabyte Swap einzurichten. Außerdem sollten Sie künftige Speichererweiterungen beachten, wenn Sie die Swap-Partition einrichten. Die VM-Paging-Algorithmen im Kernel sind so eingestellt, dass Sie am besten laufen, wenn die Swap-Partition mindestens doppelt so groß wie der Hauptspeicher ist. Zu wenig Swap kann zu einer Leistungsverminderung im *VM page scanning* Code führen, sowie Probleme verursachen, wenn Sie später mehr Speicher in Ihre Maschine bauen.

Auf größeren Systemen mit mehreren SCSI-Laufwerken (oder mehreren IDE-Laufwerken an unterschiedlichen Controllern) empfehlen wir Ihnen, Swap-Bereiche auf bis zu vier Laufwerken einzurichten. Diese Swap-Partitionen sollten ungefähr dieselbe Größe haben. Der Kernel kann zwar mit beliebigen Größen umgehen, aber die internen Datenstrukturen skalieren bis zur vierfachen Größe der größten Partition. Ungefähr gleich große Swap-Partitionen erlauben es dem Kernel, den Swap-Bereich optimal über die Laufwerke zu verteilen. Große Swap-Bereiche, auch wenn sie nicht oft gebraucht werden, sind nützlich, da sich ein speicherfressendes Programm unter Umständen auch ohne einen Neustart des Systems beenden lässt.

12.2.1.3. Warum partitionieren?

Gegen eine einzelne Partition sprechen mehrere Gründe. Jede Partition hat im Betrieb unterschiedliche Eigenschaften und die Trennung der Partitionen erlaubt es, die Dateisysteme an diese Eigenschaften anzupassen. Die Root- und `/usr`-Partitionen weisen meist nur lesende Zugriffe auf, während `/var` und `/var/tmp` hauptsächlich beschrieben werden.

Indem Sie ein System richtig partitionieren, verhindern Sie, dass eine Fragmentierung in den häufig beschriebenen Partitionen auf die meist nur gelesenen Partitionen übergreift. Wenn Sie die häufig beschriebenen Partitionen an den Rand der Platte, legen, dann wird die I/O-Leistung diesen Partitionen steigen. Die I/O-Leistung ist natürlich auch für große Partitionen wichtig, doch erzielen Sie eine größere Leistungssteigerung, wenn Sie `/var` an den Rand der

Platte legen. Schließlich sollten Sie noch die Stabilität des Systems beachten. Eine kleine Root-Partition, auf die meist nur lesend zugegriffen wird, überlebt einen schlimmen Absturz wahrscheinlich eher als eine große Partition.

12.3. Basiskonfiguration

Informationen zur Systemkonfiguration sind hauptsächlich in `/etc/rc.conf`, die meist beim Start des Systems verwendet wird, abgelegt. Der Name der Datei zeigt ihren Zweck an: Sie enthält die Konfigurationen für die `rc*` Dateien.

In `rc.conf` werden die Vorgabewerte aus `/etc/defaults/rc.conf` überschrieben. Die Vorgabedatei sollte nicht nach `/etc` kopiert werden, da sie die Vorgabewerte und keine Beispiele enthält. Jede systemspezifische Änderung wird in `rc.conf` vorgenommen.

Um den administrativen Aufwand gering zu halten, existieren in geclusterten Anwendungen mehrere Strategien, globale Konfigurationen von systemspezifischen Konfigurationen zu trennen. Der empfohlene Weg hält die globale Konfiguration in einer separaten Datei z.B. `/etc/rc.conf.local`. Zum Beispiel so:

- `/etc/rc.conf`:

  ```
  sshd_enable="YES"
  keyrate="fast"
  defaultrouter="10.1.1.254"
  ```

- `/etc/rc.conf.local`:

  ```
  hostname="node1.example.org"
  ifconfig_fxp0="inet 10.1.1.1/8"
  ```

Die `rc.conf` Datei kann dann auf jedes System mit `rsync` oder einem ähnlichen Programm verteilt werden, während die `rc.conf.local` Datei dabei systemspezifisch bleibt.

Bei einem Upgrade des Systems mit sysinstall(8) oder `make world` wird `rc.conf` nicht überschrieben, so dass die Systemkonfiguration erhalten bleibt.

> **Tipp:** Die Konfigurationsdatei `/etc/rc.conf` wird von sh(1) gelesen. Dies erlaubt es dem Systemadministrator, eine bestimmte Menge an Logik dieser Datei hinzuzufügen, was dabei helfen kann, komplexe Konfigurationsszenarien zu erstellen. Lesen Sie dazu rc.conf(5), um weitere Informationen zu diesem Thema zu erhalten.

12.4. Konfiguration von Anwendungen

Installierte Anwendungen haben typischerweise eigene Konfigurationsdateien, die eine eigene Syntax verwenden. Damit diese Dateien leicht von der Paketverwaltung gefunden und verwaltet werden können, ist es wichtig, sie vom Basissystem zu trennen.

Für gewöhnlich werden diese Dateien in `/usr/local/etc` installiert. Besitzt eine Anwendung viele Konfigurationsdateien, werden diese in einem separaten Unterverzeichnis abgelegt.

Wenn ein Port oder ein Paket installiert wird, werden normalerweise auch Beispiele für die Konfigurationsdateien installiert. Diese erkennt man gewöhnlich an dem Suffix `.default`. Wenn keine Konfigurationsdateien für eine Anwendung existieren, werden sie durch Kopieren der `.default` Dateien erstellt.

Als Beispiel sei `/usr/local/etc/apache` gezeigt:

```
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf
-rw-r--r--  1 root  wheel   2184 May 20  1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May 20  1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic
-rw-r--r--  1 root  wheel  12205 May 20  1998 magic.default
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types
-rw-r--r--  1 root  wheel   2700 May 20  1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May 20  1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May 20  1998 srm.conf.default
```

Anhand der Dateigröße erkennen Sie, dass sich nur `srm.conf` geändert hat. Eine spätere Aktualisierung des **Apache**-Ports überschreibt diese Datei nicht.

12.5. Start von Diensten

Beigetragen von Tom Rhodes.

Viele Benutzer installieren Software Dritter auf FreeBSD mithilfe der Ports-Sammlung. Häufig soll die Software bei einem Systemstart mitgestartet werden. Beispielsweise sollen die Dienste `mail/postfix` oder `www/apache13` nach einem Systemstart laufen. Dieser Abschnitt stellt die Startprozeduren für Software Dritter vor.

Unter FreeBSD werden die meisten der im System enthaltenen Dienste wie cron(8) mithilfe von Systemskripten gestartet. Diese Skripten sind abhängig von der FreeBSD- oder Hersteller-Version. Allerdings kann ein Dienst mit einfachen Skripten gestartet werden.

12.5.1. Dienste über das `rc.d`-System starten

Mit `rc.d` lässt sich der Start von Anwendungen besser steuern als mit den vorher besprochenen Startskripten. Mit den im Abschnitt rc.d besprochenen Schlüsselwörtern können Anwendungen in einer bestimmten Reihenfolge (zum Beispiel nach DNS) gestartet werden und Optionen können in `rc.conf` statt fest im Startskript der Anwendung festgelegt werden. Ein einfaches Startskript sieht wie folgt aus:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=utility
rcvar=utility_enable

command="/usr/local/sbin/utility"
```

```
load_rc_config $name

#
# DO NOT CHANGE THESE DEFAULT VALUES HERE
# SET THEM IN THE /etc/rc.conf FILE
#
utility_enable=${utility_enable-"NO"}
pidfile=${utility_pidfile-"/var/run/utility.pid"}

run_rc_command "$1"
```

Dieses Skript stellt sicher, dass **utility** nach den `DAEMON`-Pseudodiensten gestartet wird. Es stellt auch eine Methode bereit, die Prozess-ID (PID) der Anwendung in einer Datei zu speichern.

In `/etc/rc.conf` könnte für diese Anwendung die folgende Zeile stehen:

```
utility_enable="YES"
```

Die Methode erleichtert den Umgang mit Kommandozeilenargumenten, bindet Funktionen aus `/etc/rc.subr` ein, ist kompatibel zum Werkzeug rcorder(8) und lässt sich über `rc.conf` leichter konfigurieren.

12.5.2. Andere Arten, um Dienste zu starten

Dienste wie POP3 oder IMAP können über inetd(8) gestartet werden. Nach der Installation der Anwendung aus der Ports-Sammlung muss eine Konfigurationszeile in der Datei `/etc/inetd.conf` hinzugefügt oder in der aktuellen Konfiguration durch Entfernen der Kommentare aktiviert werden. Der Abschnitt Abschnitt 30.2 beschreibt den **inetd** und dessen Konfiguration.

Systemdienste können auch mit cron(8) gestartet werden. Dieser Ansatz hat einige Vorteile; nicht zuletzt, weil cron(8) die Prozesse unter dem Eigentümer der `crontab` startet, ist es möglich, dass Dienste von nicht-`root` Benutzern gestartet und gepflegt werden können.

Dies nutzt eine Eigenschaft von cron(8): Für die Zeitangabe kann `@reboot` eingesetzt werden. Damit wird das Kommando gestartet, wenn cron(8) kurz nach dem Systemboot gestartet wird.

12.6. Programme mit `cron` starten

Beigetragen von Tom Rhodes.

Ein sehr nützliches Werkzeug von FreeBSD ist cron(8). `cron` läuft im Hintergrund und überprüft fortlaufend die Datei `/etc/crontab`. Beim Start sucht `cron` neue `crontab`-Dateien im Verzeichnis `/var/cron/tabs`. In den `crontab`-Dateien wird festgelegt, welche Programme zu welchem Zeitpunkt laufen sollen.

Das Werkzeug `cron` verwendet zwei verschiedene Konfigurationsdateien: Die System-`crontab` und die Benutzer-`crontab`. Der einzige Unterschied zwischen beiden Formaten ist das sechste Feld. In der System-`crontab` gibt das sechste Feld das Konto an, unter dem ein Kommando läuft. Aus der System-`crontab` können daher Kommandos unter beliebigen Konten gestartet werden. In der Benutzer-`crontab` gibt das sechste Feld das auszuführende Kommando an. Alle Kommandos laufen unter dem Konto, unter dem die `crontab` erstellt wurde (ein wichtiges Sicherheitsmerkmal).

Anmerkung: Benutzer können mit Benutzer-`crontabs` ohne `root`-Rechte Befehle terminieren. Die Kommandos in Benutzer-`crontabs` laufen unter dem Benutzer, der die `crontab` erstellt hat.

Der Benutzer `root` kann, wie jeder andere Benutzer, eine Benutzer-`crontab` besitzen. Die Benutzer-`crontab` von `root` ist nicht mit der Datei `/etc/crontab`, der System-`crontab`, zu verwechseln. Normalerweise besitzt `root`, wegen der Existenz der System-`crontab`, keine eigene Benutzer-`crontab`.

Der folgende Auszug aus der System-`crontab` `/etc/crontab` zeigt den Aufbau einer `crontab`-Datei:

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
# ❶
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ❷
HOME=/var/log
#
#
#minute        hour    mday    month   wday    who     command ❸
#
#
*/5    *       *       *       *       root    /usr/libexec/atrun ❹
```

❶ Das Zeichen # leitet, wie in den meisten Konfigurationsdateien, einen Kommentar ein. Benutzen Sie Kommentare, um die Funktion eines Eintrags zu erläutern. Kommentare müssen in einer extra Zeile stehen. Sie können nicht in derselben Zeile wie ein Kommando stehen, da sie sonst Teil des Kommandos wären. Leerzeilen in dieser Datei werden ignoriert.

❷ Umgebungsvariablen werden mit dem Gleichheits-Zeichen (=) festgelegt. Im Beispiel werden die Variablen `SHELL`, `PATH` und `HOME` definiert. Wenn die Variable `SHELL` nicht definiert wird, benutzt `cron` die Shell `sh`. Wird die Variable `PATH` nicht gesetzt, müssen alle Pfadangaben absolut sein, da es keinen Vorgabewert für `PATH` gibt. Der Vorgabewert für `HOME` ist das Heimatverzeichnis des Accounts, dem die `crontab` gehört.

❸ In dieser Zeile werden sieben Felder beschrieben: `minute`, `hour`, `mday`, `month`, `wday`, `who` und `command`. Die ersten Felder legen den Zeitpunkt fest, an dem ein Kommando laufen soll. Das Feld `minute` legt die Minute fest, das Feld `hour` die Stunde, das Feld `mday` den Tag des Monats. Im Feld `month` wird der Monat und im Feld `wday` der Wochentag festgelegt. Alle Felder müssen numerische Werte enthalten und die Zeitangaben sind im 24-Stunden-Format. Das Feld `who` gibt es nur in der Datei `/etc/crontab` und gibt den Account an, unter dem das Kommando laufen soll. In den `crontab`-Dateien einzelner Accounts existiert dieses Feld nicht. Im letzten Feld wird schließlich das auszuführende Kommando angegeben.

❹ Diese Zeile definiert die Zeitpunkte an denen das Kommando `atrun` laufen soll. Beachten Sie die Zeichenfolge `*/5` gefolgt von mehreren `*`-Zeichen. Das Zeichen `*` ist ein Platzhalter und steht für *jede* mögliche Zeit. Diese Zeile führt das Kommando `atrun` unter dem `root`-Account alle fünf Minuten aus. Mehr über das Kommando `atrun` erfahren Sie in der Hilfeseite atrun(8).

Bei den Kommandos können beliebige Optionen angegeben werden. Wenn das Kommando zu lang ist und auf der nächsten Zeile fortgesetzt werden soll, muss am Ende der Zeile das Fortsetzungszeichen (\) angegeben werden.

Bis auf das sechste Feld, das den Account angibt, sieht jede `crontab`-Datei so wie das Beispiel aus. Das sechste Feld existiert nur in der Systemdatei `/etc/crontab`. In den restlichen `crontab`-Dateien fehlt dieses Feld.

12.6.1. `crontab` installieren

> **Wichtig:** Die nachstehende Prozedur gilt nur für Benutzer-`crontabs`. Die System-`crontab` können Sie einfach mit Ihrem Lieblingseditor editieren. Das Werkzeug `cron` bemerkt, dass sich die Datei geändert hat und wird die neue Version benutzen. Lesen Sie bitte auch die FAQ zur Meldung root: not found (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/faq/admin.html#ROOT-NOT-FOUND-CRON-ERRORS).

Eine Benutzer-`crontab`, beispielsweise die Datei `crontab`, können Sie mit jedem Editor erstellen. Die Benutzer-`crontab` installieren Sie mit dem nachstehenden Befehl:

```
# crontab crontab
```

Das Argument zum Befehl `crontab` ist die vorher erstellte Datei `crontab`.

Der Befehl `crontab -l` zeigt die installierte `crontab`-Datei an.

Benutzer, die eine eigene `crontab`-Datei ohne Vorlage erstellen wollen, können den Befehl `crontab -e` verwenden. Dieser Befehl ruft einen Editor auf und installiert beim Verlassen des Editors die `crontab`-Datei.

Wollen Sie die installierte Benutzer-`crontab` entfernen, rufen Sie den Befehl `crontab` mit der Option `-r` auf.

12.7. Das rc-System für Systemdienste

Beigetragen von Tom Rhodes.

2002 wurde das **rc.d**-System von NetBSD zum Start von Systemdiensten in FreeBSD integriert. Die zu diesem System gehörenden Dateien sind im Verzeichnis `/etc/rc.d` abgelegt. Die Skripten in diesem Verzeichnis akzeptieren die Optionen `start`, `stop` und `restart`. Beispielsweise kann sshd(8) mit dem nachstehenden Kommando neu gestartet werden:

```
# /etc/rc.d/sshd restart
```

Analog können Sie andere Dienste starten und stoppen. Normalerweise werden die Dienste beim Systemstart über Einträge in der Datei rc.conf(5) automatisch gestartet. Der Network Address Translation Dæmon wird zum Beispiel mit dem folgenden Eintrag in `/etc/rc.conf` aktiviert:

```
natd_enable="YES"
```

Wenn dort bereits die Zeile `natd_enable="NO"` existiert, ändern Sie einfach `NO` in `YES`. Die rc-Skripten starten, wie unten beschrieben, auch abhängige Dienste.

Da das **rcNG**-System primär zum automatischen Starten und Stoppen von Systemdiensten dient, funktionieren die Optionen `start`, `stop` und `restart` nur, wenn die entsprechenden Variablen in `/etc/rc.conf` gesetzt sind. Beispielsweise funktioniert das Kommando `sshd restart` nur dann, wenn in `/etc/rc.conf` die Variable `sshd_enable` auf `YES` gesetzt wurde. Wenn Sie die Optionen `start`, `stop` oder `restart` unabhängig von den Einstellungen in `/etc/rc.conf` benutzen wollen, müssen Sie den Optionen mit dem Präfix “one” verwenden. Um

beispielsweise `sshd` unabhängig von den Einstellungen in `/etc/rc.conf` neu zu starten, benutzen Sie das nachstehende Kommando:

```
# /etc/rc.d/sshd onerestart
```

Ob ein Dienst in `/etc/rc.conf` aktiviert ist, können Sie leicht herausfinden, indem Sie das entsprechende `rc.d`-Skript mit der Option `rcvar` aufrufen. Ein Administrator kann beispielsweise wie folgt prüfen, ob der `sshd`-Dienst in `/etc/rc.conf` aktiviert ist:

```
# /etc/rc.d/sshd rcvar
# sshd
$sshd_enable=YES
```

> **Anmerkung:** Die zweite Zeile (`# sshd`) wird vom Kommando `sshd` ausgegeben; sie kennzeichnet nicht die Eingabeaufforderung von `root`.

Ob ein Dienst läuft, kann mit der Option `status` abgefragt werden. Das folgende Kommando überprüft, ob der `sshd` auch wirklich gestartet wurde:

```
# /etc/rc.d/sshd status
sshd is running as pid 433.
```

Einige Dienste können über die Option `reload` neu initialisiert werden. Dazu wird dem Dienst über ein Signal mitgeteilt, dass er seine Konfigurationsdateien neu einlesen soll. Oft wird dazu das Signal `SIGHUP` verwendet. Beachten Sie aber, dass nicht alle Dienste diese Option unterstützen.

Die meisten Systemdienste werden beim Systemstart vom **rc.d**-System gestartet. Zum Beispiel aktiviert das Skript `bgfsck` die Prüfung von Dateisystemen im Hintergrund. Das Skript gibt die folgende Meldung aus, wenn es gestartet wird:

```
Starting background file system checks in 60 seconds.
```

Viele Systemdienste hängen von anderen Diensten ab. NIS und andere RPC-basierende Systeme hängen beispielsweise von dem `rpcbind`-Dienst (portmapper) ab. Im Kopf der Startskripten befinden sich die Informationen über Abhängigkeiten von anderen Diensten und weitere Metadaten.Mithilfe dieser Daten bestimmt das Programm rcorder(8) beim Systemstart die Startreihenfolge der Dienste.

Folgende Schlüsselwörter müssen im Kopf aller Startskripten verwendet werden (da sie von rc.subr(8) zum “Aktivieren” des Startskripts benötigt werden:

- `PROVIDE`: Gibt die Namen der Dienste an, die mit dieser Datei zur Verfügung gestellt werden.

Die folgenden Schlüsselwörter können im Kopf des Startskripts angegeben werden. Sie sind zwar nicht unbedingt notwendig, sind aber hilfreich beim Umgang mit rcorder(8):

- `REQUIRE`: Gibt die Namen der Dienste an, von denen dieser Dienst abhängt. Diese Datei wird *nach* den angegebenen Diensten ausgeführt.
- `BEFORE`: Zählt Dienste auf, die auf diesen Dienst angewiesen sind. Diese Datei wird *vor* den angegebenen Diensten ausgeführt.

Durch das Verwenden dieser Schlüsselwörter kann ein Administrator die Startreihenfolge von Systemdiensten feingranuliert steuern, ohne mit den Schwierigkeiten des “runlevel”-Systems anderer UNIX Systeme kämpfen zu müssen.

Weitere Informationen über das `rc.d`-System finden sich in den Manualpages zu rc(8) sowie rc.subr(8). Wenn Sie Ihre eigenen `rc.d`-Skripte schreiben wollen, sollten Sie den Artikel Practical rc.d scripting in BSD (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/rc-scripting) lesen.

12.8. Einrichten von Netzwerkkarten

Beigetragen von Marc Fonvieille.

Ein Rechner ohne Netzanschluss ist heute nicht mehr vorstellbar. Die Konfiguration einer Netzwerkkarte gehört zu den alltäglichen Aufgaben eines FreeBSD Administrators.

12.8.1. Bestimmen des richtigen Treibers

Bevor Sie anfangen, sollten Sie das Modell Ihrer Karte kennen, wissen welchen Chip die Karte benutzt und bestimmen, ob es sich um eine PCI- oder ISA-Karte handelt. Eine Aufzählung der unterstützten PCI- und ISA-Karten finden Sie in der Liste der unterstützen Geräte. Schauen Sie nach, ob Ihre Karte dort aufgeführt ist.

Wenn Sie wissen, dass Ihre Karte unterstützt wird, müssen Sie den Treiber für Ihre Karte bestimmen. `/usr/src/sys/conf/NOTES` und `/usr/src/sys/`*`arch`*`/conf/NOTES` enthalten eine Liste der verfügbaren Treiber mit Informationen zu den unterstützten Chipsätzen und Karten. Wenn Sie sich nicht sicher sind, ob Sie den richtigen Treiber ausgewählt haben, lesen Sie die Hilfeseite des Treibers. Die Hilfeseite enthält weitere Informationen über die unterstützten Geräte und macht auch auf mögliche Probleme aufmerksam.

Wenn Sie eine gebräuchliche Karte besitzen, brauchen Sie meistens nicht lange nach dem passenden Treiber zu suchen. Die Treiber zu diesen Karten sind schon im `GENERIC`-Kernel enthalten und die Karte sollte während des Systemstarts erkannt werden:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
miibus0: <MII bus> on dc0
bmtphy0: <BCM5201 10/100baseTX PHY> PHY 1 on miibus0
bmtphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc0: Ethernet address: 00:a0:cc:da:da:da
dc0: [ITHREAD]
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
miibus1: <MII bus> on dc1
bmtphy1: <BCM5201 10/100baseTX PHY> PHY 1 on miibus1
bmtphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: Ethernet address: 00:a0:cc:da:da:db
dc1: [ITHREAD]
```

Im Beispiel erkennt das System zwei Karten, die den dc(4) Treiber benutzen.

Ist der Treiber für Ihre Netzwerkkarte nicht in `GENERIC` enthalten, müssen Sie den Treiber laden, um die Karte zu benutzen. Sie können den Treiber auf zwei Arten laden:

- Am einfachsten ist es, das Kernelmodul für Ihre Karte mit kldload(8) zu laden. Allerdings gibt es nicht für alle Karten Kernelmodule; zum Beispiel gibt es keine Kernelmodule für ISA-Karten.
- Alternativ können Sie den Treiber für die Karte fest in den Kernel einbinden. Schauen Sie sich dazu `/usr/src/sys/conf/NOTES`, `/usr/src/sys/`*`arch`*`/conf/NOTES` und die Hilfeseite des Treibers, den Sie in den Kernel einbinden möchten, an. Die Übersetzung des Kernels wird in Kapitel 9 beschrieben. Wenn Ihre Karte während des Systemstarts vom Kernel (`GENERIC`) erkannt wurde, müssen Sie den Kernel nicht neu übersetzen.

12.8.1.1. Windows-NDIS-Treiber einsetzen

Leider stellen nach wie vor viele Unternehmen die Spezifikationen ihrer Treiber der Open Source Gemeinde nicht zur Verfügung, weil sie diese Informationen als Geschäftsgeheimnisse betrachten. Daher haben die Entwickler von FreeBSD und anderen Betriebssystemen nur zwei Möglichkeiten. Entweder versuchen sie in einem aufwändigen Prozess den Treiber durch *Reverse Engineering* nachzubauen, oder sie versuchen, die vorhandenen Binärtreiber der Microsoft Windows-Plattform zu verwenden. Die meisten Entwickler, darunter auch die an FreeBSD beteiligten, haben sich für den zweiten Ansatz entschieden.

Bill Paul (wpaul) ist es zu verdanken, dass es seit eine “native” Unterstützung der *Network Driver Interface Specification* (NDIS) gibt. Der FreeBSD NDISulator (auch als Project Evil bekannt) nutzt den binären Windows-Treiber, indem er diesem vorgibt, unter Windows zu laufen. Da der ndis(4)-Treiber eine Windows-Binärdatei nutzt, kann er nur auf i386- und amd64-Systemen verwendet werden.

> **Anmerkung:** Der ndis(4)-Treiber unterstützt primär PCI-, CardBus- sowie PCMCIA-Geräte, USB-Geräte werden hingegen noch nicht unterstützt.

Um den NDISulator zu verwenden, benötigen Sie drei Dinge:

1. Die Kernelquellen
2. Den Windows XP-Binärtreiber (mit der Erweiterung `.SYS`)
3. Die Konfigurationsdatei des Windows XP-Treibers (mit der Erweiterung `.INF`)

Suchen Sie die Dateien für Ihre Karte. Diese befinden sich meistens auf einer beigelegten CD-ROM, oder können von der Internetseite des Herstellers heruntergeladen werden. In den folgenden Beispielen werden die Dateien `W32DRIVER.SYS` und `W32DRIVER.INF` verwendet.

> **Anmerkung:** Sie können einen Windows/i386-Treiber nicht unter FreeBSD/amd64 einsetzen, vielmehr benötigen Sie dafür einen Windows/amd64-Treiber.

Als Nächstes kompilieren Sie den binären Treiber, um ein Kernelmodul zu erzeugen. Dazu rufen Sie als `root` ndisgen(8) auf:

```
# ndisgen /path/to/W32DRIVER.INF /path/to/W32DRIVER.SYS
```

ndisgen(8) arbeitet interaktiv, benötigt es weitere Informationen, so fragt es Sie danach. Als Ergebnis erhalten Sie ein Kernelmodul im Arbeitsverzeichnis, das Sie wie folgt laden können:

```
# kldload ./W32DRIVER.ko
```

Neben dem vorhin erzeugten Kernelmodul müssen Sie auch die Kernelmodule `ndis.ko` und `if_ndis.ko` laden. Diese Module sollten automatisch geladen werden, wenn Sie ein von ndis(4) abhängiges Modul laden. Wollen Sie die Module hingegen manuell laden, geben Sie die folgenden Befehle ein:

```
# kldload ndis
# kldload if_ndis
```

Der erste Befehl lädt dabei den NDIS-Miniport-Treiber, der zweite das tatsächliche Netzwerkgerät.

Überprüfen Sie nun die Ausgabe von dmesg(8) auf eventuelle Fehler während des Ladevorgangs. Gab es dabei keine Probleme, sollten Sie eine Ausgabe ähnlich der folgenden erhalten:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

Ab jetzt können Sie mit dem Gerät `ndis0` wie mit jeder anderen Gerätedatei (etwa `dc0`) arbeiten.

Wie jedes Kernelmodul können auch die NDIS-Module beim Systemstart automatisch geladen werden. Dazu kopieren Sie das erzeugte Modul (`W32DRIVER_SYS.ko`) in das Verzeichnis `/boot/modules`. Danach fügen Sie die folgende Zeile in `/boot/loader.conf` ein:

```
W32DRIVER_SYS_load="YES"
```

12.8.2. Konfiguration von Netzwerkkarten

Nachdem der richtige Treiber für die Karte geladen ist, muss die Karte konfiguriert werden. Unter Umständen ist die Karte schon während der Installation mit **sysinstall** konfiguriert worden.

Das nachstehende Kommando zeigt die Konfiguration der Karten eines Systems an:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8802<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:db
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet 10baseT/UTP
        status: no carrier
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
        inet6 ::1 prefixlen 128
        inet 127.0.0.1 netmask 0xff000000
```

```
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
```

Im Beispiel werden Informationen zu den folgenden Geräten angezeigt:

- `dc0`: Der erste Ethernet-Adapter
- `dc1`: Der zweite Ethernet-Adapter
- `plip0`: Die parallele Schnittstelle (falls Ihr System über eine derartige Schnittstelle verfügt)
- `lo0`: Das Loopback-Gerät

Der Name der Netzwerkkarte wird aus dem Namen des Treibers und einer Zahl zusammengesetzt. Die Zahl gibt die Reihenfolge an, in der die Geräte beim Systemstart erkannt wurden. Die dritte Karte, die den sis(4) Treiber benutzt, würde beispielsweise `sis2` heißen.

Der Adapter `dc0` aus dem Beispiel ist aktiv. Sie erkennen das an den folgenden Hinweisen:

1. `UP` bedeutet, dass die Karte konfiguriert und aktiv ist.
2. Der Karte wurde die Internet-Adresse (`inet`) `192.168.1.3` zugewiesen.
3. Die Subnetzmaske ist richtig (`0xffffff00` entspricht `255.255.255.0`).
4. Die Broadcast-Adresse `192.168.1.255` ist richtig.
5. Die MAC-Adresse der Karte (`ether`) lautet `00:a0:cc:da:da:da`.
6. Die automatische Medienerkennung ist aktiviert (`media: Ethernet autoselect (100baseTX <full-duplex>)`). Der Adapter `dc1` benutzt das Medium `10baseT/UTP`. Weitere Informationen über die einstellbaren Medien entnehmen Sie bitte der Hilfeseite des Treibers.
7. Der Verbindungsstatus (`status`) ist `active`, das heißt es wurde ein Trägersignal entdeckt. Für `dc1` wird `status: no carrier` angezeigt. Das ist normal, wenn kein Kabel an der Karte angeschlossen ist.

Wäre die Karte nicht konfiguriert, würde die Ausgabe von ifconfig(8) so aussehen:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
```

Sie brauchen die Berechtigungen von `root`, um Ihre Karte zu konfigurieren. Die Konfiguration kann auf der Kommandozeile mit ifconfig(8) erfolgen, allerdings müsste sie dann nach jedem Neustart wiederholt werden. Dauerhaft wird die Karte in `/etc/rc.conf` konfiguriert.

Öffnen Sie `/etc/rc.conf` mit Ihrem Lieblingseditor und fügen Sie für jede Karte Ihres Systems eine Zeile hinzu. In dem hier diskutierten Fall wurden die nachstehenden Zeilen eingefügt:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Ersetzen Sie `dc0`, `dc1` usw. durch die Gerätenamen Ihrer Karten und setzen Sie die richtigen IP-Adressen ein. Die Hilfeseiten des Treibers und ifconfig(8) enthalten weitere Einzelheiten über verfügbare Optionen. Die Syntax von `/etc/rc.conf` wird in rc.conf(5) erklärt.

Wenn Sie das Netz während der Installation konfiguriert haben, existieren vielleicht schon Einträge für Ihre Karten. Überprüfen Sie `/etc/rc.conf` bevor Sie weitere Zeilen hinzufügen.

In `/etc/hosts` können Sie die Namen und IP-Adressen der Rechner Ihres LANs eintragen. Weitere Informationen entnehmen Sie bitte hosts(5) und `/usr/share/examples/etc/hosts`.

> **Anmerkung:** Soll Ihr System sich auch mit dem Internet verbinden können, müssen Sie Default-Gateway und Nameserver manuell konfigurieren:
>
> ```
> # echo 'defaultrouter="Ihr_Default_Gateway"' >> /etc/rc.conf
> # echo 'nameserver Ihr_DNS_Server' >> /etc/resolv.conf
> ```

12.8.3. Test und Fehlersuche

Nachdem Sie die notwendigen Änderungen in `/etc/rc.conf` vorgenommen haben, führen Sie einen Neustart Ihres Systems durch. Dadurch werden die Adapter konfiguriert und Sie stellen sicher, dass der Start ohne Konfigurationsfehler erfolgt. Alternativ können Sie auch lediglich die Netzwerkeinstellungen neu initialisieren:

```
# /etc/rc.d/netif restart
```

> **Anmerkung:** Haben Sie ein Default-Gateway definiert (in der Datei `/etc/rc.conf`), müssen Sie auch den folgenden Befehl ausführen:
>
> ```
> # /etc/rc.d/routing restart
> ```

Wenn das System gestartet ist, sollten Sie die Netzwerkkarten testen.

12.8.3.1. Test der Ethernet-Karte

Mit zwei Tests können Sie prüfen, ob die Ethernet-Karte richtig konfiguriert ist. Testen Sie zuerst mit `ping` den Adapter selbst und sprechen Sie dann eine andere Maschine im LAN an.

Zuerst, der Test des Adapters:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

Jetzt versuchen wir, eine andere Maschine im LAN zu erreichen:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Sie können auch den Namen der Maschine anstelle von `192.168.1.2` benutzen, wenn Sie `/etc/hosts` entsprechend eingerichtet haben.

12.8.3.2. Fehlersuche

Fehler zu beheben, ist immer sehr mühsam. Indem Sie die einfachen Sachen zuerst prüfen, erleichtern Sie sich die Aufgabe. Steckt das Netwerkkabel? Sind die Netzwerkdienste richtig konfiguriert? Funktioniert die Firewall? Wird die Netwerkkarte von FreeBSD unterstützt? Lesen Sie immer die Hardware-Informationen des Releases, bevor Sie einen Fehlerbericht einsenden. Aktualisieren Sie Ihre FreeBSD-Version auf -STABLE. Suchen Sie in den Archiven der Mailinglisten oder auf dem Internet nach bekannten Lösungen.

Wenn die Karte funktioniert, die Verbindungen aber zu langsam sind, lesen Sie bitte die Hilfeseite tuning(7). Prüfen Sie auch die Netzwerkkonfiguration, da falsche Einstellungen die Ursache für langsame Verbindungen sein können.

Wenn Sie viele `device timeout` Meldungen in den Systemprotokollen finden, prüfen Sie, dass es keinen Konflikt zwischen der Netzwerkkarte und anderen Geräten Ihres Systems gibt. Überprüfen Sie nochmals die Verkabelung. Unter Umständen benötigen Sie eine neue Netzwerkkarte.

Wenn Sie in den Systemprotokollen `watchdog timeout` Fehlermeldungen finden, kontrollieren Sie zuerst die Verkabelung. Überprüfen Sie dann, ob der PCI-Steckplatz der Karte Bus Mastering unterstützt. Auf einigen älteren Motherboards ist das nur für einen Steckplatz (meistens Steckplatz 0) der Fall. Lesen Sie in der Dokumentation Ihrer Karte und Ihres Motherboards nach, ob das vielleicht die Ursache des Problems sein könnte.

Die Meldung `No route to host` erscheint, wenn Ihr System ein Paket nicht zustellen kann. Das kann vorkommen weil beispielsweise keine Default-Route gesetzt wurde oder das Netzwerkkabel nicht richtig steckt. Schauen Sie in der Ausgabe von `netstat -rn` nach, ob eine Route zu dem Zielsystem existiert. Wenn nicht, lesen Sie bitte das Kapitel 32.

Die Meldung `ping: sendto: Permission denied` wird oft von einer falsch konfigurierten Firewall verursacht. Wenn keine Regeln definiert wurden, blockiert eine aktivierte Firewall alle Pakete, selbst einfache `ping`-Pakete. Weitere Informationen erhalten Sie in Kapitel 31.

Falls die Leistung der Karte schlecht ist, setzen Sie die Medienerkennung von `autoselect` (automatisch) auf das richtige Medium. In vielen Fällen löst diese Maßnahme Leistungsprobleme. Wenn nicht, prüfen Sie nochmal die Netzwerkeinstellungen und lesen Sie die Hilfeseite tuning(7).

12.9. Virtual Hosts

Ein gebräuchlicher Zweck von FreeBSD ist das virtuelle Hosting, bei dem ein Server im Netzwerk wie mehrere Server aussieht. Dies wird dadurch erreicht, dass einem Netzwerkinterface mehrere Netzwerk-Adressen zugewiesen werden.

Ein Netzwerkinterface hat eine “echte” Adresse und kann beliebig viele “alias” Adressen haben. Die Aliase werden durch entsprechende alias Einträge in `/etc/rc.conf` festgelegt.

Ein alias Eintrag für das Interface `fxp0` sieht wie folgt aus:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Beachten Sie, dass die Alias-Einträge mit `alias0` anfangen müssen und weiter hochgezählt werden, das heißt `_alias1`, `_alias2`, und so weiter. Die Konfiguration der Aliase hört bei der ersten fehlenden Zahl auf.

Die Berechnung der Alias-Netzwerkmasken ist wichtig, doch zum Glück einfach. Für jedes Interface muss es eine Adresse geben, die die Netzwerkmaske des Netzwerkes richtig beschreibt. Alle anderen Adressen in diesem Netzwerk haben dann eine Netzwerkmaske, die mit `1` gefüllt ist (also `255.255.255.255` oder hexadezimal `0xffffffff`).

Als Beispiel betrachten wir den Fall, in dem `fxp0` mit zwei Netzwerken verbunden ist: dem Netzwerk `10.1.1.0` mit der Netzwerkmaske `255.255.255.0` und dem Netzwerk `202.0.75.16` mit der Netzwerkmaske `255.255.255.240`. Das System soll die Adressen `10.1.1.1` bis `10.1.1.5` und `202.0.75.17` bis `202.0.75.20` belegen. Wie eben beschrieben, hat nur die erste Adresse in einem Netzwerk (hier `10.0.1.1` und `202.0.75.17`) die richtige Netzwerkmaske. Alle anderen Adressen (`10.1.1.2` bis `10.1.1.5` und `202.0.75.18` bis `202.0.75.20`) erhalten die Maske `255.255.255.255`.

Die folgenden Einträge in `/etc/rc.conf` konfigurieren den Adapter entsprechend dem Beispiel:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

12.10. Konfigurationsdateien

12.10.1. `/etc` Layout

Konfigurationsdateien finden sich in einigen Verzeichnissen unter anderem in:

`/etc`	Enthält generelle Konfigurationsinformationen, die Daten hier sind systemspezifisch.
`/etc/defaults`	Default Versionen der Konfigurationsdateien.
`/etc/mail`	Enthält die sendmail(8) Konfiguration und weitere MTA Konfigurationsdateien.

`/etc/ppp`	Hier findet sich die Konfiguration für die User- und Kernel-ppp Programme.
`/etc/namedb`	Das Vorgabeverzeichnis, in dem Daten von named(8) gehalten werden. Normalerweise werden hier `named.conf` und Zonendaten abgelegt.
`/usr/local/etc`	Installierte Anwendungen legen hier ihre Konfigurationsdateien ab. Dieses Verzeichnis kann Unterverzeichnisse für bestimmte Anwendungen enthalten.
`/usr/local/etc/rc.d`	Ort für Start- und Stopskripten installierter Anwendungen.
`/var/db`	Automatisch generierte systemspezifische Datenbanken, wie die Paket-Datenbank oder die `locate`-Datenbank.

12.10.2. Hostnamen

12.10.2.1. `/etc/resolv.conf`

Wie der FreeBSD-Resolver auf das Internet Domain Name System (DNS) zugreift, wird in `/etc/resolv.conf` festgelegt.

Die gebräuchlichsten Einträge in `/etc/resolv.conf` sind:

`nameserver`	Die IP-Adresse eines Nameservers, den der Resolver abfragen soll. Bis zu drei Server werden in der Reihenfolge, in der sie aufgezählt sind, abgefragt.
`search`	Suchliste mit Domain-Namen zum Auflösen von Hostnamen. Die Liste wird normalerweise durch den Domain-Teil des lokalen Hostnamens festgelegt.
`domain`	Der lokale Domain-Name.

Beispiel für eine typische `resolv.conf`:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

> **Anmerkung:** Nur eine der Anweisungen `search` oder `domain` sollte benutzt werden.

Wenn Sie DHCP benutzen, überschreibt dhclient(8) für gewöhnlich `resolv.conf` mit den Informationen vom DHCP-Server.

12.10.2.2. `/etc/hosts`

`/etc/hosts` ist eine einfache textbasierte Datenbank, die aus alten Internetzeiten stammt. Zusammen mit DNS und NIS stellt sie eine Abbildung zwischen Namen und IP-Adressen zur Verfügung. Anstatt named(8) zu konfigurieren, können hier lokale Rechner, die über ein LAN verbunden sind, eingetragen werden. Lokale Einträge für gebräuchliche Internet-Adressen in `/etc/hosts` verhindern die Abfrage eines externen Servers und beschleunigen die Namensauflösung.

```
# $FreeBSD$
#
#
# Host Database
#
# This file should contain the addresses and aliases for local hosts that
# share this file.  Replace 'my.domain' below with the domainname of your
# machine.
#
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1         localhost localhost.my.domain
127.0.0.1       localhost localhost.my.domain
#
# Imaginary network.
#10.0.0.2       myname.my.domain myname
#10.0.0.3       myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#   10.0.0.0    -   10.255.255.255
#   172.16.0.0  -   172.31.255.255
#   192.168.0.0 -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  Do not try to invent your own network
# numbers but instead get one from your network provider (if any) or
# from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfriNIC.)
#
```

`/etc/hosts` hat ein einfaches Format:

```
[Internet Adresse] [Offizieller Hostname] [Alias1] [Alias2] ...
```

Zum Beispiel:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

Weitere Informationen entnehmen Sie bitte hosts(5).

12.10.3. Konfiguration von Logdateien

12.10.3.1. `syslog.conf`

`syslog.conf` ist die Konfigurationsdatei von syslogd(8). Sie legt fest, welche **syslog** Meldungen in welche Logdateien geschrieben werden.

```
# $FreeBSD$
```

```
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.
*.err;kern.debug;auth.notice;mail.crit          /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                      /var/log/security
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
*.err                                           root
*.notice;news.err                               root
*.alert                                         root
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote log host named loghost
#*.*                                            @loghost
# uncomment these if you’re running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice
!startslip
*.*                                             /var/log/slip.log
!ppp
*.*                                             /var/log/ppp.log
```

Weitere Informationen enthält syslog.conf(5).

12.10.3.2. `newsyslog.conf`

Die Konfigurationsdatei für newsyslog(8), das normalerweise von cron(8) aufgerufen wird, ist `newsyslog.conf`. newsyslog(8) stellt fest, ob Logdateien archiviert oder verschoben werden müssen. So wird `logfile` nach `logfile.0` geschoben und `logfile.0` nach `logfile.1` usw. Zudem können Logdateien mit gzip(1) komprimiert werden. Die Namen der Logdateien sind dann `logfile.0.gz`, `logfile.1.gz` usw.

`newsyslog.conf` legt fest, welche Logdateien wann bearbeitet und wie viele Dateien behalten werden. Logdateien können auf Basis ihrer Größe oder zu einem gewissen Zeitpunkt archiviert bzw. umbenannt werden.

```
# configuration file for newsyslog
# $FreeBSD$
#
# filename          [owner:group]    mode count size when [ZB] [/pid_file] [sig_num]
/var/log/cron                         600  3     100  *     Z
/var/log/amd.log                      644  7     100  *     Z
/var/log/kerberos.log                 644  7     100  *     Z
/var/log/lpd-errs                     644  7     100  *     Z
/var/log/maillog                      644  7     *    @T00  Z
```

```
/var/log/sendmail.st                    644  10    *    168   B
/var/log/messages                       644  5     100  *     Z
/var/log/all.log                        600  7     *    @T00  Z
/var/log/slip.log                       600  3     100  *     Z
/var/log/ppp.log                        600  3     100  *     Z
/var/log/security                       600  10    100  *     Z
/var/log/wtmp                           644  3     *    @01T05 B
/var/log/daily.log                      640  7     *    @T00  Z
/var/log/weekly.log                     640  5     1    $W6D0 Z
/var/log/monthly.log                    640  12    *    $M1D0 Z
/var/log/console.log                    640  5     100  *     Z
```

Um mehr zu erfahren, lesen Sie bitte newsyslog(8).

12.10.4. `sysctl.conf`

`sysctl.conf` sieht ähnlich wie `rc.conf` aus. Werte werden in der Form `Variable=Wert` gesetzt. Die angegebenen Werte werden gesetzt, nachdem sich das System bereits im Mehrbenutzermodus befindet. Allerdings lassen sich im Mehrbenutzermodus nicht alle Werte setzen.

Um das Protokollieren von fatalen Signalen abzustellen und Benutzer daran zu hindern, von anderen Benutzern gestartete Prozesse zu sehen, können Sie in der Datei `sysctl.conf` die folgenden Variablen setzen:

```
# Do not log fatal signal exits (e.g. sig 11)
kern.logsigexit=0

# Prevent users from seeing information about processes that
# are being run under another UID.
security.bsd.see_other_uids=0
```

12.11. Einstellungen mit sysctl

Mit sysctl(8) können Sie Änderungen an einem laufenden FreeBSD-System vornehmen. Unter anderem können Optionen des TCP/IP-Stacks oder des virtuellen Speichermanagements verändert werden. Unter der Hand eines erfahrenen Systemadministrators kann dies die Systemperformance erheblich verbessern. Über 500 Variablen können mit sysctl(8) gelesen und gesetzt werden.

Der Hauptzweck von sysctl(8) besteht darin, Systemeinstellungen zu lesen und zu verändern.

Alle auslesbaren Variablen werden wie folgt angezeigt:

```
% sysctl -a
```

Sie können auch eine spezielle Variable, z.B. `kern.maxproc` lesen:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Um eine Variable zu setzen, benutzen Sie die Syntax *`Variable`*=*`Wert`*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Mit sysctl können Sie Strings, Zahlen oder Boolean-Werte setzen. Bei Boolean-Werten setzen sie `1` für wahr und `0` für falsch.

Wenn Sie Variablen automatisch während des Systemstarts setzen wollen, fügen Sie die Variablen in die Datei `/etc/sysctl.conf` ein. Weiteres entnehmen Sie bitte der Hilfeseite sysctl.conf(5) und dem Abschnitt 12.10.4.

12.11.1. Schreibgeschützte Variablen

Contributed by Tom Rhodes.

Schreibgeschützte sysctl-Variablen können nur während des Systemstarts verändert werden.

Beispielsweise hat cardbus(4) auf einigen Laptops Schwierigkeiten, Speicherbereiche zu erkennen. Es treten dann Fehlermeldungen wie die folgende auf:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Um dieses Problem zu lösen, muss eine schreibgeschützte sysctl-Variable verändert werden. Eine OID kann in der Datei `/boot/loader.conf` überschrieben werden. Die Datei `/boot/defaults/loader.conf` enthält Vorgabewwerte für sysctl-Variablen.

Das oben erwähnte Problem wird durch die Angabe von `hw.pci.allow_unsupported_io_range=1` in `/boot/loader.conf` gelöst. Danach sollte cardbus(4) fehlerfrei funktionieren.

12.12. Tuning von Laufwerken

12.12.1. Sysctl Variablen

12.12.1.1. `vfs.vmiodirenable`

Die Variable `vfs.vmiodirenable` besitzt in der Voreinstellung den Wert 1. Die Variable kann auf den Wert 0 (ausgeschaltet) oder 1 (angeschaltet) gesetzt werden. Sie steuert, wie Verzeichnisse vom System zwischengespeichert werden. Die meisten Verzeichnisse sind klein und benutzen nur ein einzelnes Fragment, typischerweise 1 kB, im Dateisystem. Im Buffer-Cache verbrauchen sie mit 512 Bytes noch weniger Platz. Ist die Variable ausgeschaltet (auf 0) wird der Buffer-Cache nur eine limitierte Anzahl Verzeichnisse zwischenspeichern, auch wenn das System über sehr viel Speicher verfügt. Ist die Variable aktiviert (auf 1), kann der Buffer-Cache den VM-Page-Cache benutzen, um Verzeichnisse zwischenzuspeichern. Der ganze Speicher steht damit zum Zwischenspeichern von Verzeichnissen zur Verfügung. Der Nachteil bei dieser Vorgehensweise ist, dass zum Zwischenspeichern eines Verzeichnisses mindestens eine physikalische Seite im Speicher, die normalerweise 4 kB groß ist, anstelle von 512 Bytes gebraucht wird. Wir empfehlen, diese Option aktiviert zu lassen, wenn Sie Dienste zur Verfügung stellen, die viele Dateien manipulieren. Beispiele für solche Dienste sind Web-Caches, große Mail-Systeme oder Netnews. Die aktivierte Variable vermindert, trotz des verschwendeten Speichers, in aller Regel nicht die Leistung des Systems, obwohl Sie das nachprüfen sollten.

12.12.1.2. `vfs.write_behind`

In der Voreinstellung besitzt die Variable `vfs.write_behind` den Wert `1` (aktiviert). Mit dieser Einstellung schreibt das Dateisystem anfallende vollständige Cluster, die besonders beim sequentiellen Schreiben großer Dateien auftreten, direkt auf das Medium aus. Dies verhindert, dass sich im Buffer-Cache veränderte Puffer (*dirty buffers*) ansammeln, die die I/O-Verarbeitung nicht mehr beschleunigen würden. Unter bestimmten Umständen blockiert diese Funktion allerdings Prozesse. Setzen Sie in diesem Fall die Variable `vfs.write_behind` auf den Wert `0`.

12.12.1.3. `vfs.hirunningspace`

Die Variable `vfs.hirunningspace` bestimmt systemweit die Menge ausstehender Schreiboperationen, die dem Platten-Controller zu jedem beliebigen Zeitpunkt übergeben werden können. Normalerweise können Sie den Vorgabewert verwenden. Auf Systemen mit vielen Platten kann der Wert aber auf 4 bis 5 *Megabyte* erhöht werden. Beachten Sie, dass ein zu hoher Wert (größer als der Schreib-Schwellwert des Buffer-Caches) zu Leistungverlusten führen kann. Setzen Sie den Wert daher nicht zu hoch! Hohe Werte können auch Leseoperationen verzögern, die gleichzeitig mit Schreiboperationen ausgeführt werden.

Es gibt weitere Variablen, mit denen Sie den Buffer-Cache und den VM-Page-Cache beeinflussen können. Wir raten Ihnen allerdings davon ab, diese Variablen zu verändern, da das VM-System den virtuellen Speicher selbst sehr gut verwaltet.

12.12.1.4. `vm.swap_idle_enabled`

Die Variable `vm.swap_idle_enabled` ist für große Mehrbenutzer-Systeme gedacht, auf denen sich viele Benutzer an- und abmelden und auf denen es viele Prozesse im Leerlauf (*idle*) gibt. Solche Systeme fragen kontinuierlich freien Speicher an. Wenn Sie die Variable `vm.swap_idle_enabled` aktivieren, können Sie die Auslagerungs-Hysterese von Seiten mit den Variablen `vm.swap_idle_threshold1` und `vm.swap_idle_threshold2` einstellen. Die Schwellwerte beider Variablen geben die Zeit in Sekunden an, in denen sich ein Prozess im Leerlauf befinden muss. Wenn die Werte so eingestellt sind, dass Seiten früher als nach dem normalen Algorithmus ausgelagert werden, verschafft das dem Auslagerungs-Prozess mehr Luft. Aktivieren Sie diese Funktion nur, wenn Sie sie wirklich benötigen: Die Speicherseiten werden eher früher als später ausgelagert. Der Platz im Swap-Bereich wird dadurch schneller verbraucht und die Plattenaktivitäten steigen an. Auf kleine Systeme hat diese Funktion spürbare Auswirkungen. Auf großen Systemen, die sowieso schon Seiten auslagern müssen, können ganze Prozesse leichter in den Speicher geladen oder ausgelagert werden.

12.12.1.5. `hw.ata.wc`

In FreeBSD 4.3 wurde versucht, den IDE-Schreib-Zwischenspeicher abzustellen. Obwohl dies die Bandbreite zum Schreiben auf IDE-Platten verringerte, wurde es aus Gründen der Datenkonsistenz als notwenig angesehen. Der Kern des Problems ist, dass IDE-Platten keine zuverlässige Aussage über das Ende eines Schreibvorgangs treffen. Wenn der Schreib-Zwischenspeicher aktiviert ist, werden die Daten nicht in der Reihenfolge ihres Eintreffens geschrieben. Es kann sogar passieren, dass das Schreiben mancher Blöcke im Fall von starker Plattenaktivität auf unbefristete Zeit verzögert wird. Ein Absturz oder Stromausfall zu dieser Zeit kann die Dateisysteme erheblich beschädigen. Wir entschieden uns daher für die sichere Variante und stellten den Schreib-Zwischenspeicher ab. Leider war damit auch ein großer Leistungsverlust verbunden, so dass wir die Variable nach dem Release wieder aktiviert haben. Sie sollten den Wert der Variable `hw.ata.wc` auf Ihrem System überprüfen. Wenn der Schreib-Zwischenspeicher abgestellt ist, können Sie ihn aktivieren, indem Sie die Variable auf den Wert 1 setzen. Dies muss zum Zeitpunkt des Systemstarts im Boot-Loader geschehen. Eine Änderung der Variable, nachdem der Kernel gestartet ist, hat keine Auswirkungen.

Weitere Informationen finden Sie in ata(4).

12.12.1.6. `SCSI_DELAY` (`kern.cam.scsi_delay`)

Mit der Kerneloption `SCSI_DELAY` kann die Dauer des Systemstarts verringert werden. Der Vorgabewert ist recht hoch und er verzögert den Systemstart um 15 oder mehr Sekunden. Normalerweise kann dieser Wert, insbesondere mit modernen Laufwerken, auf 5 Sekunden heruntergesetzt werden (durch Setzen der `sysctl`-Variable `kern.cam.scsi_delay`). Die Variable sowie die Kerneloption verwenden für die Zeitangabe Millisekunden und *nicht* Sekunden.

12.12.2. Soft Updates

Mit tunefs(8) lassen sich Feineinstellungen an Dateisystemen vornehmen. Das Programm hat verschiedene Optionen, von denen hier nur Soft Updates betrachtet werden. Soft Updates werden wie folgt ein- und ausgeschaltet:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Ein eingehängtes Dateisystem kann nicht mit tunefs(8) modifiziert werden. Soft Updates werden am besten im Single-User Modus aktiviert, bevor Partitionen eingehangen sind.

Durch Einsatz eines Zwischenspeichers wird die Performance im Bereich der Metadaten, vorwiegend beim Anlegen und Löschen von Dateien, gesteigert. Wir empfehlen, Soft Updates auf allen Dateisystemen zu aktivieren. Allerdings sollten Sie sich über die zwei Nachteile von Soft Updates bewusst sein: Erstens garantieren Soft Updates zwar die Konsistenz der Daten im Fall eines Absturzes, aber es kann leicht passieren, dass das Dateisystem über mehrere Sekunden oder gar eine Minute nicht synchronisiert wurde. Im Fall eines Absturzes verlieren Sie mit Soft Updates unter Umständen mehr Daten als ohne. Zweitens verzögern Soft Updates die Freigabe von Datenblöcken. Eine größere Aktualisierung eines fast vollen Dateisystems, wie dem Root-Dateisystem, z.B. während eines `make installworld`, kann das Dateisystem vollaufen lassen. Dadurch würde die Aktualisierung fehlschlagen.

12.12.2.1. Details über Soft Updates

Es gibt zwei klassische Herangehensweisen, wie man die Metadaten des Dateisystems (also Daten über Dateien, wie inode Bereiche oder Verzeichniseinträge) aktualisiert auf die Platte zurückschreibt:

Das historisch übliche Verfahren waren synchrone Updates der Metadaten, d. h. wenn eine Änderung an einem Verzeichnis nötig war, wurde anschließend gewartet, bis diese Änderung tatsächlich auf die Platte zurückgeschrieben worden war. Der *Inhalt* der Dateien wurde im “Buffer Cache” zwischengespeichert und asynchron irgendwann später auf die Platte geschrieben. Der Vorteil dieser Implementierung ist, dass sie sicher funktioniert. Wenn während eines Updates ein Ausfall erfolgt, haben die Metadaten immer einen konsistenten Zustand. Eine Datei ist entweder komplett angelegt oder gar nicht. Wenn die Datenblöcke einer Datei im Fall eines Absturzes noch nicht den Weg aus dem “Buffer Cache” auf die Platte gefunden haben, kann fsck(8) das Dateisystem reparieren, indem es die Dateilänge einfach auf 0 setzt. Außerdem ist die Implementierung einfach und überschaubar. Der Nachteil ist, dass Änderungen der Metadaten sehr langsam vor sich gehen. Ein `rm -r` beispielsweise fasst alle Dateien eines Verzeichnisses der Reihe nach an, aber jede dieser Änderungen am Verzeichnis (Löschen einer Datei) wird einzeln synchron auf die Platte geschrieben. Gleiches beim Auspacken großer Hierarchien (`tar -x`).

Der zweite Fall sind asynchrone Metadaten-Updates. Das ist z. B. der Standard bei Linux/ext2fs oder die Variante `mount -o async` für *BSD UFS. Man schickt die Updates der Metadaten einfach auch noch über den “Buffer

Cache", sie werden also zwischen die Updates der normalen Daten eingeschoben. Vorteil ist, dass man nun nicht mehr auf jeden Update warten muss, Operationen, die zahlreiche Metadaten ändern, werden also viel schneller. Auch hier ist die Implementierung sehr einfach und wenig anfällig für Fehler. Nachteil ist, dass keinerlei Konsistenz des Dateisystems mehr gesichert ist. Wenn mitten in einer Operation, die viele Metadaten ändert, ein Ausfall erfolgt (Stromausfall, drücken des Reset-Tasters), dann ist das Dateisystem anschließend in einem unbestimmten Zustand. Niemand kann genau sagen, was noch geschrieben worden ist und was nicht mehr; die Datenblöcke einer Datei können schon auf der Platte stehen, während die inode Tabelle oder das zugehörige Verzeichnis nicht mehr aktualisiert worden ist. Man kann praktisch kein `fsck` mehr implementieren, das diesen Zustand wieder reparieren kann, da die dazu nötigen Informationen einfach auf der Platte fehlen. Wenn ein Dateisystem derart beschädigt worden ist, kann man es nur neu erzeugen (newfs(8)) und die Daten vom Backup zurückspielen.

Der historische Ausweg aus diesem Dilemma war ein *dirty region logging* (auch als *Journalling* bezeichnet, wenngleich dieser Begriff nicht immer gleich benutzt und manchmal auch für andere Formen von Transaktionsprotokollen gebraucht wird). Man schreibt die Metadaten-Updates zwar synchron, aber nur in einen kleinen Plattenbereich, die *logging area*. Von da aus werden sie dann asynchron auf ihre eigentlichen Bereiche verteilt. Da die *logging area* ein kleines zusammenhängendes Stückchen ist, haben die Schreibköpfe der Platte bei massiven Operationen auf Metadaten keine allzu großen Wege zurückzulegen, so dass alles ein ganzes Stück schneller geht als bei klassischen synchronen Updates. Die Komplexität der Implementierung hält sich ebenfalls in Grenzen, somit auch die Anfälligkeit für Fehler. Als Nachteil ergibt sich, dass Metadaten zweimal auf die Platte geschrieben werden müssen (einmal in die *logging area*, einmal an die richtige Stelle), so dass das im Falle regulärer Arbeit (also keine gehäuften Metadatenoperationen) eine "Pessimisierung" des Falls der synchronen Updates eintritt, es wird alles langsamer. Dafür hat man als Vorteil, dass im Falle eines Crashes der konsistente Zustand dadurch erzielbar ist, dass die angefangenen Operationen aus dem *dirty region log* entweder zu Ende ausgeführt oder komplett verworfen werden, wodurch das Dateisystem schnell wieder zur Verfügung steht.

Die Lösung von Kirk McKusick, dem Schöpfer von Berkeley FFS, waren *Soft Updates*: die notwendigen Updates der Metadaten werden im Speicher gehalten und dann sortiert auf die Platte geschrieben ("ordered metadata updates"). Dadurch hat man den Effekt, dass im Falle massiver Metadaten-Änderungen spätere Operationen die vorhergehenden, noch nicht auf die Platte geschriebenen Updates desselben Elements im Speicher "einholen". Alle Operationen, auf ein Verzeichnis beispielsweise, werden also in der Regel noch im Speicher abgewickelt, bevor der Update überhaupt auf die Platte geschrieben wird (die dazugehörigen Datenblöcke werden natürlich auch so sortiert, dass sie nicht vor ihren Metadaten auf der Platte sind). Im Fall eines Absturzes hat man ein implizites "log rewind": alle Operationen, die noch nicht den Weg auf die Platte gefunden haben, sehen danach so aus, als hätten sie nie stattgefunden. Man hat so also den konsistenten Zustand von ca. 30 bis 60 Sekunden früher sichergestellt. Der verwendete Algorithmus garantiert dabei, dass alle tatsächlich benutzten Ressourcen auch in den entsprechenden Bitmaps (Block- und inode Tabellen) als belegt markiert sind. Der einzige Fehler, der auftreten kann, ist, dass Ressourcen noch als "belegt" markiert sind, die tatsächlich "frei" sind. fsck(8) erkennt dies und korrigiert diese nicht mehr belegten Ressourcen. Die Notwendigkeit eines Dateisystem-Checks darf aus diesem Grunde auch ignoriert und das Dateisystem mittels `mount -f` zwangsweise eingebunden werden. Um noch allozierte Ressourcen freizugeben muss später ein fsck(8) nachgeholt werden. Das ist dann auch die Idee des *background fsck*: beim Starten des Systems wird lediglich ein *Schnappschuss* des Filesystems gemacht, mit dem fsck(8) dann später arbeiten kann. Alle Dateisysteme dürfen "unsauber" eingebunden werden und das System kann sofort in den Multiuser-Modus gehen. Danach wird ein Hintergrund-`fsck` für die Dateisysteme gestartet, die dies benötigen, um möglicherweise irrtümlich belegte Ressourcen freizugeben. (Dateisysteme ohne *Soft Updates* benötigen natürlich immer noch den üblichen (Vordergrund-)`fsck`, bevor sie eingebunden werden können.)

Der Vorteil ist, dass die Metadaten-Operationen beinahe so schnell ablaufen wie im asynchronen Fall (also durchaus auch schneller als beim "logging", das ja die Metadaten immer zweimal schreiben muss). Als Nachteil stehen dem die Komplexität des Codes (mit einer erhöhten Fehlerwahrscheinlichkeit in einem bezüglich Datenverlust hoch sensiblen Bereich) und ein erhöhter Speicherverbrauch entgegen. Außerdem muss man sich an einige Eigenheiten

gewöhnen: Nach einem Absturz ist ein etwas älterer Stand auf der Platte – statt einer leeren, aber bereits angelegten Datei (wie nach einem herkömmlichen `fsck` Lauf) ist auf einem Dateisystem mit *Soft Updates* keine Spur der entsprechenden Datei mehr zu sehen, da weder die Metadaten noch der Dateiinhalt je auf die Platte geschrieben wurden. Weiterhin kann der Platz nach einem `rm -r` nicht sofort wieder als verfügbar markiert werden, sondern erst dann, wenn der Update auch auf die Platte vermittelt worden ist. Dies kann besonders dann Probleme bereiten, wenn große Datenmengen in einem Dateisystem ersetzt werden, das nicht genügend Platz hat, um alle Dateien zweimal unterzubringen.

12.13. Einstellungen von Kernel Limits

12.13.1. Datei und Prozeß Limits

12.13.1.1. `kern.maxfiles`

Abhängig von den Anforderungen Ihres Systems kann `kern.maxfiles` erhöht oder erniedrigt werden. Die Variable legt die maximale Anzahl von Dateideskriptoren auf Ihrem System fest. Wenn die Dateideskriptoren aufgebraucht sind, werden Sie die Meldung `file: table is full` wiederholt im Puffer für Systemmeldungen sehen. Den Inhalt des Puffers können Sie sich mit `dmesg` anzeigen lassen.

Jede offene Datei, jedes Socket und jede FIFO verbraucht einen Dateideskriptor. Auf “dicken” Produktionsservern können leicht Tausende Dateideskriptoren benötigt werden, abhängig von der Art und Anzahl der gleichzeitig laufenden Dienste.

In älteren FreeBSD-Versionen wurde die Voreinstellung von `kern.maxfile` aus der Kernelkonfigurationsoption `maxusers` bestimmt. `kern.maxfiles` wächst proportional mit dem Wert von `maxusers`. Wenn Sie einen angepassten Kernel kompilieren, empfiehlt es sich diese Option entsprechend der maximalen Benutzerzahl Ihres Systems einzustellen. Obwohl auf einer Produktionsmaschine vielleicht nicht 256 Benutzer gleichzeitig angemeldet sind, können die benötigten Ressourcen ähnlich denen eines großen Webservers sein.

Die Variable `kern.maxusers` wird beim Systemstart automatisch aus dem zur Verfügung stehenden Hauptspeicher bestimmt. Im laufenden Betrieb kann dieser Wert aus der (nur lesbaren) sysctl-Variable `kern.maxusers` ermittelt werden. Falls ein System für diese Variable einen anderen Wert benötigt, kann der Wert über den Loader angepasst werden. Häufig verwendete Werte sind dabei 64, 128, sowie 256. Es ist empfehlenswert, die Anzahl der Dateideskriptoren nicht auf einen Wert größer 256 zu setzen, es sei denn, Sie benötigen wirklich eine riesige Anzahl von ihnen. Viele der von `kern.maxusers` auf einen Standardwert gesetzten Parameter können beim Systemstart oder im laufenden Betrieb in der Datei `/boot/loader.conf` (sehen Sie sich dazu auch loader.conf(5) sowie die Datei `/boot/defaults/loader.conf` an) an Ihre Bedürfnisse angepasst werden, so wie es bereits an anderer Stelle dieses Dokuments beschrieben ist.

Ältere FreeBSD-Versionen setzen diesen Wert selbst, wenn Sie in der Konfigurationsdatei den Wert `0` [1] angeben. Wenn Sie den Wert selbst bestimmen wollen, sollten Sie `maxusers` mindestens auf `4` setzen. Dies gilt insbesondere dann, wenn Sie beabsichtigen, das X Window-System zu benutzen oder Software zu kompilieren. Der Grund dafür ist, dass der wichtigste Wert, der durch `maxusers` bestimmt wird, die maximale Anzahl an Prozessen ist, die auf `20 + 16 * maxusers` gesetzt wird. Wenn Sie also `maxusers` auf 1 setzen, können gleichzeitig nur 36 Prozesse laufen, von denen ungefähr 18 schon beim Booten des Systems gestartet werden. Dazu kommen nochmals etwa 15 Prozesse beim Start des X Window-Systems. Selbst eine einfache Aufgabe wie das Lesen einer Manualpage benötigt neun Prozesse zum Filtern, Dekomprimieren und Betrachten der Datei. Für die meisten Benutzer sollte es ausreichen,

`maxusers` auf 64 zu setzen, womit 1044 gleichzeitige Prozesse zur Verfügung stehen. Wenn Sie allerdings den gefürchteten Fehler proc table full beim Start eines Programms oder auf einem Server mit einer großen Benutzerzahl (wie `ftp.FreeBSD.org`) sehen, dann sollten Sie den Wert nochmals erhöhen und den Kernel neu bauen.

> **Anmerkung:** Die Anzahl der Benutzer, die sich auf einem Rechner anmelden kann, wird durch `maxusers` *nicht* begrenzt. Der Wert dieser Variablen legt neben der möglichen Anzahl der Prozesse eines Benutzers weitere sinnvolle Größen für bestimmte Systemtabellen fest.

12.13.1.2. `kern.ipc.somaxconn`

Die Variable `kern.ipc.somaxconn` beschränkt die Größe der Warteschlange (*Listen-Queue*) für neue TCP-Verbindungen. Der Vorgabewert von `128` ist normalerweise zu klein, um neue Verbindungen auf einem stark ausgelasteten Webserver zuverlässig zu handhaben. Auf solchen Servern sollte der Wert auf `1024` oder höher gesetzt werden. Ein Dienst (z.B. sendmail(8), oder **Apache**) kann die Größe der Queue selbst einschränken. Oft gibt es die Möglichkeit, die Größe der Listen-Queue in einer Konfigurationsdatei einzustellen. Eine große Listen-Queue übersteht vielleicht auch einen Denial of Service Angriff (DoS).

12.13.2. Netzwerk Limits

Die Kerneloption `NMBCLUSTERS` schreibt die Anzahl der Netzwerkpuffer (Mbufs) fest, die das System besitzt. Eine zu geringe Anzahl Mbufs auf einem Server mit viel Netzwerkverkehr verringert die Leistung von FreeBSD. Jeder Mbuf-Cluster nimmt ungefähr 2 kB Speicher in Anspruch, so dass ein Wert von 1024 insgesamt 2 Megabyte Speicher für Netzwerkpuffer im System reserviert. Wie viele Cluster benötigt werden, lässt sich durch eine einfache Berechnung herausfinden. Wenn Sie einen Webserver besitzen, der maximal 1000 gleichzeitige Verbindungen servieren soll und jede der Verbindungen je einen 16 kB großen Puffer zum Senden und Empfangen braucht, brauchen Sie ungefähr 32 MB Speicher für Netzwerkpuffer. Als Daumenregel verdoppeln Sie diese Zahl, so dass sich für `NMBCLUSTERS` der Wert 2x32 MB / 2 kB = 32768 ergibt. Für Maschinen mit viel Speicher sollten Werte zwischen 4096 und 32768 genommen werden. Sie können diesen Wert nicht willkürlich erhöhen, da dies bereits zu einem Absturz beim Systemstart führen kann. Mit der Option `-m` von netstat(1) können Sie den Gebrauch der Netzwerkpuffer kontrollieren.

Die Netzwerkpuffer können beim Systemstart mit der Loader-Variablen `kern.ipc.nmbclusters` eingestellt werden. Nur auf älteren FreeBSD-Systemen müssen Sie die Kerneloption `NMBCLUSTERS` verwenden.

Die Anzahl der sendfile(2) Puffer muss auf ausgelasteten Servern, die den Systemaufruf sendfile(2) oft verwenden, vielleicht erhöht werden. Dazu können Sie die Kerneloption `NSFBUFS` verwenden oder die Anzahl der Puffer in `/boot/loader.conf` (siehe loader(8)) setzen. Die Puffer sollten erhöht werden, wenn Sie Prozesse im Zustand `sfbufa` sehen. Die schreibgeschützte sysctl-Variable `kern.ipc.nsfbufs` zeigt die Anzahl eingerichteten Puffer im Kernel. Der Wert dieser Variablen wird normalerweise von `kern.maxusers` bestimmt. Manchmal muss die Pufferanzahl jedoch manuell eingestellt werden.

> **Wichtig:** Auch wenn ein Socket nicht blockierend angelegt wurde, kann der Aufruf von sendfile(2) blockieren, um auf freie `struct sf_buf` Puffer zu warten.

12.13.2.1. `net.inet.ip.portrange.*`

Die sysctl-Variable `net.inet.ip.portrange.*` legt die Portnummern für TCP- und UDP-Sockets fest. Es gibt drei Bereiche: den niedrigen Bereich, den normalen Bereich und den hohen Bereich. Die meisten Netzprogramme benutzen den normalen Bereich. Dieser Bereich umfasst in der Voreinstellung die Portnummern 500 bis 5000 und wird durch die Variablen `net.inet.ip.portrange.first` und `net.inet.ip.portrange.last` festgelegt. Die festgelegten Bereiche für Portnummern werden von ausgehenden Verbindungen benutzt. Unter bestimmten Umständen, beispielsweise auf stark ausgelasteten Proxy-Servern, sind alle Portnummern für ausgehende Verbindungen belegt. Bereiche für Portnummern spielen auf Servern keine Rolle, die hauptsächlich eingehende Verbindungen verarbeiten (wie ein normaler Webserver) oder nur eine begrenzte Anzahl ausgehender Verbindungen öffnen (beispielsweise ein Mail-Relay). Wenn Sie keine freien Portnummern mehr haben, sollten Sie die Variable `net.inet.ip.portrange.last` langsam erhöhen. Ein Wert von `10000`, `20000` oder `30000` ist angemessen. Beachten Sie auch eine vorhandene Firewall, wenn Sie die Bereiche für Portnummern ändern. Einige Firewalls sperren große Bereiche (normalerweise aus den kleinen Portnummern) und erwarten, dass hohe Portnummern für ausgehende Verbindungen verwendet werden. Daher kann es erforderlich sein, den Wert von `net.inet.ip.portrange.first` zu erhöhen.

12.13.2.2. TCP Bandwidth Delay Product Begrenzung

Die TCP Bandwidth Delay Product Begrenzung gleicht TCP/Vegas von NetBSD. Die Begrenzung wird aktiviert, indem Sie die sysctl-Variable `net.inet.tcp.inflight.enable` auf den Wert `1` setzen. Das System wird dann versuchen, für jede Verbindung, das Produkt aus der Übertragungsrate und der Verzögerungszeit zu bestimmen. Dieses Produkt begrenzt die Datenmenge, die für einen optimales Durchsatz zwischengespeichert werden muss.

Diese Begrenzung ist nützlich, wenn Sie Daten über Verbindungen mit einem hohen Produkt aus Übertragungsrate und Verzögerungszeit wie Modems, Gigabit-Ethernet oder schnellen WANs, zur Verfügung stellen. Insbesondere wirkt sich die Begrenzung aus, wenn die Verbindung die TCP-Option *Window-scaling* verwendet oder große Sende-Fenster (*send window*) benutzt. Schalten Sie die Debug-Meldungen aus, wenn Sie die Begrenzung aktiviert haben. Dazu setzen Sie die Variable `net.inet.tcp.inflight.debug` auf `0`. Auf Produktions-Systemen sollten Sie zudem die Variable `net.inet.tcp.inflight.min` mindestens auf den Wert `6144` setzen. Allerdings kann ein zu hoher Wert, abhängig von der Verbindung, die Begrenzungsfunktion unwirksam machen. Die Begrenzung reduziert die Datenmenge in den Queues von Routern und Switches, sowie die Datenmenge in der Queue der lokalen Netzwerkkarte. Die Verzögerungszeit (*Round Trip Time*) für interaktive Anwendungen sinkt, da weniger Pakete zwischengespeichert werden. Dies gilt besonders für Verbindungen über langsame Modems. Die Begrenzung wirkt sich allerdings nur auf das Versenden von Daten aus (Uploads, Server). Auf den Empfang von Daten (Downloads) hat die Begrenzung keine Auswirkungen.

Die Variable `net.inet.tcp.inflight.stab` sollte *nicht* angepasst werden. Der Vorgabewert der Variablen beträgt `20`, das heißt es werden maximal zwei Pakete zu dem Produkt aus Übertragungsrate und Verzögerungszeit addiert. Dies stabilisiert den Algorithmus und verbessert die Reaktionszeit auf Veränderungen. Bei langsamen Verbindungen können sich aber die Laufzeiten der Pakete erhöhen (ohne diesen Algorithmus wären sie allerdings noch höher). In solchen Fällen können Sie versuchen, den Wert der Variablen auf `15`, `10` oder `5` zu erniedrigen. Gleichzeitig müssen Sie vielleicht auch `net.inet.tcp.inflight.min` auf einen kleineren Wert (beispielsweise `3500`) setzen. Ändern Sie diese Variablen nur ab, wenn Sie keine anderen Möglichkeiten mehr haben.

12.13.3. Virtueller Speicher (*Virtual Memory*)

12.13.3.1. `kern.maxvnodes`

Ein vnode ist die interne Darstellung einer Datei oder eines Verzeichnisses. Die Erhöhung der Anzahl der für das Betriebssystem verfügbaren vnodes verringert also die Schreib- und Lesezugriffe auf Ihre Festplatte. vnodes werden im Normalfall vom Betriebssystem automatisch vergeben und müssen nicht von Ihnen angepasst werden. In einigen Fällen stellt der Zugriff auf eine Platte allerdings einen Flaschenhals dar, daher sollten Sie in diesem Fall die Anzahl der möglichen vnodes erhöhen, um dieses Problem zu beheben. Beachten Sie dabei aber die Größe des inaktiven und freien Hauptspeichers.

Um die Anzahl der derzeit verwendeten vnodes zu sehen, geben Sie Folgendes ein:

```
# sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

Die maximal mögliche Anzahl der vnodes erhalten Sie durch die Eingabe von:

```
# sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

Wenn sich die Anzahl der genutzten vnodes dem maximal möglichen Wert nähert, sollten Sie den Wert `kern.maxvnodes` zuerst um etwa 1.000 erhöhen. Beobachten Sie danach die Anzahl der vom System genutzten `vfs.numvnodes`. Nähert sich der Wert wiederum dem definierten Maximum, müssen Sie `kern.maxvnodes` nochmals erhöhen. Sie sollten nun eine Änderung Ihres Speicherverbrauches (etwa über top(1)) registrieren können und über mehr aktiven Speicher verfügen.

12.14. Hinzufügen von Swap-Bereichen

Egal wie vorausschauend Sie planen, manchmal entspricht ein System einfach nicht Ihren Erwartungen. Es ist leicht, mehr Swap-Bereiche hinzuzufügen. Dazu stehen Ihnen drei Wege offen: Sie können eine neue Platte einbauen, den Swap-Bereich über NFS ansprechen oder eine Swap-Datei auf einer existierenden Partition einrichten.

Für Informationen zur Verschlüsselung von Swap-Partitionen, zu den dabei möglichen Optionen sowie zu den Gründen für eine Verschlüsselung des Auslagerungsspeichers lesen Sie bitte Abschnitt 19.17 des Handbuchs.

12.14.1. Swap auf einer neuen Festplatte

Der einfachste Weg, zusätzlich einen Swap-Bereich einzurichten, ist der Einbau einer neuen Platte, die Sie sowieso gebrauchen können. Die Anordnung von Swap-Bereichen wird in Abschnitt 12.2 des Handbuchs besprochen.

12.14.2. Swap-Bereiche über NFS

Swap-Bereiche über NFS sollten Sie nur dann einsetzen, wenn Sie über keine lokale Platte verfügen, da es durch die zur Verfügung stehende Bandbreite limitiert wird und außerdem den NFS-Server zusätzlich belastet.

12.14.3. Swap-Dateien

Sie können eine Datei festgelegter Größe als Swap-Bereich nutzen. Im folgenden Beispiel werden wir eine 64 MB große Datei mit Namen `/usr/swap0` benutzen, Sie können natürlich einen beliebigen Namen für den Swap-Bereich benutzen.

Beispiel 12-1. Erstellen einer Swap-Datei

1. Der `GENERIC`-Kernel unterstützt bereits RAM-Disks (md(4)), welche für diese Aktion benötigt werden. Wenn Sie einen eigenen Kernel erstellen, vergewissern Sie sicher, dass die folgende Zeile in ihrer Kernel-Konfigurationsdatei enthalten ist:
   ```
   device   md
   ```
   Informationen, wie man einen eigenen Kernel erstellen kann, erhalten Sie in Kapitel 9.
2. Legen Sie die Swap-Datei `/usr/swap0` an:
   ```
   # dd if=/dev/zero of=/usr/swap0 bs=1024k count=64
   ```
3. Setzen Sie die richtigen Berechtigungen für `/usr/swap0`:
   ```
   # chmod 0600 /usr/swap0
   ```
4. Aktivieren Sie die Swap-Datei `/etc/rc.conf`:
   ```
   swapfile="/usr/swap0"   # Set to name of swapfile if aux swapfile desired.
   ```
5. Um die Swap-Datei zu aktivieren, führen Sie entweder einen Neustart durch oder geben das folgende Kommando ein:
   ```
   # mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0
   ```

12.15. Energie- und Ressourcenverwaltung

Verfasst von Hiten Pandya und Tom Rhodes.

Es ist wichtig, Hardware effizient einzusetzen. Vor der Einführung des *Advanced Configuration and Power Interface* (ACPI) konnten Stromverbrauch und Wärmeabgabe eines Systems nur schlecht von Betriebssystemen gesteuert werden. Die Hardware wurde vom BIOS gesteuert, was die Kontrolle der Energieverwaltung für den Anwender erschwerte. Das *Advanced Power Management (APM)* erlaubte es lediglich, einige wenige Funktionen zu steuern, obwohl die Überwachung von Energie- und Ressourcenverbrauch zu den wichtigsten Aufgaben eines Betriebssystems gehört, um auf verschiedene Ereignisse, beispielsweise einen unerwarteten Temperaturanstieg, reagieren können.

Dieser Abschnitt erklärt das Advanced Configuration and Power Interface (ACPI).

12.15.1. Was ist ACPI?

Advanced Configuration and Power Interface (ACPI) ist ein Standard verschiedener Hersteller, der die Verwaltung von Hardware und Energiesparfunktionen festlegt. Die ACPI-Funktionen können von einem Betriebssystem gesteuert werden. Der Vorgänger des ACPI, “Advanced Power Management” (APM), erwies sich in modernen Systemen als unzureichend.

12.15.2. Mängel des Advanced Power Managements (APM)

Das *Advanced Power Management (APM)* steuert den Energieverbrauch eines Systems auf Basis der Systemaktivität. Das APM-BIOS wird von dem Hersteller des Systems zur Verfügung gestellt und ist auf die spezielle Hardware angepasst. Der APM-Treiber des Betriebssystems greift auf das *APM Software Interface* zu, das den Energieverbrauch regelt. APM findet sich in der Regel nur noch in Systemen, die vor 2001 produziert wurden.

Das APM hat hauptsächlich vier Probleme. Erstens läuft die Energieverwaltung unabhängig vom Betriebssystem in einem (herstellerspezifischen) BIOS. Beispielsweise kann das APM-BIOS die Festplatten nach einer konfigurierbaren Zeit ohne die Zustimmung des Betriebssystems herunterfahren. Zweitens befindet sich die ganze APM-Logik im BIOS; das Betriebssystem hat gar keine APM-Komponenten. Bei Problemen mit dem APM-BIOS muss das Flash-ROM aktualisiert werden. Diese Prozedur ist gefährlich, da sie im Fehlerfall das System unbrauchbar machen kann. Zum Dritten ist APM eine Technik, die herstellerspezifisch ist und nicht koordiniert wird. Fehler im BIOS eines Herstellers werden nicht unbedingt im BIOS anderer Hersteller korrigiert. Das letzte Problem ist, dass im APM-BIOS nicht genügend Platz vorhanden ist, um eine durchdachte oder eine auf den Zweck der Maschine zugeschnittene Energieverwaltung zu implementieren.

Das *Plug and Play BIOS (PNPBIOS)* war ebenfalls unzureichend. Das PNPBIOS verwendet eine 16-Bit-Technik. Damit das Betriebssystem das PNPBIOS ansprechen kann, muss es in einer 16-Bit-Emulation laufen.

Der APM-Treiber von FreeBSD ist in der Hilfeseite apm(4) beschrieben.

12.15.3. Konfiguration des ACPI

Das Modul `acpi.ko` wird standardmäßig beim Systemstart vom loader(8) geladen und sollte daher *nicht* fest in den Kernel eingebunden werden. Dadurch kann `acpi.ko` ohne einen Neubau des Kernels ersetzt werden und das Modul ist leichter zu testen. Wenn Sie in der Ausgabe von dmesg(8) das Wort ACPI sehen, ist das Modul geladen worden. Das ACPI-Modul im laufenden Betrieb zu laden, führt oft nicht zum gewünschten Ergebnis. Treten bei Ihrem System Probleme auf, können Sie ACPI auch komplett deaktivieren. Dazu definieren Sie die Variable `hint.acpi.0.disabled="1"` in der Datei `/boot/loader.conf`. Alternativ können Sie die Variable auch am loader(8)-Prompt eingeben. Das Modul kann im laufenden Betrieb nicht entfernt werden, da es zur Kommunikation mit der Hardware verwendet wird.

> **Anmerkung:** ACPI und APM können nicht zusammen verwendet werden. Das zuletzt geladene Modul beendet sich, sobald es bemerkt, dass das andere Modul geladen ist.

Mit acpiconf(8) können Sie das System in einen Ruhemodus (*sleep mode*) versetzen. Es gibt verschiedene Modi (von `1` bis `5`), die Sie auf der Kommandozeile mit `-s` angeben können. Für die meisten Anwender sind die Modi `1` und `3` völlig ausreichend. Der Modus `5` schaltet das System aus (*Soft-off*) und entspricht dem folgenden Befehl:

```
# halt -p
```

Verschiedene Optionen können als sysctl(8)-Variablen gesetzt werden. Lesen Sie dazu die Manualpages zu acpi(4) sowie acpiconf(8).

12.16. ACPI-Fehlersuche

Verfasst von Nate Lawson. Mit Beiträgen von Peter Schultz und Tom Rhodes.

ACPI ist ein gänzlich neuer Weg, um Geräte aufzufinden und deren Stromverbrauch zu regulieren. Weiterhin bietet ACPI einen einheitlichen Zugriff auf Geräte, die vorher vom BIOS verwaltet wurden. Es werden zwar Fortschritte gemacht, dass ACPI auf allen Systemen läuft, doch tauchen immer wieder Fehler auf: fehlerhafter Bytecode der ACPI-Machine-Language (AML) einiger Systemplatinen, ein unvollständiges FreeBSD-Kernel-Subsystem oder Fehler im ACPI-CA-Interpreter von Intel.

Dieser Abschnitt hilft Ihnen, zusammen mit den Betreuern des FreeBSD-ACPI-Subsystems, Fehlerquellen zu finden und Fehler zu beseitigen. Danke, dass Sie diesen Abschnitt lesen; hoffentlich hilft er, Ihre Systemprobleme zu lösen.

12.16.1. Fehlerberichte einreichen

Anmerkung: Bevor Sie einen Fehlerbericht einreichen, stellen Sie bitte sicher, dass Ihr BIOS und die Firmware Ihres Controllers aktuell sind.

Wenn Sie sofort einen Fehlerbericht einsenden wollen, schicken Sie bitte die folgenden Informationen an die Mailingliste freebsd-acpi (mailto:freebsd-acpi@FreeBSD.org):

- Beschreiben Sie den Fehler und alle Umstände, unter denen der Fehler auftritt. Geben Sie ebenfalls den Typ und das Modell Ihres Systems an. Wenn Sie einen neuen Fehler entdeckt haben, versuchen Sie möglichst genau zu beschreiben, wann der Fehler das erste Mal aufgetreten ist.
- Die Ausgabe von dmesg(8) nach der Eingabe von `boot -v`. Geben Sie auch alle Fehlermeldungen an, die erscheinen, wenn Sie den Fehler provozieren.
- Die Ausgabe von dmesg(8) nach der Eingabe von `boot -v` und mit deaktiviertem ACPI, wenn das Problem ohne ACPI nicht auftritt.
- Die Ausgabe von `sysctl hw.acpi`. Dieses Kommando zeigt die vom System unterstützten ACPI-Funktionen an.
- Die URL, unter der die ACPI-Source-Language (ASL) liegt. Schicken Sie bitte *nicht* die ASL an die Mailingliste, da die ASL sehr groß sein kann. Eine Kopie der ASL erstellen Sie mit dem nachstehenden Befehl:

  ```
  # acpidump -td > name-system.asl
  ```

  Setzen Sie bitte für *`name`* den Namen Ihres Kontos und für *`system`* den Hersteller und das Modell Ihres Systems ein. Zum Beispiel: `njl-FooCo6000.asl`.

Obwohl die meisten Entwickler die Mailingliste freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) lesen, sollten Sie Fehlerberichte an die Liste freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) schicken. Seien Sie bitte geduldig; wir haben alle Arbeit außerhalb des Projekts. Wenn der Fehler nicht offensichtlich ist, bitten wir Sie vielleicht, einen offiziellen Fehlerbericht (PR) mit send-pr(1) einzusenden. Geben Sie im Fehlerbericht bitte dieselben Informationen wie oben an. Mithilfe der PRs verfolgen und lösen wir Probleme. Senden Sie bitte keinen PR ein, ohne vorher den Fehlerbericht an die Liste freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) zu senden. Wir benutzen die PRs als Erinnerung an bestehende Probleme und nicht zum Sammeln aller Probleme. Es kann sein, dass der Fehler schon von jemand anderem gemeldet wurde.

12.16.2. ACPI-Grundlagen

ACPI gibt es in allen modernen Rechnern der ia32- (x86), ia64- (Itanium) und amd64- (AMD) Architektur. Der vollständige Standard bietet Funktionen zur Steuerung und Verwaltung der CPU-Leistung, der Stromversorgung, von Wärmebereichen, Batterien, eingebetteten Controllern und Bussen. Auf den meisten Systemen wird nicht der vollständige Standard implementiert. Arbeitsplatzrechner besitzen meist nur Funktionen zur Verwaltung der Busse, während Notebooks Funktionen zur Temperaturkontrolle und Ruhezustände besitzen.

Ein ACPI konformes System besitzt verschiedene Komponenten. Die BIOS- und Chipsatz-Hersteller stellen mehrere statische Tabellen bereit (zum Beispiel die Fixed-ACPI-Description-Table, FADT). Die Tabellen enthalten beispielsweise die mit SMP-Systemen benutzte APIC-Map, Konfigurationsregister und einfache Konfigurationen. Zusätzlich gibt es die Differentiated-System-Description-Table (DSDT), die Bytecode enthält. Die Tabelle ordnet Geräte und Methoden in einem baumartigen Namensraum an.

Ein ACPI-Treiber muss die statischen Tabellen einlesen, einen Interpreter für den Bytecode bereitstellen und die Gerätetreiber im Kernel so modifizieren, dass sie mit dem ACPI-Subsystem kommunizieren. Für FreeBSD, Linux und NetBSD hat Intel den Interpreter ACPI-CA, zur Verfügung gestellt. Der Quelltext zu ACPI-CA befindet sich im Verzeichnis `src/sys/contrib/dev/acpica`. Die Schnittstelle von ACPI-CA zu FreeBSD befindet sich unter `src/sys/dev/acpica/Osd`. Treiber, die verschiedene ACPI-Geräte implementieren, befinden sich im Verzeichnis `src/sys/dev/acpica`.

12.16.3. Häufige Probleme

Damit ACPI richtig funktioniert, müssen alle Teile funktionieren. Im Folgenden finden Sie eine Liste mit Problemen und möglichen Umgehungen oder Fehlerbehebungen. Die Liste ist nach der Häufigkeit, mit der die Probleme auftreten, sortiert.

12.16.3.1. Mausprobleme

Es kann vorkommen, dass die Maus nicht mehr funktioniert, wenn Sie nach einem Suspend weiterarbeiten wollen. Ist dies bei Ihnen der Fall, reicht es meistens aus, den Eintrag `hint.psm.0.flags="0x3000"` in Ihre `/boot/loader.conf` aufzunehmen. Besteht das Problem weiterhin, sollten Sie einen Fehlerbericht an das FreeBSD Project senden.

12.16.3.2. Suspend/Resume

ACPI kennt drei Suspend-to-RAM-Zustände (STR): `S1`-`S3`. Es gibt einen Suspend-to-Disk-Zustand: `S4`. Der Zustand `S5` wird Soft-Off genannt. In diesem Zustand befindet sich ein Rechner, wenn die Stromversorgung angeschlossen ist, der Rechner aber nicht hochgefahren ist. Der Zustand `S4` kann auf zwei Arten implementiert werden: `S4`BIOS und `S4`OS. Im ersten Fall wird der Suspend-to-Disk-Zustand durch das BIOS hergestellt im zweiten Fall alleine durch das Betriebssystem.

> **Anmerkung:** Die Suspend-Zustände sind Ruhezustände, in denen der Rechner weniger Energie als im Normalbetrieb benötigt. Resume bezeichnet die Rückkehr zum Normalbetrieb.

Die Suspend-Zustände können Sie mit dem Kommando `sysctl hw.acpi` ermitteln. Das Folgende könnte beispielsweise ausgegeben werden:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Diese Ausgabe besagt, dass mit dem Befehl `acpiconf -s` die Zustände `S3`, `S4`OS und `S5` eingestellt werden können. Hätte `s4bios` den Wert `1`, gäbe es den Zustand `S4`BIOS anstelle von `S4`OS.

Wenn Sie die Suspend- und Resume-Funktionen testen, fangen Sie mit dem `S1`-Zustand an, wenn er angeboten wird. Dieser Zustand wird am ehesten funktionieren, da der Zustand wenig Treiber-Unterstützung benötigt. Der Zustand `S2` ist ähnlich wie `S1`, allerdings hat ihn noch niemand implementiert. Als nächstes sollten Sie den Zustand `S3` ausprobieren. Dies ist der tiefste STR-Schlafzustand. Dieser Zustand ist auf massive Treiber-Unterstützung angewiesen, um die Geräte wieder richtig zu initialisieren. Wenn Sie Probleme mit diesem Zustand haben, können Sie die Mailingliste freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) anschreiben. Erwarten Sie allerdings nicht zu viel: Es gibt viele Treiber und Geräte, an denen noch gearbeitet und getestet wird.

Ein häufiges Problem mit Suspend/Resume ist, dass viele Gerätetreiber ihre Firmware, Register und Gerätespeicher nicht korrekt speichern, wiederherstellen und/oder reinitialisieren. Um dieses Problem zu lösen, sollten Sie zuerst die folgenden Befehle ausführen:

```
# sysctl debug.bootverbose=1
# sysctl debug.acpi.suspend_bounce=1
# acpiconf -s 3
```

Dieser Test emuliert einen Suspend/Resume-Zyklus für alle Geräte (ohne dass diese dabei wirklich in den Status `S3` wechseln). In vielen Fällen reicht dies bereits aus, um Probleme (beispielsweise verlorener Firmware-Status, Timeouts, hängende Geräte) zu entdecken. Beachten Sie dabei, dass das Gerät bei diesem Test nicht wirklich in den Status `S3` wechseln. Es kann also vorkommen, dass manche Geräte weiterhin mit Strom versorgt werden (dies wäre bei einem wirklichen Wechsel in den Status `S3` NICHT möglich. Andere Geräte werden normal weiterarbeiten, weil sie über keine Suspend/Resume-Funktionen verfügen.

Schwierigere Fälle können den Einsatz zusätzlicher Hardware (beispielsweise serielle Ports/Kabel für die Verbindung über eine serielle Konsole oder Firewire-Ports/Kabel für dcons(4)) sowie Kenntnisse im Bereich Kerneldebugging erforderlich machen.

Um das Problem einzugrenzen, entfernen Sie soviele Treiber wie möglich aus dem Kernel. Sie können das Problem isolieren, indem Sie einen Treiber nach dem anderen laden, bis der Fehler wieder auftritt. Typischerweise verursachen binäre Treiber wie `nvidia.ko`, X11-Grafiktreiber und USB-Treiber die meisten Fehler, hingegen laufen Ethernet-Treiber für gewöhnlich sehr zuverlässig. Wenn ein Treiber zuverlässig geladen und entfernt werden kann, können Sie den Vorgang automatisieren, indem Sie die entsprechenden Kommandos in die Dateien `/etc/rc.suspend` und `/etc/rc.resume` einfügen. In den Dateien finden Sie ein deaktiviertes Beispiel, das einen Treiber lädt und wieder entfernt. Ist die Bildschirmanzeige bei der Wiederaufnahme des Betriebs gestört, setzen Sie bitte die Variable `hw.acpi.reset_video` auf `0`. Versuchen Sie auch, die Variable `hw.acpi.sleep_delay` auf kürzere Zeitspannen zu setzen.

Die Suspend- und Resume-Funktionen können Sie auch auf einer neuen Linux-Distribution mit ACPI testen. Wenn es mit Linux funktioniert, liegt das Problem wahrscheinlich bei einem FreeBSD-Treiber. Es hilft uns, das Problem zu lösen, wenn Sie feststellen können, welcher Treiber das Problem verursacht. Beachten Sie bitte, dass die ACPI-Entwickler normalerweise keine anderen Treiber pflegen (beispielsweise Sound- oder ATA-Treiber). Es ist wohl das beste, die Ergebnisse der Fehlersuche an die Mailingliste freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) und den Entwickler des Treibers zu schicken. Wenn Ihnen danach ist, versuchen Sie, den Fehler in der Resume-Funktion zu finden, indem Sie einige printf(3)-Anweisungen in den Code des fehlerhaften Treibers einfügen.

Schließlich können Sie ACPI noch abschalten und stattdessen APM verwenden. Wenn die Suspend- und Resume-Funktionen mit APM funktionieren, sollten Sie vielleicht besser APM verwenden (insbesondere mit alter Hardware von vor dem Jahr 2000). Die Hersteller benötigten einige Zeit, um ACPI korrekt zu implementieren, daher gibt es mit älterer Hardware oft ACPI-Probleme.

12.16.3.3. Temporäre oder permanente Systemhänger

Die meisten Systemhänger entstehen durch verlorene Interrupts oder einen Interrupt-Sturm. Probleme werden verursacht durch die Art, in der das BIOS Interrupts vor dem Systemstart konfiguriert, durch eine fehlerhafte APIC-Tabelle und durch die Zustellung des System-Control-Interrupts (SCI).

Anhand der Ausgabe des Befehls `vmstat -i` können Sie verlorene Interrupts von einem Interrupt-Sturm unterscheiden. Untersuchen Sie die Ausgabezeile, die `acpi0` enthält. Ein Interrupt-Sturm liegt vor, wenn der Zähler öfter als ein paar Mal pro Sekunde hochgezählt wird. Wenn sich das System aufgehangen hat, versuchen Sie mit der Tastenkombination **Ctrl**+**Alt**+**Esc** in den Debugger DDB zu gelangen. Geben Sie dort den Befehl `show interrupts` ein.

Wenn Sie Interrupt-Probleme haben, ist es vorerst wohl am besten, APIC zu deaktivieren. Tragen Sie dazu die Zeile `hint.apic.0.disabled="1"` in `loader.conf` ein.

12.16.3.4. Abstürze (Panics)

Panics werden so schnell wie möglich behoben; mit ACPI kommt es aber selten dazu. Zuerst sollten Sie die Panic reproduzieren und dann versuchen einen *backtrace* (eine Rückverfolgung der Funktionsaufrufe) zu erstellen. Richten Sie dazu den DDB über die serielle Schnittstelle (siehe Abschnitt 27.6.5.3) oder eine gesonderte dump(8)-Partition ein. In DDB können Sie den *backtrace* mit dem Kommando `tr` erstellen. Falls Sie den *backtrace* vom Bildschirm abschreiben müssen, schreiben Sie bitte mindestens die fünf ersten und die fünf letzten Zeile der Ausgabe auf.

Versuchen Sie anschließend, das Problem durch einen Neustart ohne ACPI zu beseitigen. Wenn das funktioniert hat, können Sie versuchen, das verantwortliche ACPI-Subsystem durch Setzen der Variablen `debug.acpi.disable` herauszufinden. Die Hilfeseite acpi(4) enthält dazu einige Beispiele.

12.16.3.5. Nach einem Suspend oder einem Stopp startet das System wieder

Setzen Sie zuerst in loader.conf(5) die Variable `hw.acpi.disable_on_poweroff` auf `0`. Damit wird verhindert, dass ACPI während des Systemabschlusses die Bearbeitung verschiedener Ereignisse deaktiviert. Auf manchen Systemen muss die Variable den Wert `1` besitzen (die Voreinstellung). Normalerweise wird der unerwünschte Neustart des Systems durch Setzen dieser Variablen behoben.

12.16.3.6. Andere Probleme

Wenn Sie weitere Probleme mit ACPI haben (Umgang mit einer Docking-Station, nicht erkannte Geräte), schicken Sie bitte eine Beschreibung an die Mailingliste. Allerdings kann es sein, dass einige Probleme von noch unvollständigen Teilen des ACPI-Subsystems abhängen und es etwas dauern kann bis diese Teile fertig sind. Seien Sie geduldig und rechnen Sie damit, dass wir Ihnen Fehlerbehebungen zum Testen senden.

12.16.4. ASL, `acpidump` und IASL

Ein häufiges Problem ist fehlerhafter Bytecode des BIOS-Herstellers. Dies erkennen Sie an Kernelmeldungen auf der Konsole wie die folgende:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Oft können Sie das Problem dadurch lösen, dass Sie eine aktuelle BIOS-Version einspielen. Die meisten Meldungen auf der Konsole sind harmlos, wenn aber beispielsweise der Batteriestatus falsch angezeigt wird, können Sie in den Meldungen nach Problemen mit der AML-Machine-Language (AML) suchen. Der Bytecode der AML wird aus der ACPI-Source-Language (ASL) übersetzt und in einer Tabelle, der DSDT, abgelegt. Eine Kopie der ASL können Sie mit dem Befehl acpidump(8) erstellen. Verwenden Sie mit diesem Befehl sowohl die Option `-t` (die Inhalte der statischen Tabellen anzeigen) als auch die Option `-d` (die AML in ASL zurückübersetzen). Ein Beispiel für die Syntax finden Sie im Abschnitt Fehlerberichte einreichen.

Sie können einfach prüfen, ob sich die ASL übersetzen lässt. Für gewöhnlich können Sie Warnungen während des Übersetzens ignorieren. Fehlermeldungen führen normal dazu, dass ACPI fehlerhaft arbeitet. Ihre ASL übersetzen Sie mit dem nachstehenden Kommando:

```
# iasl ihre.asl
```

12.16.5. Die ASL reparieren

Auf lange Sicht ist es unser Ziel, dass ACPI ohne Eingriffe des Benutzers läuft. Zurzeit entwickeln wir allerdings noch Umgehungen für Fehler der BIOS-Hersteller. Der Microsoft-Interpreter (`acpi.sys` und `acpiec.sys`) prüft die ASL nicht streng gegen den Standard. Daher reparieren BIOS-Hersteller, die ACPI nur unter Windows testen, ihre ASL nicht. Wir hoffen, dass wir das vom Standard abweichende Verhalten des Microsoft-Interpreters dokumentieren und in FreeBSD replizieren können. Dadurch müssen Benutzer ihre ASL nicht selbst reparieren. Sie können Ihre ASL selbst reparieren, wenn Sie ein Problem umgehen und uns helfen möchten. Senden Sie uns bitte die mit diff(1) erstellte Differenz zwischen alter und neuer ASL. Wir werden versuchen, den Interpreter ACPI-CA zu korrigieren, damit die Fehlerbehebung nicht mehr erforderlich ist.

Die nachfolgende Liste enthält häufige Fehlermeldungen, deren Ursache und eine Beschreibung, wie die Fehler korrigiert werden:

12.16.5.1. Abhängigkeiten vom Betriebssystem

Einige AMLs gehen davon aus, dass die Welt ausschließlich aus verschiedenen Windows-Versionen besteht. FreeBSD kann vorgeben, irgendein Betriebssystem zu sein. Versuchen Sie das Betriebssystem, das Sie in der ASL finden, in der Datei `/boot/loader.conf` anzugeben: `hw.acpi.osname="Windows 2001"`.

12.16.5.2. Fehlende Return-Anweisungen

Einige Methoden verzichten auf die vom Standard vorgeschriebene Rückgabe eines Wertes. Obwohl der Interpreter ACPI-CA dies nicht beheben kann, besitzt FreeBSD die Möglichkeit, den Rückgabewert implizit zu setzen. Wenn Sie wissen, welcher Wert zurückgegeben werden muss, können Sie die fehlenden Return-Anweisungen selbst einsetzen. Die Option `-f` zwingt `iasl`, die ASL zu übersetzen.

12.16.5.3. Überschreiben der vorgegebenen AML

Nachdem Sie Ihre ASL in der Datei `ihre.asl` angepasst haben, übersetzen Sie die ASL wie folgt:

```
# iasl ihre.asl
```

Mit der Option `-f` erzwingen Sie das Erstellen der AML auch wenn während der Übersetzung Fehler auftreten. Beachten Sie, dass einige Fehler, wie fehlende Return-Anweisungen, automatisch vom Interpreter umgangen werden.

In der Voreinstellung erstellt der Befehl `iasl` die Ausgabedatei `DSDT.aml`. Wenn Sie diese Datei anstelle der fehlerhaften Kopie des BIOS laden wollen, editieren Sie `/boot/loader.conf` wie folgt:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Stellen Sie bitte sicher, dass sich die Datei `DSDT.aml` im Verzeichnis `/boot` befindet.

12.16.6. ACPI-Meldungen zur Fehlersuche erzeugen

Der ACPI-Treiber besitzt flexible Möglichkeiten zur Fehlersuche. Sie können sowohl die zu untersuchenden Subsysteme als auch die zu erzeugenden Ausgaben festlegen. Die zu untersuchenden Subsysteme werden als so genannte “layers” angegeben. Die Subsysteme sind in ACPI-CA-Komponenten (`ACPI_ALL_COMPONENTS`) und ACPI-Hardware (`ACPI_ALL_DRIVERS`) aufgeteilt. Welche Meldungen ausgegeben werden, wird über “level” gesteuert. “level” reicht von `ACPI_LV_ERROR` (es werden nur Fehler ausgegeben) bis zu `ACPI_LV_VERBOSE` (alles wird ausgegeben). “level” ist eine Bitmaske, sodass verschiedene Stufen auf einmal (durch Leerzeichen getrennt) angegeben werden können. Die erzeugte Ausgabemenge passt vielleicht nicht in den Konsolenpuffer. In diesem Fall sollten Sie die Ausgaben mithilfe einer seriellen Konsole sichern. Die möglichen Werte für “layers” und “level” werden in der Hilfeseite acpi(4) beschrieben.

Die Ausgaben zur Fehlersuche sind in der Voreinstellung nicht aktiviert. Wenn ACPI im Kernel enthalten ist, fügen Sie `options ACPI_DEBUG` zur Kernelkonfigurationsdatei hinzu. Sie können die Ausgaben zur Fehlersuche global aktivieren, indem Sie in der Datei `/etc/make.conf` die Zeile `ACPI_DEBUG=1` einfügen. Das Modul `acpi.ko` können Sie wie folgt neu übersetzen:

```
# cd /sys/modules/acpi/acpi
&& make clean &&
make ACPI_DEBUG=1
```

Installieren Sie anschließend `acpi.ko` im Verzeichnis `/boot/kernel`. In der Datei `loader.conf` stellen Sie “level” und “layer” ein. Das folgende Beispiel aktiviert die Ausgabe von Fehlern für alle ACPI-CA-Komponenten und alle ACPI-Hardwaretreiber (wie CPU, LID):

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Wenn ein Problem durch ein bestimmtes Ereignis, beispielsweise den Start nach einem Ruhezustand, hervorgerufen wird, können Sie die Einstellungen für “level” und “layer” auch mit dem Kommando `sysctl` vornehmen. In diesem Fall müssen Sie die Datei `loader.conf` nicht editieren. Auf der `sysctl`-Kommandozeile geben Sie dieselben Variablennamen wie in `loader.conf` an.

12.16.7. ACPI-Informationsquellen

Weitere Informationen zu ACPI erhalten Sie an den folgenden Stellen:

- die FreeBSD ACPI (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi) Mailingliste,
- die Archive der ACPI-Mailingliste: http://lists.FreeBSD.org/pipermail/freebsd-acpi/,
- die alten Archive der ACPI-Mailingliste: http://home.jp.FreeBSD.org/mail-list/acpi-jp/,
- die ACPI-Spezifikation (Version 2.0): http://acpi.info/spec.htm,
- in den nachstehenden FreeBSD-Hilfeseiten: acpi(4), acpi_thermal(4), acpidump(8), iasl(8) und acpidb(8),
- DSDT debugging resource (http://www.cpqlinux.com/acpi-howto.html#fix_broken_dsdt) (als Beispiel wird Compaq erläutert, die Ressource ist aber dennoch nützlich).

Fußnoten

1. Der verwendete Algorithmus setzt `maxusers` auf die Speichergröße des Systems. Der minimale Wert beträgt dabei `32`, das Maximum ist `384`.

Kapitel 13. FreeBSDs Bootvorgang

Übersetzt von Hans-Christian Ebke.

13.1. Übersicht

Das Starten des Computers und das Laden des Betriebssystems wird im Allgemeinen als “Bootstrap-Vorgang” bezeichnet, oder einfach als “Booten”. FreeBSDs Bootvorgang ermöglicht große Flexibilität, was das Anpassen dessen anbelangt, was passiert, wenn das System gestartet wird. Es kann zwischen verschiedenen Betriebssystemen, die auf demselben Computer installiert sind oder verschiedenen Versionen desselben Betriebssystems oder installierten Kernels gewählt werden.

Dieses Kapitel zeigt die zur Verfügung stehenden Konfigurationsmöglichkeiten und wie man den Bootvorgang anpasst. Dies schließt alles ein, bis der Kernel gestartet worden ist, der dann alle Geräte gefunden hat und init(8) gestartet hat. Falls Sie sich nicht ganz sicher sind, wann dies passiert: Es passiert, wenn die Farbe des Textes während des Bootvorgangs von weiß zu Hellgrau wechselt.

Dieses Kapitel informiert über folgende Punkte:

- Die Komponenten des FreeBSD-Bootvorgangs und deren Interaktion.
- Die Optionen, mit denen Sie den FreeBSD-Bootvorgang steuern können.
- Wie Geräte mit device.hints(5) konfiguriert werden.

nur x86: Dieses Kapitel erklärt den Bootvorgang von FreeBSD auf Intel X86 Plattformen.

13.2. Das Problem des Bootens

Wenn der Computer eingeschaltet wird und das Betriebssystem gestartet werden soll, entsteht ein interessantes Dilemma, denn der Computer weiß per Definition nicht, wie er irgendetwas tut, bis das Betriebssystem gestartet wurde. Das schließt das Starten von Programmen, die sich auf der Festplatte befinden, ein. Wenn nun der Computer kein Programm von der Festplatte starten kann, sich das Betriebssystem aber dummerweise genau dort befindet, wie wird es dann gestartet?

Dieses Problem ähnelt einer Geschichte des Barons von Münchhausen. Dort war eine Person in einen Sumpf gefallen und hat sich selbst an den Riemen seiner Stiefel (engl. *bootstrap*) herausgezogen. In den jungen Jahren des Computerzeitalters wurde mit dem Begriff Bootstrap dann die Technik das Betriebssystem zu laden bezeichnet und wurde hinterher mit booten abgekürzt.

Auf x86-Plattformen ist das BIOS (Basic Input/Output System) dafür verantwortlich, das Betriebssystem zu laden. Dazu liest das BIOS den Master Bootsektor (MBR; Master Boot Record) aus, der sich an einer bestimmten Stelle auf der Festplatte/Diskette befinden muss. Das BIOS kann den MBR selbstständig laden und ausführen und geht davon aus, dass dieser die restlichen Dinge, die für das Laden des Betriebssystems notwendig sind, selbst oder mit Hilfe des BIOS erledigen kann.

Der Code innerhalb des MBRs wird für gewöhnlich als *Boot-Manager* bezeichnet, insbesondere, wenn eine Interaktion mit dem Anwender stattfindet. Ist dies der Fall, verwaltet der Boot-Manager zusätzlichen Code im ersten

Track der Platte oder in Dateisystemen anderer Betriebssysteme. (Boot-Manager werden manchmal auch als *Boot Loader* bezeichnet, unter FreeBSD wird dieser Begriff aber für eine spätere Phase des Systemstarts verwendet.) Zu den bekanntesten Boot-Managern gehören **boot0** (der auch als **Boot Easy** bekannte Standard-Boot-Manager von FreeBSD), **Grub**, **GAG**, sowie **LILO**. (Von diesen Boot-Managern hat nur **boot0** innerhalb des MBRs Platz.)

Falls nur ein Betriebssystem installiert ist, ist der Standard MBR ausreichend. Dieser MBR sucht nach dem ersten bootbaren Slice (das dabei als *active* gekennzeichnet ist) auf dem Laufwerk und führt den dort vorhandenen Code aus, um das restliche Betriebssystem zu laden. Der von fdisk(8) in der Voreinstellung installierte MBR ist ein solcher MBR und basiert auf `/boot/mbr`.

Falls mehrere Betriebssysteme installiert sind, sollte man einen anderen Boot-Manager installieren, der eine Liste der verfügbaren Betriebssysteme anzeigt und einen wählen lässt, welches man booten möchte. Der nächste Abschnitt beschreibt zwei Boot-Manager mit diesen Fähigkeiten.

Das restliche FreeBSD-Bootstrap-System ist in drei Phasen unterteilt. Die erste Phase wird vom MBR durchgeführt, der gerade genug Funktionalität besitzt um den Computer in einen bestimmten Status zu verhelfen und die zweite Phase zu starten. Die zweite Phase führt ein wenig mehr Operationen durch und startet schließlich die dritte Phase, die das Laden des Betriebssystems abschließt. Der ganze Prozess wird in drei Phasen durchgeführt, weil der PC Standard die Größe der Programme, die in Phase eins und zwei ausgeführt werden, limitiert. Durch das Verketten der durchzuführenden Aufgaben wird es FreeBSD möglich, ein sehr flexibles Ladeprogramm zu besitzen.

Als nächstes wird der Kernel gestartet, der zunächst nach Geräten sucht und sie für den Gebrauch initialisiert. Nach dem Booten des Kernels übergibt dieser die Kontrolle an den Benutzer Prozess init(8), der erst sicherstellt, dass alle Laufwerke benutzbar sind und die Ressourcen Konfiguration auf Benutzer Ebene startet. Diese wiederum mountet Dateisysteme, macht die Netzwerkkarten für die Kommunikation mit dem Netzwerk bereit und startet generell alle Prozesse, die auf einem FreeBSD-System normalerweise beim Hochfahren gestartet werden.

13.3. Boot-Manager und Boot-Phasen

13.3.1. Der Boot-Manager

Der Code im MBR oder im Boot-Manager wird manchmal auch als *stage zero* des Boot-Prozesses bezeichnet. Dieser Abschnitt beschreibt zwei der weiter oben erwähnten Boot-Manager: **boot0** sowie **LILO**.

Der boot0 Boot-Manager: Der vom FreeBSD-Installationsprogramm oder boot0cfg(8) in der Voreinstelung installierte Master Boot Record (MBR) basiert auf `/boot/boot0`. Bei **boot0** handelt es sich um ein sehr einfaches Programm, da im MBR lediglich 446 Bytes verfügbar sind, weil der restliche Platz für die Partitionstabelle sowie den `0x55AA`-Identifier am Ende des MBRs benötigt wird. Falls Sie **boot0** verwenden und mehrere Betriebssysteme auf Ihrer Festplatte installiert haben, werden Sie beim Starten des Computers eine Anzeige ähnlich der folgenden sehen:

Beispiel 13-1. `boot0`-Screenshot

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1

Default: F2
```

Diverse Betriebssysteme, insbesondere Windows, überschreiben den MBR ungefragt mit ihrem eigenen. Falls einem dies passiert sein sollte, kann man mit folgendem Kommando den momentanen MBR durch den FreeBSD-MBR ersetzen:

```
# fdisk -B -b /boot/boot0 Gerät
```

Bei *`Gerät`* handelt es sich um das Gerät, von dem gebootet wird, also beispielsweise `ad0` für die erste IDE-Festplatte, `ad2` für die erste IDE-Festplatte am zweiten IDE-Controller, `da0` für die erste SCSI-Festplatte, usw. Diese Einstellungen können aber über boot0cfg(8) angepasst werden.

Der LILO-Boot-Manager: Damit dieser Boot-Manager auch FreeBSD booten kann, starten Sie zuerst Linux und fügen danach folgende Zeilen in die Konfigurationsdatei `/etc/lilo.conf` ein:

```
other=/dev/hdXY
table=/dev/hdX
loader=/boot/chain.b
label=FreeBSD
```

Dabei müssen Sie die primäre Partition von FreeBSD sowie dessen Platte im Linux-Format angeben. Dazu ersetzen Sie *`X`* durch die Linux-Bezeichnung der Platte und *`Y`* durch die von Linux verwendete Partitionsnummer. Wenn Sie ein SCSI-Laufwerk verwenden, müssen Sie *`/dev/sd`* anstelle von *`/dev/hd`* verwenden. Die Zeile `loader=/boot/chain.b` kann weggelassen werden, wenn beide Betriebssysteme auf der gleichen Platte installiert sind. Geben Sie danach `/sbin/lilo -v` ein, um Ihre Änderungen zu übernehmen. Achtung Sie dabei besonders auf etwaige Fehlermeldungen.

13.3.2. Phase Eins, `/boot/boot1` und Phase Zwei, `/boot/boot2`

Im Prinzip sind die erste und die zweite Phase Teile desselben Programms, im selben Bereich auf der Festplatte. Aufgrund von Speicherplatz-Beschränkungen wurden sie aufgeteilt, aber man installiert sie eigentlich generell zusammen. Beide werden entweder vom Installer oder von **bsdlabel** aus der kombinierten Datei `/boot/boot` kopiert.

Beide Phasen befinden sich außerhalb des Dateisystems im Bootsektor des Boot-Slices, wo boot0 oder ein anderer Boot-Manager ein Programm erwarten, das den weiteren Bootvorgang durchführen kann. Die Anzahl der dabei verwendeten Sektoren wird durch die Größe von `/boot/boot` bestimmt.

`boot1` ist ein sehr einfaches Programm, da es nur 512 Bytes groß sein darf, und es besitzt gerade genug Funktionalität, um FreeBSDs *bsdlabel*, das Informationen über den Slice enthält, auszulesen, und um `boot2` zu finden und auszuführen.

`boot2` ist schon ein wenig umfangreicher und besitzt genügend Funktionalität, um Dateien in FreeBSDs Dateisystem zu finden. Außerdem hat es eine einfache Schnittstelle, die es ermöglicht, den zu ladenden Kernel oder Loader auszuwählen.

Da der Loader einen weitaus größeren Funktionsumfang hat und eine schöne und einfach zu bedienende Boot-Konfigurations-Schnittstelle zur Verfügung stellt, wird er gewöhnlich von `boot2` anstatt des Kernels gestartet. Früher war es jedoch dazu da den Kernel direkt zu starten.

Beispiel 13-2. `boot2`-Screenshot

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
```

`boot:`

Um das installierte `boot1` und `boot2` zu ersetzen, benutzt man bsdlabel(8):

```
# bsdlabel -B disklice
```

Wobei *`Slice`* das Laufwerk und die Slice darstellt, von dem gebootet wird, beispielsweise `ad0s1` für die erste Slice auf der ersten IDE-Festplatte.

> **Dangerously Dedicated Mode:** Wenn man nur den Festplatten-Namen, also z.B. `ad0`, in bsdlabel(8) benutzt wird eine "dangerously dedicated disk" erstellt, ohne Slices. Das ist ein Zustand, den man meistens nicht hervorrufen möchte. Aus diesem Grund sollte man ein bsdlabel(8)-Kommando noch einmal prüfen, bevor man **Return** betätigt.

13.3.3. Phase drei, `/boot/loader`

Der boot-loader ist der letzte von drei Schritten im Bootstrap-Prozess und kann im Dateisystem normalerweise unter `/boot/loader` gefunden werden.

Der Loader soll eine benutzerfreundliche Konfigurations-Schnittstelle sein mit einem einfach zu bedienenden eingebauten Befehlssatz, ergänzt durch einen umfangreichen Interpreter mit einem komplexeren Befehlssatz.

13.3.3.1. Loader Ablauf

Der Loader sucht während seiner Initialisierung nach Konsolen und Laufwerken, findet heraus, von welchem Laufwerk er gerade bootet, und setzt dementsprechend bestimmte Variablen. Dann wird ein Interpreter gestartet, der Befehle interaktiv oder von einem Skript empfangen kann.

Danach liest der Loader die Datei `/boot/loader.rc` aus, welche ihn standardmäßig anweist `/boot/defaults/loader.conf` zu lesen, wo sinnvolle Standardeinstellungen für diverse Variablen festgelegt werden und wiederum `/boot/loader.conf` für lokale Änderungen an diesen Variablen ausgelesen wird. Anschließend arbeitet dann `loader.rc` entsprechend dieser Variablen und lädt die ausgewählten Module und den gewünschten Kernel.

In der Voreinstellung wartet der Loader 10 Sekunden lang auf eine Tastatureingabe und bootet den Kernel, falls keine Taste betätigt wurde. Falls doch eine Taste betätigt wurde wird dem Benutzer eine Eingabeaufforderung angezeigt. Sie nimmt einen einfach zu bedienenden Befehlssatz entgegen, der es dem Benutzer erlaubt, Änderungen an Variablen vorzunehmen, Module zu laden, alle Module zu entladen oder schließlich zu booten bzw. neu zu booten.

13.3.3.2. Die eingebauten Befehle des Loaders

Hier werden nur die gebräuchlichsten Befehle bearbeitet. Für eine erschöpfende Diskussion aller verfügbaren Befehle konsultieren Sie bitte loader(8).

autoboot *`Sekunden`*

> Es wird mit dem Booten des Kernels fortgefahren, falls keine Taste in der gegebenen Zeitspanne betätigt wurde. In der gegebenen Zeitspanne, Vorgabe sind 10 Sekunden, wird ein Countdown angezeigt.

boot [*-options*] [*Kernelname*]

Bewirkt das sofortige Booten des Kernels mit den gegebenen Optionen, falls welche angegeben wurden, und mit den angegebenen Kernel, falls denn einer angegeben wurde. Das übergeben eines Kernelnamens ist nur nach einem *unload*-Befehl anwendbar, andernfalls wird der zuvor verwendete Kernel benutzt.

boot-conf

Bewirkt die automatische Konfiguration der Module, abhängig von den entsprechenden Variablen. Dieser Vorgang ist identisch zu dem Vorgang, den der Bootloader ausführt und daher nur sinnvoll, wenn zuvor `unload` benutzt wurde und Variablen (gewöhnlich `kernel`) verändert wurden.

help [*Thema*]

Zeigt die Hilfe an, die zuvor aus der Datei `/boot/loader.help` gelesen wird. Falls `index` als Thema angegeben wird, wird die Liste der zur Verfügung stehenden Hilfe-Themen angezeigt.

include *Dateiname* ...

Verarbeitet die angegebene Datei. Das Einlesen und Interpretieren geschieht Zeile für Zeile und wird im Falle eines Fehlers umgehend unterbrochen.

load [`-t` *Typ*] *Dateiname*

Lädt den Kernel, das Kernel-Modul, oder die Datei des angegebenen Typs. Optionen, die auf den Dateinamen folgen, werden der Datei übergeben.

ls [`-l`] [*Pfad*]

Listet die Dateien im angegebenen Pfad auf, oder das root-Verzeichnis(/), falls kein Pfad angegeben wurde. Die Option `-l` bewirkt, dass die Dateigrössen ebenfalls angezeigt werden.

lsdev [`-v`]

Listet alle Geräte auf, für die Module geladen werden können. Die Option `-v` bewirkt eine detailreichere Ausgabe.

lsmod [`-v`]

Listet alle geladenen Module auf. Die Option `-v` bewirkt eine detailreichere Ausgabe.

more *Dateiname*

Zeigt den Dateinhalt der angegebenen Datei an, wobei eine Pause alle `LINES` Zeilen gemacht wird.

reboot

Bewirkt einen umgehenden Neustart des Systems.

set *Variable*
set *Variable=Wert*

Setzt die Umgebungsvariablen des Loaders.

unload

Entlädt sämtliche geladenen Module.

13.3.3.3. Beispiele für die Loader Bedienung

Hier ein paar praktische Beispiele für die Bedienung des Loaders.

- Um den gewöhnlichen Kernel im Single-User Modus zu starten:

  ```
  boot -s
  ```

- Um alle gewöhnlichen Kernelmodule zu entladen und dann nur den alten (oder jeden beliebigen anderen) Kernel zu laden:

  ```
  unload
  load kernel.old
  ```

  Es kann `kernel.GENERIC` verwendet werden, um den allgemeinen Kernel zu bezeichnen, der vorinstalliert wird. `kernel.old` bezeichnet den Kernel, der vor dem aktuellen installiert war (falls man einen neuen Kernel kompiliert und installiert hat zum Beispiel).

  > **Anmerkung:** Der folgende Befehl lädt die gewöhnlichen Module mit einem anderen Kernel:
  >
  > ```
  > unload
  > set kernel="kernel.old"
  > boot-conf
  > ```

- Folgendes lädt ein Kernelkonfigurations-Skript (ein automatisiertes Skript, dass dasselbe tut, was der Benutzer normalerweise von Hand an der Eingabeaufforderung durchführen würde):

  ```
  load -t userconfig_script /boot/kernel.conf
  ```

13.3.3.4. Willkommmensbildschirme während des Bootvorgangs

Contributed by Joseph J. Barbish. Übersetzt von Benedict Reuschling.

Die Willkommmensbildschirme erzeugen einen visuell viel ansprechenderen Bootvorgang im Vergleich zu den herkömmlichen Bootmeldungen. Diese Bildschirme werden entweder bis zu einem Konsolen-Login-Prompt oder dem eines X-Display Managers angezeigt.

Es existieren zwei grundlegende Umgebungen in FreeBSD. Die erste ist die altbekannte, auf virtuellen Konsolen basierte Kommandozeile. Nachdem das System den Bootvorgang abgeschlossen hat, wird ein Anmeldebildschirm auf der Konsole anzeigt. Die zweite Umgebung ist die graphische X11-Desktop Umgebung. Nachdem X11 und eine der Graphischen Oberflächen, wie **GNOME**, **KDE**, oder **XFce** installiert wurden, kann der X11-Desktop über das Kommando `startx` gestartet werden.

Manche Benutzer ziehen den graphischen Anmeldebildschirm von X11 dem traditionellen, textbasierten Anmeldeprompt vor. Display-Manager wie **XDM** für Xorg, **gdm** für **GNOME** und **kdm** für **KDE** (und viele weitere aus der Ports-Sammlung) bieten einen graphischen statt dem konsolenbasierten Anmeldebildschirm. Nach einer erfolgreichen Anmeldung kann der Benutzer die graphische Oberfläche verwenden.

In der Kommandozeilen-Umgebung würde der Willkommensbildschirm alle Erkennungsmeldungen des Bootvorgangs und die Startmeldungen von Diensten verstecken, bevor der Anmeldebildschirm erscheint. In der X11-Umgebung erhalten die Anwender einen klareren visuellen Eindruck des Startvorgangs, ähnlich zu dem, den Microsoft Windows (oder ein nicht-Unix-artiger Systemtyp) zur Verfügung stellt.

13.3.3.4.1. Willkommensbildschirm-Funktionalität

Die Willkommensbildschirm-Funktionalität unterstützt nur 256-Farben Bitmaps (`.bmp`), ZSoft PCX (`.pcx`) oder TheDraw (`.bin`) Dateien. Zusätzlich muss die Willkommensbildschirm-Datei eine Auflösung von 320 mal 200 Pixeln oder weniger besitzen, damit Standard-VGA Geräte damit arbeiten können.

Um grössere Bilder bis zu einer maximalen Auflösung von 1024 mal 768 Pixeln zu verwenden, muss Unterstützung für VESA in FreeBSD enthalten sein. Dies kann durch das Laden des VESA-Moduls während des Systemstarts geschehen, oder durch Hinzufügen der `VESA`-Kernelkonfigurationsoption und anschliessendem Bau des Kernels (Lesen Sie dazu Kapitel 9). Die VESA-Unterstützung ermöglicht es den Benutzern, Willkommensbildschirme als Vollbild anzuzeigen, die den gesamten Bildschirm ausfüllen.

Wenn der Willkommensbildschirm beim Bootvorgang angezeigt wird, kann dieser jederzeit mit einem beliebigen Tastendruck ausgeschaltet werden.

Der Willkommensbildschirm ist standardmässig so eingestellt, dass er als Bildschirmschoner ausserhalb von X11 verwendet wird. Nach einer bestimmten Zeit der Untätigkeit wird der Willkommensbildschirm angezeigt und wechselt durch verschiedene Stufen der Intensität von hell zu einem sehr dunklen Bild und wieder zurück. Dieses Verhalten des Standard-Willkommensbildschirms (Screen-Saver) kann durch hinzufügen einer `saver=`-Zeile in `/etc/rc.conf` geändert werden. Die Option `saver=` besitzt mehrere eingebaute Screen-Saver, aus denen man wählen kann, und deren komplette Liste in der splash(4)-Manualpage enthalten ist. Der Standard-Screen-Saver ist “warp”. Beachten Sie, dass sich die `saver=`-Option in `/etc/rc.conf` nur auf virtuelle Konsolen bezieht. Sie hat keinen Effekt auf X11-Display-Manager.

Ein paar Nachrichten des Bootloaders und ganz besonders das Menü mit den Bootoptionen und dem Warte-Countdown werden zur Bootzeit angezeigt, selbst wenn der Willkommensbildschirm aktiviert ist.

Dateien mit Beispiel-Willkommensbildschirmen können von der Galerie auf http://artwork.freebsdgr.orgb (http://artwork.freebsdgr.org/node/3/) heruntergeladen werden. Durch die Installation des Ports `sysutils/bsd-splash-changer` können Willkommensbildschirme von einer zufällig ausgewählten Sammlung von Bildern bei jedem Neustart angezeigt werden.

13.3.3.4.2. Aktivieren der Willkommensbildschirm-Funktionalität

Die Willkommensbildschirm-Datei (`.bmp`, `.pcx` oder `.bin`) muss im Wurzelverzeichnis, z.B. `/boot` abgelegt werden.

Für die Standard-Auflösung (256-Farben, 320 mal 200 Pixel oder weniger) beim Booten bearbeiten Sie die Datei `/boot/loader.conf`, so dass diese die folgenden Zeilen enthält:

```
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Für grössere Video-Auflösungen bis zum Maximum von 1024 mal 768 Pixeln ändern Sie die Datei `/boot/loader.conf`, damit diese die folgenden Zeilen enthält:

```
vesa_load="YES"
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Das Beispiel oben nimmt an, dass */boot/splash.bmp* als Willkommensbildschirm verwendet wird. Wenn eine PCX-Datei verwendet werden soll, benutzen Sie die folgenden Zeilen, inklusive der `vesa_load="YES"`-Zeile, abhängig von der Auflösung.

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

In der Version 8.3 kann als weitere Option ascii-Kunst im TheDraw (https://en.wikipedia.org/wiki/TheDraw) Format verwendet werden.

```
splash_txt="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bin"
```

Wie das Beispiel oben demonstriert, ist der Dateiname nicht auf “splash” beschränkt. Es ist beliebig, so lange es den Dateityp BMP oder PCX besitzt, z.B. *splash_640x400*`.bmp` oder *blue_wave*`.pcx`.

Weitere interessante Optionen für `loader.conf` sind:

`beastie_disable="YES"`

> Diese Option verhindert die Anzeige des Menüs mit den Bootoptionen, aber der Countdown ist immer noch aktiv. Selbst wenn das Bootmenü deaktiviert ist, kann während des Countdowns eine der korrespondierenden Optionen ausgewählt werden.

`loader_logo="beastie"`

> Dies ersetzt die Standardanzeige des Wortes “FreeBSD”. Stattdessen wird wie in der Vergangenheit auf der rechten Seite des Bootmenüs das bunte Beastie-Logo angezeigt.

Für weitere Informationen lesen Sie die Manualpages splash(4), loader.conf(5) und vga(4).

13.4. Kernel Interaktion während des Bootprozesses

Wenn der Kernel einmal geladen ist, entweder durch den Loader (die Standardmethode) oder durch boot2 (den Loader umgehend), verhält sich gemäß seiner Boot-Flags, falls es welche gibt.

13.4.1. Kernel Boot-Flags

Es folgt eine Auflistung der gebräuchlichsten Boot-Flags:

`-a`

> Bewirkt, dass der Benutzer während der Kernel-Initialisierung gefragt wird, welches Gerät als Root-Dateisystem gemounted werden soll.

`-C`

> Es wird von CD-ROM gebootet.

`-c`

UserConfig, das Boot-Zeit Konfigurationsprogramm, wird gestartet.

`-s`

Bewirkt den Start des Single-User Modus.

`-v`

Zeigt mehr Informationen während des Starten des Kernels an.

Anmerkung: Andere Boot-Flags sind in der Hilfeseite boot(8) erläutert.

13.5. Konfiguration von Geräten

Beigetragen von Tom Rhodes.

Der Boot-Loader liest während des Systemstarts die Datei device.hints(5), die Variablen, auch “device hints” genannt, zur Konfiguration von Geräten enthält.

Die Variablen können auch mit Kommandos in der Phase 3 des Boot-Loaders bearbeitet werden. Neue Variablen werden mit `set` gesetzt, `unset` löscht schon definierte Variablen und `show` zeigt Variablen an. Variablen aus `/boot/device.hints` können zu diesem Zeitpunkt überschrieben werden. Die hier durchgeführten Änderungen sind nicht permanent und beim nächsten Systemstart nicht mehr gültig.

Nach dem Systemstart können alle Variablen mit kenv(1) angezeigt werden.

Pro Zeile enthält `/boot/device.hints` eine Variable. Kommentare werden, wie üblich, durch # eingeleitet. Die verwendete Syntax lautet:

```
hint.driver.unit.keyword="value"
```

Der Boot-Loader verwendet die nachstehende Syntax:

```
set hint.driver.unit.keyword=value
```

Der Gerätetreiber wird mit *`driver`*, die Nummer des Geräts mit *`unit`* angegeben. *`keyword`* ist eine Option aus der folgenden Liste:

- `at`: Gibt den Bus, auf dem sich das Gerät befindet, an.
- `port`: Die Startadresse des I/O-Bereichs.
- `irq`: Gibt die zu verwendende Unterbrechungsanforderung (IRQ) an.
- `drq`: Die Nummer des DMA Kanals.
- `maddr`: Die physikalische Speicheradresse des Geräts.
- `flags`: Setzt verschiedene gerätespezifische Optionen.
- `disabled`: Deaktiviert das Gerät, wenn der Wert auf `1` gesetzt wird.

Ein Gerätetreiber kann mehr Optionen, als die hier beschriebenen, besitzen oder benötigen. Schlagen Sie die Optionen bitte in der Online-Hilfe des Treibers nach. Weitere Informationen erhalten Sie in device.hints(5), kenv(1), loader.conf(5) und loader(8).

13.6. Init: Initialisierung der Prozess-Kontrolle

Nachdem der Kernel den Bootprozess abgeschlossen hat, übergibt er die Kontrolle an den Benutzer-Prozess init(8). Dieses Programm befindet sich in `/sbin/init`, oder dem Pfad, der durch die Variable `init_path` im `Loader` spezifiziert wird.

13.6.1. Der automatische Reboot-Vorgang

Der automatische Reboot-Vorgang stellt sicher, dass alle Dateisysteme des Systems konsistent sind. Falls dies nicht der Fall ist und die Inkonsistenz nicht durch fsck(8) behebbar ist, schaltet init(8) das System in den Single-User Modus, damit der Systemadministrator sich des Problems annehmen kann.

13.6.2. Der Single-User Modus

Das Schalten in diesen Modus kann erreicht werden durch den automatischen Reboot-Vorgang, durch das Booten mit der Option `-s` oder das Setzen der `boot_single` Variable in `Loader`.

Weiterhin kann der Single-User Modus aus dem Mehrbenutzermodus heraus durch den Befehl shutdown(8) ohne die reboot (`-r`) oder halt (`-h`) Option erreicht werden.

Falls die System-Konsole (`console`) in `/etc/ttys` auf `insecure` (dt.: unsicher) gesetzt ist, fordert das System allerdings zur Eingabe des Passworts von `root` auf, bevor es den Single-User Modus aktiviert.

Beispiel 13-3. Auf insecure gesetzte Konsole in `/etc/ttys`

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off insecure
```

> **Anmerkung:** Eine Konsole sollte auf `insecure` gesetzt sein, wenn die physikalische Sicherheit der Konsole nicht gegeben ist und sichergestellt werden soll, dass nur Personen, die das Passwort von `root` kennen, den Single-User Modus benutzen können. Es bedeutet nicht, dass die Konsole "unsicher" laufen wird. Daher sollte man `insecure` wählen, wenn man auf Sicherheit bedacht ist, nicht `secure`.

13.6.3. Mehrbenutzermodus

Stellt init(8) fest, dass das Dateisystem in Ordnung ist, oder der Benutzer den Single-User Modus beendet, schaltet das System in den Mehrbenutzermodus, in dem dann die Ressourcen Konfiguration des Systems gestartet wird.

13.6.3.1. Ressourcen Konfiguration, rc-Dateien

Das Ressourcen Konfigurationssystem (engl. *resource configuration*, rc) liest seine Standardkonfiguration von `/etc/defaults/rc.conf` und System-spezifische Details von `/etc/rc.conf`. Dann mountet es die Dateisysteme gemäß `/etc/fstab`, startet die Netzwerkdienste, diverse System Daemons und führt schließlich die Start-Skripten der lokal installierten Anwendungen aus.

Die rc(8) Handbuch Seite ist eine gute Quelle für Informationen über das Ressourcen Konfigurationssystem und ebenso über die Skripte an sich.

13.7. Der Shutdown-Vorgang

Im Falle eines regulären Herunterfahrens durch shutdown(8) führt init(8) `/etc/rc.shutdown` aus, sendet dann sämtlichen Prozessen ein `TERM` Signal und schließlich ein `KILL` Signal an alle Prozesse, die sich nicht schnell genug beendet haben.

FreeBSD-Systeme, die Energieverwaltungsfunktionen unterstützen, können Sie mit dem Kommando `shutdown -p now` ausschalten. Zum Neustart des Systems benutzen Sie `shutdown -r now`. Das Kommando shutdown(8) kann nur von `root` oder Mitgliedern der Gruppe `operator` benutzt werden. Sie können auch die Kommandos halt(8) und reboot(8) verwenden. Weitere Informationen finden Sie in den Hilfeseiten der drei Kommandos.

> **Anmerkung:** Unter FreeBSD müssen Sie die acpi(4)-Unterstützung im Kernel aktivieren oder das Modul geladen haben, damit Sie die Energieverwaltungsfunktionen benutzen können.

Kapitel 14. Benutzer und grundlegende Account-Verwaltung

Beigetragen von Neil Blakey-Milner. Übersetzt von Robert Drehmel.

14.1. Übersicht

Einen FreeBSD-Computer können mehrere Benutzer zur selben Zeit benutzen, allerdings kann immer nur einer vor der Konsole sitzen [1], über das Netzwerk können beliebig viele Benutzer angemeldet sein. Jeder Benutzer muss einen Account haben, um das System benutzen zu können.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- die verschiedenen Account-Typen von FreeBSD kennen,
- wissen, wie Accounts angelegt werden,
- wissen, wie Sie Accounts löschen,
- wie Sie Attribute eines Accounts, wie den Loginnamen oder die Login-Shell ändern,
- wissen, wie Sie Limits für einen Account setzen, um beispielsweise Ressourcen, wie Speicher oder CPU-Zeit, einzuschränken,
- wie Sie mit Gruppen die Verwaltung der Accounts vereinfachen.

Vor dem Lesen dieses Kapitels sollten Sie

- die Grundlagen von UNIX und FreeBSD (Kapitel 4) verstanden haben.

14.2. Einführung

Jeder Zugriff auf das System geschieht über Accounts und alle Prozesse werden von Benutzern gestartet, also sind Benutzer- und Account-Verwaltung von wesentlicher Bedeutung in FreeBSD-Systemen.

Mit jedem Account eines FreeBSD-Systems sind bestimmte Informationen verknüpft, die diesen Account identifizieren.

Loginnamen

> Den Loginnamen geben Sie bei der Anmeldung ein, wenn Sie dazu mit `login:` aufgefordert werden. Loginnamen müssen auf dem System eindeutig sein, das heißt auf einem System kann es nicht zwei Accounts mit demselben Loginnamen geben. In passwd(5) wird beschrieben, wie ein gültiger Loginname gebildet wird. Normalerweise sollten Sie Namen verwenden, die aus Kleinbuchstaben bestehen und bis zu acht Zeichen lang sind.

Passwort

> Mit jedem Account ist ein Passwort verknüpft. Wenn das Passwort leer ist, wird es bei der Anmeldung nicht abgefragt. Das ist allerdings nicht zu empfehlen, daher sollte jeder Account ein Passwort besitzen.

User ID (UID)

Die UID ist üblicherweise eine Zahl zwischen 0 und 65535[2], die einen Account eindeutig identifiziert. Intern verwendet FreeBSD nur die UID, Loginnamen werden zuerst in eine UID umgewandelt, mit der das System dann weiter arbeitet. Das bedeutet, dass Sie Accounts mit unterschiedlichen Loginnamen aber gleicher UID einrichten können. Vom Standpunkt des Systems handelt es sich dabei um denselben Account. In der Praxis werden Sie diese Eigenschaft des Systems wahrscheinlich nicht benutzen.

Group ID (GID)

Die GID ist üblicherweise eine Zahl zwischen 0 und 65536[2], die eine Gruppe eindeutig identifiziert. Mit Gruppen kann der Zugriff auf Ressourcen über die GID anstelle der UID geregelt werden. Einige Konfigurationsdateien werden durch diesen Mechanismus deutlich kleiner. Ein Account kann mehreren Gruppen zugehören.

Login-Klasse

Login-Klassen erweitern das Gruppenkonzept. Sie erhöhen die Flexibilität des Systems in der Handhabung der verschiedenen Accounts.

Gültigkeit von Passwörtern

Ein regelmäßiges Ändern des Passworts wird in der Voreinstellung von FreeBSD nicht erzwungen. Sie können allerdings einen Passwortwechsel nach einer gewissen Zeit auf Basis einzelner Accounts erzwingen.

Verfallszeit eines Accounts

In der Voreinstellung verfallen unter FreeBSD keine Accounts. Wenn Sie Accounts einrichten, die nur für eine bestimmte Zeit gültig sein sollen, beispielsweise Accounts für Teilnehmer eines Praktikums, können Sie angeben, wie lange der Account gültig sein soll. Nachdem die angegebene Zeitspanne verstrichen ist, kann dieser Account nicht mehr zum Anmelden verwendet werden, obwohl alle Verzeichnisse und Dateien, die diesem Account gehören, noch vorhanden sind.

vollständiger Benutzername

FreeBSD identifiziert einen Account eindeutig über den Loginnamen, der aber keine Ähnlichkeit mit dem richtigen Namen des Benutzers haben muss. Der vollständige Benutzername kann daher beim Einrichten eines Accounts angegeben werden.

Heimatverzeichnis

Das Heimatverzeichnis gibt den vollständigen Pfad zu dem Verzeichnis an, in dem sich der Benutzer nach erfolgreicher Anmeldung befindet. Es ist üblich, alle Heimatverzeichnisse unter `/home/`*`Loginname`* oder `/usr/home/`*`Loginname`* anzulegen. Im Heimatverzeichnis oder in dort angelegten Verzeichnissen werden die Dateien eines Benutzers gespeichert.

Login-Shell

Grundsätzlich ist die Schnittstelle zum System eine Shell, von denen es viele unterschiedliche gibt. Die bevorzugte Shell eines Benutzers kann seinem Account zugeordnet werden.

Es gibt drei Haupttypen von Accounts: Der Superuser, Systembenutzer und Benutzer-Accounts. Der Superuser-Account, normalerweise `root` genannt, wird benutzt, um das System ohne Beschränkungen auf Privilegien zu verwalten. Systembenutzer starten Dienste. Abschließend werden Benutzer-Accounts von echten Menschen genutzt, die sich einloggen, Mails lesen und so weiter.

14.3. Der Superuser-Account

Der Superuser-Account, normalerweise `root` genannt, ist vorkonfiguriert und erleichtert die Systemverwaltung, sollte aber nicht für alltägliche Aufgaben wie das Verschicken und Empfangen von Mails, Entdecken des Systems oder Programmierung benutzt werden.

Das ist so, da der Superuser im Gegensatz zu normalen Benutzer-Accounts ohne Beschränkungen operiert und falsche Anwendung des Superuser-Accounts in spektakulären Katastrophen resultieren kann. Benutzer-Accounts sind nicht in der Lage, das System versehentlich zu zerstören, deswegen ist es generell am besten normale Benutzer-Accounts zu verwenden, solange man nicht hauptsächlich die extra Privilegien benötigt.

Kommandos, die Sie als Superuser eingeben, sollten Sie immer doppelt und dreifach überprüfen, da ein zusätzliches Leerzeichen oder ein fehlender Buchstabe irreparablen Datenverlust bedeuten kann.

Das erste, das Sie tun sollten, nachdem Sie dieses Kapitel gelesen haben, ist einen unprivilegierten Benutzer für Ihre eigene normale Benutzung zu erstellen, wenn Sie das nicht bereits getan haben. Das trifft immer zu, egal ob Sie ein Mehrbenutzersystem oder ein System laufen haben, welches Sie alleine benutzen. Später in diesem Kapitel besprechen wir, wie man zusätzliche Accounts erstellt und wie man zwischen dem normalen Benutzer und dem Superuser wechselt.

14.4. System-Accounts

Systembenutzer starten Dienste wie DNS, Mail-Server, Web-Server und so weiter. Der Grund dafür ist die Sicherheit; wenn die Programme von dem Superuser gestartet werden, können Sie ohne Einschränkungen handeln.

Beispiele von Systembenutzern sind `daemon`, `operator`, `bind` (für den Domain Name Service) und `news` und `www`.

`nobody` ist der generische unprivilegierte Systembenutzer. Bedenken Sie aber, dass je mehr Dienste `nobody` benutzen, desto mehr Dateien und Prozesse diesem Benutzer gehören und dieser Benutzer damit umso privilegierter wird.

14.5. Benutzer-Accounts

Benutzer-Accounts sind das primäre Mittel des Zugriffs für Menschen auf das System und isolieren Benutzer und Umgebung, schützen die Benutzer davor, das System oder Daten anderer Benutzer zu beschädigen und erlauben Benutzern ihre Umgebung selbst einzurichten, ohne das sich dies auf andere auswirkt.

Jede Person, die auf Ihr System zugreift, sollte ihren eigenen Account besitzen. Das erlaubt Ihnen herauszufinden, wer was macht und hält Leute davon ab, die Einstellungen der anderen zu verändern oder Mails zu lesen, die nicht für sie bestimmt waren.

Jeder Benutzer kann sich eine eigene Umgebung mit alternativen Shells, Editoren, Tastaturbelegungen und Sprachen einrichten.

14.6. Accounts verändern

Unter UNIX gibt es verschiedene Kommandos, um Accounts zu verändern. Die gebräuchlichsten Kommandos sind unten, gefolgt von einer detaillierten Beschreibung, zusammengefasst.

Kommando	Zusammenfassung
adduser(8)	Das empfohlene Werkzeug, um neue Accounts zu erstellen.
rmuser(8)	Das empfohlene Werkzeug, um Accounts zu löschen.
chpass(1)	Ein flexibles Werkzeug, um Informationen in der Account-Datenbank zu verändern.
passwd(1)	Ein einfaches Werkzeug, um Passwörter von Accounts zu ändern.
pw(8)	Ein mächtiges und flexibles Werkzeug um alle Informationen über Accounts zu ändern.

14.6.1. `adduser`

adduser(8) ist ein einfaches Programm um neue Benutzer hinzuzufügen. Es erstellt `passwd` und `group` Einträge für den Benutzer, genauso wie ein home Verzeichnis, kopiert ein paar vorgegebene Dotfiles aus `/usr/share/skel` und kann optional dem Benutzer eine „Willkommen“-Nachricht zuschicken.

Beispiel 14-1. Einen Benutzer unter FreeBSD anlegen

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```

Anmerkung: Wenn Sie das Passwort eingeben, werden weder Passwort noch Sternchen angezeigt. Passen Sie auf, dass Sie das Passwort korrekt eingeben.

14.6.2. `rmuser`

Benutzen Sie rmuser(8), um einen Account vollständig aus dem System zu entfernen. rmuser(8) führt die folgenden Schritte durch:

1. Entfernt den crontab(1) Eintrag des Benutzers (wenn dieser existiert).
2. Entfernt alle at(1) jobs, die dem Benutzer gehören.
3. Schließt alle Prozesse des Benutzers.
4. Entfernt den Benutzer aus der lokalen Passwort-Datei des Systems.
5. Entfernt das Heimatverzeichnis des Benutzers (falls es dem Benutzer gehört).
6. Entfernt eingegangene E-Mails des Benutzers aus `/var/mail`.
7. Entfernt alle Dateien des Benutzers aus temporären Dateispeicherbereichen wie `/tmp`.
8. Entfernt den Loginnamen von allen Gruppen, zu denen er gehört, aus `/etc/group`.

Anmerkung: Wenn eine Gruppe leer wird und der Gruppenname mit dem Loginnamen identisch ist, wird die Gruppe entfernt; das ergänzt sich mit den einzelnen Benutzer-Gruppen, die von adduser(8) für jeden neuen Benutzer erstellt werden.

Der Superuser-Account kann nicht mit rmuser(8) entfernt werden, da dies in den meisten Fällen das System unbrauchbar macht.

Als Vorgabe wird ein interaktiver Modus benutzt, der sicherzustellen versucht, dass Sie wissen, was Sie tun.

Beispiel 14-2. Interaktives Löschen von Account mit `rmuser`

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Updating password file, updating databases, done.
Updating group file: trusted (removing group jru -- personal group is empty) done.
Removing user's incoming mail file /var/mail/jru: done.
Removing files belonging to jru from /tmp: done.
Removing files belonging to jru from /var/tmp: done.
Removing files belonging to jru from /var/tmp/vi.recover: done.
#
```

14.6.3. chpass

chpass(1) ändert Informationen der Benutzerdatenbank wie Passwörter, Shells und persönliche Informationen.

Nur Systemadministratoren, mit Superuser-Rechten, können die Informationen und Passwörter der anderen Benutzer mit chpass(1) verändern.

Werden keine Optionen neben dem optionalen Loginnamen angegeben, zeigt chpass(1) einen Editor mit Account-Informationen an und aktualisiert die Account-Datenbank, wenn dieser verlassen wird.

> **Anmerkung:** Unter FreeBSD wird nach dem Verlassen des Editors nach dem Passwort gefragt, es sei denn, man ist als Superuser angemeldet.

Beispiel 14-3. Interaktives `chpass` des Superusers

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Der normale Benutzer kann nur einen kleinen Teil dieser Informationen verändern und natürlich nur die Daten des eigenen Accounts.

Beispiel 14-4. Interaktives `chpass` eines normalen Benutzers

```
#Changing user database information for jru.
Shell: /usr/local/bin/tcsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

> **Anmerkung:** chfn(1) und chsh(1) sind nur Verweise auf chpass(1) genauso wie ypchpass(1), ypchfn(1) und ypchsh(1). NIS wird automatisch unterstützt, deswegen ist es nicht notwendig das `yp` vor dem Kommando einzugeben. NIS wird später in Kapitel 30 besprochen.

14.6.4. passwd

passwd(1) ist der übliche Weg, Ihr eigenes Passwort als Benutzer zu ändern oder das Passwort eines anderen Benutzers als Superuser.

> **Anmerkung:** Um unberechtigte Änderungen zu verhindern, muss bei einem Passwortwechsel zuerst das ursprüngliche Passwort eingegeben werden.

Beispiel 14-5. Wechseln des Passworts

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done

# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Beispiel 14-6. Als Superuser das Passwort eines anderen Accounts verändern

```
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

> **Anmerkung:** Wie bei chpass(1) ist yppasswd(1) nur ein Verweis auf passwd(1). NIS wird von jedem dieser Kommandos unterstützt.

14.6.5. pw

pw(8) ist ein Kommandozeilenprogramm, mit dem man Accounts und Gruppen erstellen, entfernen, verändern und anzeigen kann. Dieses Kommando dient als Schnittstelle zu den Benutzer- und Gruppendateien des Systems. pw(8) besitzt eine Reihe mächtiger Kommandozeilenschalter, die es für die Benutzung in Shell-Skripten geeignet machen, doch finden neue Benutzer die Bedienung des Kommandos komplizierter, als die der anderen hier vorgestellten Kommandos.

14.7. Benutzer einschränken

Wenn ein System von mehreren Benutzern verwendet wird, ist es vielleicht notwendig, den Gebrauch des Systems zu beschränken. FreeBSD bietet dem Systemadministrator mehrere Möglichkeiten die System-Ressourcen, die ein einzelner Benutzer verwenden kann, einzuschränken. Diese Limitierungen sind in zwei Kategorien eingeteilt: Festplattenkontingente und andere Ressourcenbeschränkungen.

Festplatten-Kontingente schränken den Plattenplatz, der einem Benutzer zur Verfügung steht, ein. Sie bieten zudem, ohne aufwändige Berechnung, einen schnellen Überblick über den verbrauchten Plattenplatz. Kontingente werden in Abschnitt 19.15 diskutiert.

Die Login-Klassen werden in `/etc/login.conf` definiert. Auf die präzisen Semantiken gehen wir hier nicht weiter ein, sie können jedoch in login.conf(5) nachgelesen werden. Es ist ausreichend zu sagen, dass jeder Benutzer einer Login-Klasse zugewiesen wird (standardmäßig `default`) und dass jede Login-Klasse mit einem Satz von Login-Fähigkeiten verbunden ist. Eine Login-Fähigkeit ist ein *`Name=Wert`* Paar, in dem *`Name`* die Fähigkeit bezeichnet und *`Wert`* ein willkürlicher Text ist, der je nach *`Name`* entsprechend verarbeitet wird. Login-Klassen und -Fähigkeiten zu definieren, ist fast schon selbsterklärend und wird auch in login.conf(5) beschrieben.

> **Anmerkung:** Das System verwendet die Datei `/etc/login.conf` normalerweise nicht direkt, sondern nur über die Datenbank `/etc/login.conf.db`, da diese eine schnellere Abfrage erlaubt. Der nachstehende Befehl erzeugt die Datenbank `/etc/login.conf.db` aus der Datei `/etc/login.conf`:
>
> ```
> # cap_mkdb /etc/login.conf
> ```

Ressourcenbeschränkungen unterscheiden sich von normalen Login-Fähigkeiten zweifach. Erstens gibt es für jede Beschränkung ein aktuelles und ein maximales Limit. Das aktuelle Limit kann vom Benutzer oder einer Anwendung beliebig bis zum maximalen Limit verändert werden. Letzteres kann der Benutzer nur heruntersetzen. Zweitens gelten die meisten Ressourcenbeschränkungen für jeden vom Benutzer gestarteten Prozess, nicht für den Benutzer selbst. Beachten Sie jedoch, dass diese Unterschiede durch das spezifische Einlesen der Limits und nicht durch das System der Login-Fähigkeiten entstehen (das heißt, Ressourcenbeschränkungen sind *keine* Login-Fähigkeiten).

Hier befinden sich die am häufigsten benutzten Ressourcenbeschränkungen (der Rest kann zusammen mit den anderen Login-Fähigkeiten in login.conf(5) gefunden werden):

`coredumpsize`

> Das Limit der Größe einer core-Datei, die von einem Programm generiert wird, unterliegt aus offensichtlichen Gründen anderen Limits der Festplattenbenutzung (zum Beispiel `filesize` oder Festplattenkontingenten). Es wird aber trotzdem oft als weniger harte Methode zur Kontrolle des Festplattenplatz-Verbrauchs verwendet: Da Benutzer die core-Dateien nicht selbst erstellen, und sie oft nicht löschen, kann sie diese Option davor retten, dass ihnen kein Festplattenspeicher mehr zur Verfügung steht, sollte ein großes Programm, wie **emacs**, abstürzen.

`cputime`

> Die maximale Rechenzeit, die ein Prozess eines Benutzers verbrauchen darf. Überschreitet der Prozess diesen Wert, wird er vom Kernel beendet.

Anmerkung: Die Rechen*zeit* wird limitiert, nicht die prozentuale Prozessorenbenutzung, wie es in einigen Feldern in top(1) und ps(1) dargestellt wird. Letzteres war zu der Zeit, als dies hier geschrieben wurde nicht möglich und würde eher nutzlos sein: Ein Compiler – ein wahrscheinlich legitimer Vorgang – kann leicht fast 100% des Prozessors in Anspruch nehmen.

`filesize`

Hiermit lässt sich die maximale Größe einer Datei bestimmen, die der Benutzer besitzen darf. Im Gegensatz zu Festplattenkontingenten ist diese Beschränkung nur für jede einzelne Datei gültig und nicht für den Platz, den alle Dateien eines Benutzers verwenden.

`maxproc`

Das ist die maximale Anzahl von Prozessen, die ein Benutzer starten darf, und beinhaltet sowohl Vordergrund- als auch Hintergrundprozesse. Natürlich darf dieser Wert nicht höher sein als das System-Limit, das in `kern.maxproc` angegeben ist. Vergessen Sie auch nicht, dass ein zu kleiner Wert den Benutzer in seiner Produktivität einschränken könnte; es ist oft nützlich, mehrfach eingeloggt zu sein, oder *Pipelines* [3] zu verwenden. Ein paar Aufgaben, wie die Kompilierung eines großen Programms, starten mehrere Prozesse (zum Beispiel make(1), cc(1) und andere).

`memorylocked`

Dieses Limit gibt an, wie viel virtueller Speicher von einem Prozess maximal im Arbeitsspeicher festgesetzt werden kann. (siehe auch mlock(2)). Ein paar systemkritische Programme, wie amd(8), verhindern damit einen Systemzusammenbruch, der auftreten könnte, wenn sie aus dem Speicher genommen werden.

`memoryuse`

Bezeichnet den maximalen Speicher, den ein Prozess benutzen darf und beinhaltet sowohl Arbeitsspeicher-, als auch Swap- Benutzung. Es ist kein allübergreifendes Limit für den Speicherverbrauch, aber ein guter Anfang.

`openfiles`

Mit diesem Limit lässt sich die maximale Anzahl der von einem Prozess des Benutzers geöffneten Dateien festlegen. In FreeBSD werden Dateien auch verwendet, um Sockets und *IPC*-Kanäle [4] darzustellen. Setzen Sie es deshalb nicht zu niedrig. Das System-Limit ist im `kern.maxfiles` sysctl(8) definiert.

`sbsize`

Dieses Limit beschränkt den Netzwerk-Speicher, und damit die mbufs, die ein Benutzer verbrauchen darf. Es stammt aus einer Antwort auf einen DoS-Angriff, bei dem viele Netzwerk-Sockets geöffnet wurden, kann aber generell dazu benutzt werden Netzwerk-Verbindungen zu beschränken.

`stacksize`

Das ist die maximale Größe, auf die der Stack eines Prozesses heranwachsen darf. Das allein ist natürlich nicht genug, um den Speicher zu beschränken, den ein Programm verwenden darf. Es sollte deshalb in Verbindung mit anderen Limits gesetzt werden.

Beim Setzen von Ressourcenbeschränkungen sind noch andere Dinge zu beachten. Nachfolgend ein paar generelle Tipps, Empfehlungen und verschiedene Kommentare.

- Von `/etc/rc` beim Hochfahren des Systems gestartete Prozesse werden der `daemon` Login-Klasse zugewiesen.
- Obwohl das mitgelieferte `/etc/login.conf` eine Quelle von vernünftigen Limits darstellt, können nur Sie, der Administrator, wissen, was für Ihr System angebracht ist. Ein Limit zu hoch anzusetzen könnte Ihr System für Missbrauch öffnen, und ein zu niedriges Limit der Produktivität einen Riegel vorschieben.
- Benutzer des X-Window Systems (X11) sollten wahrscheinlich mehr Ressourcen zugeteilt bekommen als andere Benutzer. X11 beansprucht selbst schon eine Menge Ressourcen, verleitet die Benutzer aber auch, mehrere Programme gleichzeitig laufen zu lassen.
- Bedenken Sie, dass viele Limits für einzelne Prozesse gelten und nicht für den Benutzer selbst. Setzt man zum Beispiel `openfiles` auf 50, kann jeder Prozess des Benutzers bis zu 50 Dateien öffnen. Dadurch ist die maximale Anzahl von Dateien, die von einem Benutzer geöffnet werden können, `openfiles` mal `maxproc`. Das gilt auch für den Speicherverbrauch.

Weitere Informationen über Ressourcenbeschränkungen, Login-Klassen und -Fähigkeiten enthalten die Hilfeseiten cap_mkdb(1), getrlimit(2) und login.conf(5).

14.8. Gruppen

Eine Gruppe ist einfach eine Zusammenfassung von Accounts. Gruppen werden durch den Gruppennamen und die GID (group ID) identifiziert. Der Kernel von FreeBSD (und den meisten anderen UNIX Systemen) entscheidet anhand der UID und der Gruppenmitgliedschaft eines Prozesses, ob er dem Prozess etwas erlaubt oder nicht. Im Unterschied zur UID kann ein Prozess zu einer Reihe von Gruppen gehören. Wenn jemand von der GID eines Benutzers oder Prozesses spricht, meint er damit meistens die erste Gruppe der Gruppenliste.

Die Zuordnung von Gruppennamen zur GID steht in `/etc/group`, einer Textdatei mit vier durch Doppelpunkte getrennten Feldern. Im ersten Feld steht der Gruppenname, das zweite enthält ein verschlüsseltes Passwort, das dritte gibt die GID an und das vierte besteht aus einer Komma separierten Liste der Mitglieder der Gruppe. Diese Datei kann einfach editiert werden (natürlich nur, wenn Sie dabei keine Syntaxfehler machen). Eine ausführliche Beschreibung der Syntax dieser Datei finden Sie in group(5).

Wenn Sie `/etc/group` nicht händisch editieren möchten, können Sie pw(8) zum Editieren benutzen. Das folgende Beispiel zeigt das Hinzufügen einer Gruppe mit dem Namen `teamtwo`:

Beispiel 14-7. Setzen der Mitgliederliste einer Gruppe mit pw(8)

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

Die Zahl `1100` ist die GID der Gruppe `teamtwo`. Momentan hat `teamtwo` noch keine Mitglieder und ist daher ziemlich nutzlos. Um das zu ändern, nehmen wir nun `jru` in `teamtwo` auf.

Beispiel 14-8. Ein Gruppenmitglied mit `pw` hinzufügen

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

Als Argument von `-M` geben Sie eine Komma separierte Liste von Mitgliedern an, die in die Gruppe aufgenommen werden sollen. Aus den vorherigen Abschnitten ist bekannt, dass die Passwort-Datei ebenfalls eine Gruppe für jeden Benutzer enthält. Das System teilt dem Benutzer automatisch eine Gruppe zu, die aber vom `groupshow` Kommando von pw(8) nicht angezeigt wird. Diese Information wird allerdings von id(1) und ähnlichen Werkzeugen angezeigt. Das heißt, dass pw(8) nur `/etc/group` manipuliert, es wird nicht versuchen, zusätzliche Informationen aus `/etc/passwd` zu lesen.

Beispiel 14-9. Hinzufügen eines neuen Gruppenmitglieds mittels pw(8)

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
teamtwo:*:1100:jru,db
```

Die Argumente zur Option `-m` ist eine durch Komma getrennte Liste von Benutzern, die der Gruppe hinzugefügt werden sollen. Anders als im vorherigen Beispiel werden diese Benutzer in die Gruppe aufgenommen und ersetzen nicht die Liste der bereits bestehenden Benutzer in der Gruppe.

Beispiel 14-10. Mit `id` die Gruppenzugehörigkeit bestimmen

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

Wie Sie sehen, ist `jru` Mitglied von `jru` und `teamtwo`.

Weitere Informationen entnehmen Sie bitte pw(8).

Fußnoten

1. Außer Sie verwenden, wie in Kapitel 27 besprochen, zusätzliche Terminals
2. Für UIDs und GIDs können Zahlen bis einschließlich 4294967295 verwendet werden. Allerdings können solche IDs erhebliche Probleme mit Anwendungen verursachen, die Annahmen über den Wertebereich der IDs treffen.
3. *Pipeline* = *Leitung*. Mit *Pipes* sind Verbindungen zwischen zwei Sockets in meistens zwei verschiedenen Prozessen gemeint.
4. *IPC* steht für *Interprocess Communication*.

Kapitel 15. Sicherheit

Viel von diesem Kapitel stammt aus der security(7) Manualpage von Matthew Dillon. Übersetzt von Martin Heinen.

15.1. Übersicht

Dieses Kapitel bietet eine Einführung in die Konzepte der Systemsicherheit. Neben einigen Daumenregeln werden weiterführende Themen wie S/Key, OpenSSL und Kerberos diskutiert. Die meisten der hier besprochenen Punkte treffen sowohl auf die Systemsicherheit sowie die Internetsicherheit zu. Das Internet hat aufgehört ein “friedlicher” Ort zu sein, an dem Sie nur nette Leute finden werden. Es ist unumgänglich, dass Sie Ihre Daten, Ihr geistiges Eigentum, Ihre Zeit und vieles mehr vor dem Zugriff von Hackern schützen.

FreeBSD besitzt eine Reihe von Werkzeugen und Mechanismen, um die Integrität und die Sicherheit Ihrer Systeme und Netzwerke zu gewährleisten.

Nachdem Sie dieses Kapitel durchgearbeitet haben, werden Sie:

- Grundlegende auf FreeBSD bezogene Sicherheitsaspekte kennen.
- Die verschiedenen Verschlüsselungsmechanismen von FreeBSD, wie DES oder MD5, kennen.
- Wissen, wie Sie ein Einmalpasswörter zur Authentifizierung verwenden.
- TCP-Wrapper für **inetd** einrichten können.
- Wissen, wie Sie **Kerberos5** unter FreeBSD einrichten.
- Firewalls mit IPFW erstellen können.
- Wissen, wie Sie IPsec konfigurieren und ein VPN zwischen FreeBSD/Windows Systemen einrichten,
- **OpenSSH**, FreeBSDs Implementierung von SSH, konfigurieren und benutzen können.
- **Portaudit** anwenden können, um Softwarepakete Dritter, die Sie über die Ports-Sammlung installieren, auf bekannte Sicherheitslücken hin zu überprüfen.
- Mit FreeBSD-Sicherheitshinweisen umgehen können.
- Eine Vorstellung davon haben, was Prozessüberwachung (*Process Accounting*) ist und wie Sie diese Funktion unter FreeBSD aktivieren können.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Grundlegende Konzepte von FreeBSD und dem Internet verstehen.

Dieses Buch behandelt weitere Sicherheitsthemen. Beispielsweise werden vorgeschriebene Zugriffskontrollen in Kapitel 17 und Firewalls in Kapitel 31 besprochen.

15.2. Einführung

Sicherheit ist ein Konzept, das beim Systemadministrator anfängt und aufhört. Obwohl alle BSD UNIX Mehrbenutzersysteme über Sicherheitsfunktionen verfügen, ist es wohl eine der größten Aufgaben eines Systemadministrators zusätzliche Sicherheitsmechanismen zu erstellen und zu pflegen. Maschinen sind nur so sicher

wie sie gemacht werden und Sicherheitsanforderungen stehen oft der Benutzerfreundlichkeit entgegen. Auf UNIX Systemen können sehr viele Prozesse gleichzeitig laufen und viele dieser Prozesse sind Server, das heißt von außen kann auf sie zugegriffen werden. In einer Zeit, in der die Minicomputer und Mainframes von gestern die Desktops von heute sind und Rechner immer mehr vernetzt werden, kommt der Sicherheit eine große Bedeutung zu.

Zur Systemsicherheit gehört auch die Beschäftigung mit verschiedenen Arten von Angriffen, auch solchen, die versuchen, ein System still zu legen, oder sonst unbrauchbar zu machen ohne `root` zu kompromittieren. Sicherheitsaspekte lassen sich in mehrere Kategorien unterteilen:

1. Denial-of-Service Angriffe.
2. Kompromittierte Accounts.
3. Kompromittierter `root`-Account durch zugreifbare Server.
4. Kompromittierter `root`-Account durch kompromittierte Accounts.
5. Einrichten von Hintertüren.

Ein Denial-of-Service (Verhinderung von Diensten, DoS) Angriff entzieht einer Maschine Ressourcen, die sie zur Bereitstellung von Diensten benötigt. Meist versuchen Denial-of-Service Angriffe die Dienste oder den Netzwerkstack einer Maschine zu überlasten, um so die Maschine auszuschalten oder nicht nutzbar zu machen. Einige Angriffe versuchen, Fehler im Netzwerkstack auszunutzen, und die Maschine mit einem einzigen Paket auszuschalten. Diese Art des Angriffs kann nur verhindert werden, indem der entsprechende Fehler im Kernel behoben wird. Oft können Angriffe auf Dienste durch die Angabe von Optionen verhindert werden, die die Last, die ein Dienst auf das System unter widrigen Umständen ausüben kann, begrenzt. Angriffen auf das Netzwerk ist schwerer zu begegnen. Außer durch Trennen der Internetverbindung ist zum Beispiel einem Angriff mit gefälschten Paketen nicht zu begegnen. Diese Art von Angriff wird Ihr System zwar nicht unbrauchbar machen, kann aber die Internetverbindung sättigen.

Kompromittierte Accounts kommen noch häufiger als DoS Angriffe vor. Viele Systemadministratoren lassen auf ihren Maschinen noch die Dienste **telnetd**, **rlogind**, **rshd** und **ftpd** laufen. Verbindungen zu diesen Servern werden nicht verschlüsselt. Wenn Sie eine größere Benutzerzahl auf Ihrem System haben, die sich von einem entfernten System anmelden, ist die Folge davon, dass das Passwort eines oder mehrerer Benutzer ausgespäht wurde. Ein aufmerksamer Systemadministrator wird die Logs über Anmeldungen von entfernten Systemen auf verdächtige Quelladressen, auch für erfolgreiche Anmeldungen, untersuchen.

Es ist immer davon auszugehen, dass ein Angreifer, der Zugriff auf einen Account hat, Zugang zum `root`-Account erlangt. Allerdings gibt der Zugriff auf einen Account auf einem gut gesicherten und gepflegten System nicht notwendig Zugriff auf den `root`-Account. Diese Unterscheidung ist wichtig, da ein Angreifer, der keinen Zugang zu `root` besitzt, seine Spuren nicht verwischen kann. Er kann höchstens die Dateien des betreffenden Benutzers verändern oder die Maschine stilllegen. Kompromittierte Accounts sind sehr häufig, da Benutzer meist nicht dieselben Vorsichtsmaßnahmen wie Administratoren treffen.

Es gibt viele Wege, Zugang zum `root`-Account eines Systems zu bekommen: Ein Angreifer kann das Passwort von `root` kennen, er kann einen Fehler in einem Server entdecken, der unter `root` läuft und dann über eine Netzwerkverbindung zu diesem Server einbrechen. Oder er kennt einen Fehler in einem SUID-`root` Programm, der es ihm erlaubt, `root` zu werden, wenn er einmal einen Account kompromittiert hat. Wenn ein Angreifer einen Weg gefunden hat, `root` zu werden, braucht er vielleicht keine Hintertür auf dem System installieren. Viele der heute bekannten und geschlossenen Sicherheitslöcher, die zu einem `root` Zugriff führen, verlangen vom Angreifer einen erheblichen Aufwand, um seine Spuren zu verwischen. Aus diesem Grund wird er sich wahrscheinlich entschließen, eine Hintertür (engl. *Backdoor*) zu installieren. Eine Hintertür erlaubt es dem Angreifer leicht auf den `root`-Account zuzugreifen. Einem klugen Systemadministrator erlaubt sie allerdings auch, den Einbruch zu entdecken. Wenn Sie es

einem Angreifer verwehren, Hintertüren zu installieren, kann das schädlich für Ihre Sicherheit sein, da es vielleicht verhindert, dass die Lücke, die der Angreifer für den Einbruch ausgenutzt hat, entdeckt wird.

Sicherheitsmaßnahmen sollten immer in mehreren Schichten angelegt werden. Die Schichten können wie folgt eingeteilt werden:

1. Absichern von `root` und Accounts.
2. Absichern von unter `root` laufenden Servern und SUID/SGID Programmen.
3. Absichern von Accounts.
4. Absichern der Passwort-Datei.
5. Absichern des Kernels, der Geräte und von Dateisystemen.
6. Schnelles Aufdecken von unbefugten Veränderungen des Systems.
7. Paranoia.

Die einzelnen Punkte der obigen Liste werden im nächsten Abschnitt genauer behandelt.

15.3. Absichern von FreeBSD

> **Kommandos und Protokolle:** In diesem Abschnitt werden Anwendungen **fett** gekennzeichnet, spezifische Kommandos werden in einer `Fixschrift` dargestellt und Protokolle verwenden die normale Schriftart. Diese typographische Konvention hilft, Begriffe wie ssh zu unterscheiden, die sowohl Protokoll als auch Kommando sein können.

Die folgenden Abschnitte behandeln die im letzten Abschnitt erwähnten Methoden Ihr FreeBSD-System zu sichern.

15.3.1. Absichern von `root` und Accounts

Zuallererst, kümmern Sie sich nicht um die Absicherung von Accounts, wenn Sie `root` noch nicht abgesichert haben. Auf den meisten Systemen ist `root` ein Passwort zugewiesen. Sie sollten *immer* davon ausgehen, dass dieses Passwort kompromittiert ist. Das heißt nicht, dass Sie das Passwort entfernen sollten, da es meist für den Konsolenzugriff notwendig ist. Vielmehr heißt es, dass Sie das Passwort nicht außerhalb der Konsole, auch nicht zusammen mit su(1), verwenden sollten. Stellen Sie sicher, dass Ihre PTYs in `ttys` als unsicher markiert sind und damit Anmeldungen von `root` mit `telnet` oder `rlogin` verboten sind. Wenn Sie andere Anwendungen wie **SSH** zum Anmelden benutzen, vergewissern Sie sich, dass dort ebenfalls Anmeldungen als `root` verboten sind. Für **SSH** editieren Sie `/etc/ssh/sshd_config` und überprüfen, dass `PermitRootLogin` auf `no` gesetzt ist. Beachten Sie jede Zugriffsmethode – Dienste wie FTP werden oft vergessen. Nur an der Systemkonsole sollte ein direktes Anmelden als `root` möglich sein.

Natürlich müssen Sie als Systemadministrator `root`-Zugriff erlangen können. Dieser sollte aber durch zusätzliche Passwörter geschützt sein. Ein Weg, Zugang zu `root` zu ermöglichen, ist es, berechtigte Mitarbeiter in `/etc/group` in die Gruppe `wheel` aufzunehmen. Die Personen, die Mitglieder in der Gruppe `wheel` sind, können mit `su` zu `root` wechseln. Ihre Mitarbeiter sollten niemals die Gruppe `wheel` als primäre Gruppe in `/etc/passwd` besitzen. Mitarbeiter sollten der Gruppe `staff` angehören und über `/etc/group` in `wheel` aufgenommen werden. Es sollten auch nur die Mitarbeiter, die wirklich `root` Zugriff benötigen in `wheel` aufgenommen werden. Mit anderen

Authentifizierungsmethoden müssen Sie niemanden in `wheel` aufnehmen. Wenn Sie z.B. **Kerberos** benutzen, wechseln Sie mit ksu(1) zu `root` und der Zugriff wird mit der Datei `.k5login` geregelt. Dies ist vielleicht eine bessere Lösung, da es der `wheel`-Mechanismus einem Angreifer immer noch möglich macht, den `root`-Account zu knacken, nachdem er einen Mitarbeiter-Account geknackt hat. Obwohl der `wheel`-Mechanismus besser als gar nichts ist, ist er nicht unbedingt die sicherste Lösung.

Um ein Konto komplett zu sperren, verwenden Sie den Befehl pw(8):

```
#pw lock staff
```

Danach ist es diesem Benutzer nicht mehr möglich (auch nicht mit ssh(1)), sich anzumelden.

Eine weitere Möglichkeit, bestimmte Benutzer zu sperren, ist es, das verschlüsselte Passwort durch das Zeichen "`*`" zu ersetzen. Da ein verschlüsseltes Passwort niemals diesem Zeichen entsprechen kann, kann sich der betroffene Benutzer ebenfalls nicht mehr anmelden. Beispielsweise müsste dazu das Konto

```
foobar:R9DT/Fa1/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

wie folgt abgeändert werden:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Durch diese Änderung wird der Benutzer `foobar` daran gehindert, sich auf konventionellem Wege am System anzumelden. Diese Maßnahmen greifen allerdings nicht, wenn das betroffene System auch eine Anmeldung über **Kerberos** oder ssh(1) erlaubt.

Diese Sicherheitsmechanismen setzen voraus, dass Sie sich von einer restriktiven Maschine auf einer weniger restriktiven Maschine anmelden. Wenn zum Beispiel auf Ihrem Hauptrechner alle möglichen Arten von Servern laufen, so sollten auf Ihrer Workstation keine Server laufen. Um Ihre Workstation vernünftig abzusichern, sollten auf Ihr so wenig Server wie möglich bis hin zu keinem Server laufen. Sie sollten zudem über einen Bildschirmschoner verfügen, der mit einem Passwort gesichert ist. Natürlich kann ein Angreifer, der physikalischen Zugang zu einer Maschine hat, jede Art von Sicherheitsmechanismen umgehen. Dieses Problem sollten Sie daher auch in Ihren Überlegungen berücksichtigen. Beachten Sie dabei aber, dass der Großteil der Einbrüche über das Netzwerk erfolgt und die Einbrecher keinen Zugang zu der Maschine besitzen.

Mit **Kerberos** können Sie das Passwort eines Mitarbeiters an einer Stelle ändern und alle Maschinen, auf denen der Mitarbeiter einen Account hat, beachten die Änderung sofort. Wird der Account eines Mitarbeiters einmal kompromittiert, so sollte die Fähigkeit, das Passwort mit einem Schlag auf allen Maschinen zu ändern, nicht unterschätzt werden. Mit einzelnen Passwörtern wird es schwierig, das Passwort auf N Maschinen zu ändern. Mit **Kerberos** können Sie auch Beschränkungen für Passwörter festlegen: Nicht nur das Ticket kann nach einiger Zeit ungültig werden, Sie können auch festlegen, dass ein Benutzer nach einer bestimmten Zeit, z.B. nach einem Monat, das Passwort wechseln muss.

15.3.2. Absichern von unter `root` laufenden Servern und SUID/SGID Programmen

Ein kluger Systemadministrator lässt nur die Dienste, die er wirklich braucht, laufen; nicht mehr und auch nicht weniger. Beachten Sie, dass Server von Dritten die fehleranfälligsten sind. Wenn Sie z.B. eine alte Version von **imapd** oder **popper** laufen lassen, ist das so, als würden Sie der ganzen Welt freien Zugang zu `root` geben. Lassen Sie keine Server laufen, die Sie vorher nicht genau überprüft haben. Viele Server müssen nicht unter `root` laufen, zum Beispiel können **ntalk**, **comsat** und **finger** in speziellen *Sandkästen* unter einem Benutzer laufen. Ein Sandkasten ist keine perfekte Lösung, wenn Sie nicht eine Menge Arbeit in die Konfiguration investieren, doch

bewährt sich hier das Prinzip, die Sicherheit in Schichten aufzubauen. Wenn es einem Angreifer gelingt, in einen Server, der in einem Sandkasten läuft, einzubrechen, dann muss er immer noch aus dem Sandkasten selber ausbrechen. Je mehr Schichten der Angreifer zu durchbrechen hat, desto kleiner sind seine Aussichten auf Erfolg. In der Vergangenheit wurden praktisch in jedem Server, der unter `root` läuft, Lücken gefunden, die zu einem `root` Zugriff führten. Dies betrifft selbst die grundlegenden Systemdienste. Wenn Sie eine Maschine betreiben, auf der man sich nur mit **SSH** anmelden kann, dann stellen Sie die Dienste **telnetd**, **rshd** oder **rlogind** ab!

In der Voreinstellung laufen unter FreeBSD **ntalkd**, **comsat** und **finger** nun in einem Sandkasten. Ein weiteres Programm, das in einem Sandkasten laufen sollte, ist named(8). In `/etc/defaults/rc.conf` sind die notwendigen Argumente, um **named** in einem Sandkasten laufen zu lassen, in kommentierter Form schon enthalten. Abhängig davon, ob Sie ein neues System installieren oder ein altes System aktualisieren, sind die hierfür benötigten Benutzer noch nicht installiert. Ein kluger Systemadministrator sollte immer nach Möglichkeiten suchen, Server in einem Sandkasten laufen zu lassen.

Einige Server wie **sendmail**, **popper**, **imapd** und **ftpd** werden normalerweise nicht in Sandkästen betrieben. Zu einigen Servern gibt es Alternativen, aber diese wollen Sie vielleicht wegen der zusätzlich nötigen Arbeit nicht installieren (ein weiteres Beispiel für den Widerspruch zwischen Sicherheit und Benutzerfreundlichkeit). In diesem Fall müssen Sie die Server unter `root` laufen lassen und auf die eingebauten Mechanismen vertrauen, Einbrüche zu entdecken.

Weitere potentielle Löcher, die zu einem `root`-Zugriff führen können, sind die auf dem System installierten SUID- und SGID-Programme. Die meisten dieser Programme wie **rlogin** stehen in `/bin`, `/sbin`, `/usr/bin`, oder `/usr/sbin`. Obwohl nichts 100% sicher ist, können Sie davon ausgehen, dass die SUID- und SGID-Programme des Basissystems ausreichend sicher sind. Allerdings werden ab und an in diesen Programmen Löcher gefunden. 1998 wurde in `Xlib` ein Loch gefunden, das **xterm**, der normal mit SUID installiert wird, verwundbar machte. Es ist besser auf der sicheren Seite zu sein, als sich später zu beklagen, darum wird ein kluger Systemadministrator den Zugriff auf SUID-Programme mit einer Gruppe, auf die nur Mitarbeiter zugreifen können, beschränken. SUID-Programme, die niemand benutzt, sollten mit `chmod 000` deaktiviert werden. Zum Beispiel braucht ein Server ohne Bildschirm kein **xterm** Programm. SGID-Programme sind vergleichbar gefährlich. Wenn ein Einbrecher Zugriff auf SGID-`kmem` Programm erhält, kann er vielleicht `/dev/kmem` und damit die verschlüsselte Passwortdatei lesen. Dies kompromittiert unter Umständen jeden Account, der mit einem Passwort geschützt ist. Alternativ kann ein Einbrecher, der in die Gruppe `kmem` eingebrochen ist, die Tastendrücke auf PTYs verfolgen. Dies schließt auch PTYs mit ein, auf denen sich ein Benutzer mit sicheren Methoden anmeldet. Ein Einbrecher, der Zugriff auf die `tty` Gruppe hat, kann auf fast jeden Terminal anderer Benutzer schreiben. Wenn der Benutzer einen Terminal-Emulator benutzt, der über eine Tastatur-Simulation verfügt, könnte der Angreifer Daten generieren, die den Terminal veranlassen, ein Kommando unter diesem Benutzer laufen zu lassen.

15.3.3. Absichern von Accounts

Accounts sind für gewöhnlich sehr schwierig abzusichern. Während Sie drakonische Beschränkungen für Ihre Mitarbeiter einrichten und deren Passwörter als ungültig markieren können, werden Sie das vielleicht bei den normalen Accounts nicht durchsetzen. Wenn Sie über ausreichend Macht verfügen, gelingt es Ihnen vielleicht doch, ansonsten müssen Sie diese Accounts aufmerksam überwachen. Wegen der zusätzlichen Administrationsarbeit und der nötigen technischen Unterstützung ist die Verwendung von **SSH** und **Kerberos** mit normalen Accounts erschwert, obwohl das natürlich sicherer als die Verwendung von verschlüsselten Passwörtern ist.

15.3.4. Absichern der Passwort-Datei

Der einzig sichere Weg ist, so viele Accounts wie möglich als ungültig zu markieren und **SSH** oder **Kerberos** zu benutzen, um auf sie zuzugreifen. Obwohl die Datei `/etc/spwd.db`, die die verschlüsselten Passwörter enthält, nur von `root` gelesen werden kann, mag ein Angreifer lesenden Zugriff auf diese Datei erlangen, ohne die Fähigkeit sie auch zu beschreiben.

Ihre Überwachungsskripten sollten Änderungen an der Passwort-Datei melden (siehe Überprüfen der Integrität von Dateien weiter unten).

15.3.5. Absichern des Kernels, der Geräte und von Dateisystemen

Wenn ein Angreifer `root`-Zugriff erlangt, kann er so ziemlich alles mit Ihrem System anstellen, doch sollten Sie es ihm nicht zu leicht machen. Die meisten modernen Kernel haben zum Beispiel einen Gerätetreiber, der es erlaubt, Pakete abzuhören. Unter FreeBSD wird das Gerät `bpf` genannt. Für gewöhnlich wird ein Angreifer versuchen, dieses Gerät zu nutzen, um Pakete abzuhören. Sie sollten ihm diese Gelegenheit nicht geben und auf den meisten Systemen ist das Gerät `bpf` nicht nötig.

Auch wenn Sie `bpf` nicht verwenden, müssen Sie sich immer noch um `/dev/mem` und `/dev/kmem` sorgen. Außerdem kann der Angreifer immer noch auf die rohen Geräte (*raw devices*) schreiben. Weiterhin gibt es ein Programm zum Nachladen von Modulen in den Kernel: kldload(8). Ein unternehmungslustiger Angreifer kann dies benutzen, um sein eigenes `bpf` oder ein anderes zum Abhören geeignetes Gerät in den laufenden Kernel einzubringen. Um dieses Problem zu vermeiden, müssen Sie den Kernel auf einem höheren Sicherheitslevel laufen lassen, mindestens auf securelevel 1.

Das Securelevel des Kernels kann auf verschiedene Wege gesetzt werden. Der einfachste Weg ist das erhöhen des Securelevel des laufenden Kernels durch ein `sysctl` der `kern.securelevel` Kernel Variablen:

```
# sysctl kern.securelevel=1
```

Standardmässig bootet der FreeBSD Kernel mit einem Securelevel von -1. Der Securelevel wird solange bei -1 bleiben, bis er entweder durch den Administrator oder von init(8) durch einen Eintrag im Startup Script verändert wird. Der Securelevel kann während des Systemstarts durch das Setzen der Variable `kern_securelevel_enable` auf `YES` und der Wert der Variable `kern_securelevel` auf den gewünschten Securelevel in der `/etc/rc.conf` erhöht werden.

Der Standard Securelevel von einem FreeBSD-System direkt nach dem Start ist -1. Dies wird “insecure mode” genannt, da zum Beispiel unveränderliche Dateiflags abgeschaltet werden könnten, von allen Geräten gelesen und auf alle geschrieben werden kann.

Sobald der Securelevel auf den Wert 1 oder höher gesetzt ist, werden die append-only und die unveränderlichen Dateien geschützt, die Flags können nicht abgeschaltet werden und der Zugriff auf raw Devices ist verboten. Höhere Levels verbieten mehr Aktionen. Für einen vollständige Liste aller Securelevels, lesen Sie bitte die security(7) Manual Seite (oder die Manual Seite von init(8) für ältere Releases als FreeBSD 7.0).

> **Anmerkung:** Das Erhöhen des Securelevels auf 1 oder höher kann einige Probleme mit X11 verursachen (Zugriff auf `/dev/io` wird geblockt), ebenso die Installation von FreeBSD aus den Quellen (der `installworld` Teil muss zeitweilig die append-only und die unveränderlichen Flags einiger Dateien zurücksetzen), und auch noch in einigen anderen Fällen. Manchmal kann es, wie bei X11, durch das sehr frühe Starten von xdm(1) im Boot Prozess möglich sein, dies zu umgehen, wenn der Securelevel noch niedrig genug ist. Workarounds wie dieser sind nicht f¨r alle Securelevels und für alle Einschränkungen, die sie schaffen, möglich. Ein bisschen Vorausplanung ist eine gute Idee. Das Verständnis für die Beschränkungen, die durch jedes Securelevel

verursacht werden, ist wichtig, da sie die einfache Benutzung des Systems verschlechtern. Es vereinfacht auch die Wahl einer Standardeinstellung und schützt vor Überraschungen.

Wenn das Securelevel des Kernel auf einen Wert von 1 oder höher gesetzt ist, kann es sinnvoll sein das `schg` Flag auf kritische Startdateien, Verzeichnisse und Scripte (z.B. alles was läuft bis zu dem Punkt auf dem das Securelevel gesetzt ist) zu setzen. Dies könnte etwas übertrieben sein, und auch das Upgrade des Systems ist sehr viel schwerer, wenn es auf einem hohen Securelevel läuft. Ein strengerer Kompromiss ist es, das System auf einem höheren Securelevel laufen zu lassen, aber keine `schg` Flags für alle Systemdateien und Verzeichnisse zu setzen. Eine andere Möglichkeit ist es, einfach die Verzeichnisse `/` und `/usr` read-only zu mounten. Es sei darauf hingewiesen, dass Sie nicht vor lauter Überlegen das Wichtigste, nämlich die Entdeckung eines Eindringens, vergessen.

15.3.6. Überprüfen der Integrität von Dateien

Sie können die Systemkonfiguration und die Dateien nur so weit schützen, wie es die Benutzbarkeit des Systems nicht einschränkt. Wenn Sie zum Beispiel mit `chflags` die Option `schg` auf die meisten Dateien in `/` und `/usr` setzen, kann das Ihre Arbeit mehr behindern als nützen. Die Maßnahme schützt zwar die Dateien, schließt aber auch eine Möglichkeit, Veränderungen zu entdecken, aus. Die letzte Schicht des Sicherheitsmodells – das Aufdecken von Einbrüchen – ist sicherlich die wichtigste. Alle Sicherheitsmaßnahmen sind nichts wert, oder wiegen Sie in falscher Sicherheit, wenn Sie nicht in der Lage sind, einen möglichen Einbruch zu entdecken. Die Hälfte der Sicherheitsmaßnahmen hat die Aufgabe, einen Einbruch zu verlangsamen, um es zu ermöglichen, den Einbrecher auf frischer Tat zu ertappen.

Der beste Weg, einen Einbruch zu entdecken, ist es, nach veränderten, fehlenden oder unerwarteten Dateien zu suchen. Der wiederum beste Weg, nach veränderten Dateien zu suchen, ist es, die Suche von einem anderen (oft zentralen) besonders geschützten System durchzuführen. Es ist wichtig, dass Ihre Sicherheitsüberprüfungen vor einem Angreifer verborgen bleiben und daher sind sie auf einem besonders geschützten System gut aufgehoben. Um dies optimal auszunutzen, müssen Sie dem besonders geschützten System Zugriffsrechte auf die zu schützenden Systeme geben. Sie können die Dateisysteme der zu schützenden Systeme schreibgeschützt für das besonders geschützte System exportieren, oder Sie können der besonders geschützten Maschine **SSH** auf die anderen Maschinen erlauben, indem Sie **SSH**-Schlüsselpaare installieren. Mit Ausnahme des verursachten Netzwerkverkehrs ist die NFS-Methode die am wenigsten sichtbare. Sie erlaubt es Ihnen, nahezu unentdeckt die Dateisysteme der Clients zu beobachten. Wenn Ihr besonders geschütztes System mit den Clients über einen Switch verbunden ist, ist die NFS-Methode oft das Mittel der Wahl. Wenn das besonders geschützte System allerdings mit einem Hub verbunden ist, oder der Zugriff über mehrere Router geschieht, ist die NFS-Methode aus der Netzwerksicht zu unsicher. In einem solchen Fall ist **SSH** besser geeignet, auch wenn es deutliche Spuren hinterlässt.

Wenn das besonders geschützte System lesenden Zugriff auf die Clients hat, müssen Sie Skripten schreiben, die die Überwachung durchführen. Wenn Sie die NFS-Methode verwenden, können Sie dazu einfache Systemwerkzeuge wie find(1) und md5(1) benutzen. Am besten berechnen Sie einmal am Tag MD5-Prüfsummen der Dateien, Konfigurationsdateien in `/etc` und `/usr/local/etc` sollten öfter überprüft werden. Wenn Unstimmigkeiten zwischen den auf der besonders geschützten Maschine gehaltenen MD5-Prüfsummen und den ermittelten Prüfsummen festgestellt werden, sollte Ihr System einen Systemadministrator benachrichtigen, der den Unstimmigkeiten dann nachgehen sollte. Ein gutes Skript überprüft das System auch auf verdächtige SUID-Programme sowie gelöschte oder neue Dateien in `/` und `/usr`.

Wenn Sie **SSH** anstelle von NFS benutzen, wird das Erstellen der Skripten schwieriger. Sie müssen die Skripten und die Programme wie `find` mit `scp` auf den Client kopieren. Damit machen Sie die Überprüfung für einen Angreifer sichtbar. Außerdem kann der SSH-Client auf dem Zielsystem schon kompromittiert sein. Zusammenfassend kann

der Einsatz von **SSH** nötig sein, wenn Sie über ungesicherte Verbindungen arbeiten, aber der Umgang mit dieser Methode ist auch sehr viel schwieriger.

Ein gutes Sicherheitsskript wird auch Dateien von Benutzern, die den Zugriff auf ein System ermöglichen, wie `.rhosts`, `.shosts`, `.ssh/authorized_keys` usw., auf Veränderungen untersuchen, die über die Möglichkeiten einer Überprüfung mit `MD5` (die ja nur Veränderungen erkennen kann) hinausgehen.

Wenn Sie über große Partitionen verfügen, kann es zu lange dauern, jede Datei zu überprüfen. In diesem Fall sollten Sie beim Einhängen des Dateisystems Optionen setzen, die das Ausführen von SUID-Programmen verbieten. mount(8) stellt dazu `nosuid` zur Verfügung. Sie sollten diese Dateien aber trotzdem mindestens einmal die Woche überprüfen, da das Ziel dieser Schicht das Aufdecken eines Einbruchs, auch wenn er nicht erfolgreich war, ist.

Die Prozessüberwachung (siehe accton(8)) des Betriebssystems steht ein günstiges Werkzeug zur Verfügung, dass sich bei der Analyse eines Einbruchs als nützlich erweisen kann. Insbesondere können Sie damit herausfinden, wie der Einbrecher in das System eingedrungen ist, vorausgesetzt die Dateien der Prozessüberwachung sind noch alle intakt.

Schließlich sollten die Sicherheitsskripten die Logdateien analysieren. Dies sollte so sicher wie möglich durchgeführt werden, nützlich ist das Schreiben von Logdateien auf entfernte Systeme mit `syslog`. Ein Einbrecher wird versuchen, seine Spuren zu verwischen. Die Logdateien sind wichtig für den Systemadministrator, da er aus ihnen den Zeitpunkt und die Art des Einbruchs bestimmen kann. Eine Möglichkeit, die Logdateien unverändert aufzuheben, ist es, die Systemkonsole auf einen seriellen Port zu legen und die Informationen dort von einer gesicherten Maschine auszulesen.

15.3.7. Paranoia

Es schadet nicht, ein bisschen paranoid zu sein. Grundsätzlich darf ein Systemadministrator jede Sicherheitsmaßnahme treffen, die die Bedienbarkeit des Systems nicht einschränkt. Er kann auch Maßnahmen treffen, die die Bedienbarkeit einschränken, wenn er diese vorher genau durchdacht hat. Was noch wichtiger ist: Halten Sie sich nicht sklavisch an dieses Dokument, sondern führen Sie eigene Maßnahmen ein, um nicht einem künftigen Angreifer, der auch Zugriff auf dieses Dokument hat, alle Ihre Methoden zu verraten.

15.3.8. Denial-of-Service Angriffe

Dieser Abschnitt behandelt Denial-of-Service Angriffe (DoS). Ein DoS-Angriff findet typischerweise auf der Paketebene statt. Während Sie nicht viel gegen moderne Angriffe mit falschen Paketen, die das Netzwerk sättigen, ausrichten können, können Sie sehr wohl den Schaden begrenzen, den solche Angriffe verursachen können und insbesondere einen kompletten Serverausfall verhindern, indem Sie beispielsweise folgende Vorkehrungen treffen:

1. Begrenzen von `fork()` Aufrufen.
2. Begrenzen von Sprungbrett-Angriffen (ICMP response Angriffen, **ping** zu Broadcast-Adressen usw.).
3. Kernel-Cache für Routen.

Ein häufiger DoS-Angriff gegen forkende Server versucht den Server dazu zu bringen, solange neue Prozesse zu starten, bis das System den ganzen Speicher und alle Dateideskriptoren verbraucht hat, was dann zu einem Ausfall des Servers führt. inetd(8) besitzt einige Optionen, um diese Art von Angriffen zu begrenzen. Beachten Sie bitte, dass es möglich ist, einen Ausfall einer Maschine zu verhindern, doch ist es generell nicht möglich, den Ausfall eines Dienstes bei dieser Art von Angriffen zu verhindern. Lesen Sie sich bitte die Manualpages von **inetd** gut durch und achten Sie speziell auf die Optionen `-c`, `-C` und `-R`. Angriffe mit gefälschten IP-Adressen umgehen `-C`, so dass

normalerweise eine Kombination der Optionen benutzt werden muss. Manche Server, die nicht von **inetd** gestartet werden, besitzen Optionen, um den Start über `fork()` einzuschränken.

Sendmail besitzt die Option `-OMaxDaemonChildren`, die besser als die eingebauten Optionen zur Begrenzung der Systemauslastung funktioniert. Sie sollten beim Start von **sendmail** `MaxDaemonChildren` so hoch setzen, dass Sie die erwartete Auslastung gut abfangen können. Allerdings sollten Sie den Wert nicht so hoch setzen, dass der Rechner über seine eigenen Füße fällt. Es ist auch klug, **Sendmail** im Queue-Modus (`-ODeliveryMode=queued`) laufen zu lassen. Der Dæmon (`sendmail -bd`) sollte getrennt von den Queue-Läufen (`sendmail -q15m`) laufen. Wenn Sie trotzdem eine sofortige Auslieferung der Post wünschen, können Sie die Queue in einem geringeren Intervall, etwa `-q1m`, abarbeiten. Geben Sie für *dieses* **Sendmail** aber einen vernünftigen Wert für `MaxDaemonChildren` an, um Fehler zu verhindern.

Syslogd kann direkt angegriffen werden. Daher empfehlen wir Ihnen unbedingt die Option `-s` zu benutzen. Sollte das nicht möglich sein, benutzen Sie bitte `-a`.

Vorsicht ist auch mit Diensten geboten, die automatisch eine Rückverbindung eröffnen, wie der reverse-identd der **TCP-Wrapper**. Diese Funktion der **TCP-Wrapper** sollten Sie normalerweise nicht benutzen.

Es empfiehlt sich sehr, interne Dienste vor externen Zugriffen durch eine Firewall an der Grenze Ihres Netzwerks zu schützen. Dahinter steckt mehr die Idee, das Netzwerk vor Überlastung durch Angriffe von außen zu schützen, als interne Dienste vor einem `root`-Zugriff aus dem Netz zu schützen. Konfigurieren Sie immer eine Firewall, die alle Zugriffe blockiert, das heißt blockieren Sie *alles* außer den Ports A, B, C, D und M-Z. Damit können Sie Zugriffe auf alle niedrigen Ports blockieren und Zugriffe auf spezielle Dienste wie **named**, wenn Sie den primären Namensdienst für eine Zone anbieten, **ntalkd** oder **sendmail** erlauben. Wenn Sie die Firewall so konfigurieren, das sie in der Voreinstellung alle Zugriffe erlaubt, ist es sehr wahrscheinlich, dass Sie vergessen, eine Reihe von Diensten zu blockieren bzw. einen internen Dienst einführen und dann vergessen die Firewall zu aktualisieren. Sie können immer die höheren Portnummern öffnen, ohne die niedrigen Portnummern, die nur von `root` benutzt werden dürfen, zu kompromittieren. Beachten Sie bitte auch, dass es FreeBSD erlaubt, die Portnummern, die für dynamische Verbindungen zur Verfügung stehen, zu konfigurieren. Mit `sysctl` lassen sich verschiedene Bereiche der `net.inet.ip.portrange` Variablen setzen (eine Liste erhalten Sie mit `sysctl -a | fgrep portrange`). So können Sie zum Beispiel die Portnummern 4000 bis 5000 für den normalen Bereich und die Nummern 49152 bis 65535 für den hohen Bereich vorsehen. Dies erleichtert Ihnen die Konfiguration der Firewall, da Sie nun Zugriffe auf Ports unterhalb von 4000, mit Ausnahme der Dienste, die von außen erreichbar sein sollen, blockieren können.

Eine andere Form eines DoS-Angriffs nutzt einen Server als Sprungbrett, der Server wird dabei so angegriffen, dass seine Antworten ihn selber, das lokale Netzwerk oder einen anderen Server überlasten. Der am häufigsten verwendete Angriff dieser Art ist der *ICMP ping broadcast Angriff* . Der Angreifer fälscht dazu **ping**-Pakete, die zu der Broadcast-Adresse Ihres LANs gesendet werden, indem er darin als Quelladresse die Adresse des Opfers einsetzt. Wenn die Router an der Grenze Ihres Netzwerks **ping**-Pakete auf Broadcast-Adressen nicht abwehren, wird Ihr LAN genügend Netzwerkverkehr generieren, um das Ziel des Angriffs zu überlasten. Dies kann besonders effektiv sein, wenn der Angreifer diese Methode mit mehreren Dutzend Broadcast-Adressen über mehrere Netzwerke einsetzt. Es wurden schon Broadcast-Angriffe mit über 120 Megabit pro Sekunde gemessen. Ein zweiter Sprungbrett-Angriff wird gegen das Fehlerbehandlungssystem von ICMP eingesetzt. Indem ein Angreifer Pakete konstruiert, die eine ICMP-Fehlermeldung hervorrufen, kann er das einkommende Netzwerk des Servers sättigen und diesen wiederum veranlassen sein ausgehendes Netzwerk mit ICMP-Antworten zu sättigen. Diese Art des Angriffs kann den kompletten Speicher des Servers aufbrauchen und damit den Server stilllegen, insbesondere wenn der Server nicht in der Lage ist, die generierten ICMP-Antworten schnell genug abzuführen. Verwenden Sie die **sysctl**-Variable `net.inet.icmp.icmplim`, um die Auswirkungen solcher Angriffe zu begrenzen. Die letzte weit verbreitete Form von Sprungbrett-Angriffen verwendet interne **inetd**-Dienste wie den UDP **echo**-Dienst. Der Angreifer fälscht dazu einfach ein UDP-Paket, indem er als Quellport den **echo**-Port von Server A und als Zielport den **echo**-Port von Server B angibt, wobei beide Server in Ihrem LAN stehen. Die beiden Server werden nun dieses

Paket zwischen sich hin und her schicken. Der Angreifer kann die beiden Server und das LAN einfach damit überlasten, dass er mehrere Pakete dieser Art generiert. Ähnliche Probleme gibt es mit dem internen **chargen**-Port, daher sollten Sie die internen **inetd**-Testdienste abstellen.

Gefälschte IP-Pakete können dazu benutzt werden, den Kernel-Cache für Routen zu überlasten. Schauen Sie sich bitte die `sysctl`-Parameter `net.inet.ip.rtexpire`, `rtminexpire` und `rtmaxcache` an. Ein Angriff der gefälschte Pakete mit zufälligen Quelladressen einsetzt, bewirkt, dass der Kernel eine Route im Route-Cache anlegt, die Sie sich mit `netstat -rna | fgrep W3` ansehen können. Diese Routen verfallen für gewöhnlich nach 1600 Sekunden. Wenn der Kernel feststellt, dass die Routingtabelle im Cache zu groß geworden ist, wird er dynamisch den Wert von `rtexpire` verringern. Dieser Wert wird aber nie kleiner werden als `rtminexpire`. Daraus ergeben sich zwei Probleme:

1. Der Kernel reagiert nicht schnell genug, wenn ein Server mit einer niedrigen Grundlast plötzlich angegriffen wird.
2. `rtminexpire` ist nicht klein genug, um einen anhaltenden Angriff zu überstehen.

Wenn Ihre Server über eine T3 oder eine noch schnellere Leitung mit dem Internet verbunden sind, ist es klug, mit sysctl(8) die Werte für `rtexpire` und `rtminexpire` händisch zu setzen. Setzen Sie bitte keinen der Werte auf Null, außer Sie wollen die Maschine zum Erliegen bringen. Ein Wert von 2 Sekunden für beide Parameter sollte ausreichen, um die Routingtabelle vor einem Angriff zu schützen.

15.3.9. Anmerkungen zum Zugriff mit Kerberos und SSH

Es gibt ein paar Punkte, die Sie beachten sollten, wenn Sie **Kerberos** oder **SSH** einsetzen wollen. **Kerberos** 5 ist ein ausgezeichnetes Authentifizierungsprotokoll. Leider gibt es Fehler in den für **Kerberos** angepassten Versionen von **telnet** und **rlogin**, die sie ungeeignet für den Umgang mit binären Datenströmen machen. Weiterhin verschlüsselt **Kerberos** Ihre Sitzung nicht, wenn Sie nicht die `-x` Option verwenden, mit **SSH** wird dagegen alles verschlüsselt.

Ein Problem mit SSH sind Weiterleitungen von Verbindungen. Wenn Sie von einer sicheren Maschine, auf der sich Ihre Schlüssel befinden, eine Verbindung zu einer ungesicherten Maschine aufmachen, wird für die Dauer der Sitzung ein Port für Weiterleitungen geöffnet. Ein Angreifer, der auf der unsicheren Maschine Zugang zu `root` hat, kann diesen Port benutzen, um Zugriff auf andere Maschinen zu erlangen, die mit Ihren Schlüsseln zugänglich sind.

Wir empfehlen Ihnen, für die Logins Ihrer Mitarbeiter immer **SSH** zusammen mit **Kerberos** einzusetzen. Damit reduzieren Sie die Abhängigkeit von potentiell gefährdeten Schlüsseln und schützen gleichzeitig die Passwörter mit **Kerberos**. **SSH**-Schlüsselpaare sollten nur für automatisierte Aufgaben von einem besonders gesicherten Server eingesetzt werden (**Kerberos** kann für diese Art von Aufgaben nicht eingesetzt werden). Weiterhin empfehlen wir Ihnen, das Weiterreichen von Schlüsseln in der **SSH**-Konfiguration abzustellen bzw. die `from=IP/DOMAIN` Option in `authorized_keys` zu verwenden, die den Schlüssel nur von bestimmten Maschinen aus nutzbar macht.

15.4. DES, Blowfish, MD5, und Crypt

Teile umgeschrieben und aktualisiert von Bill Swingle.

Jedem Benutzer eines UNIX Systems ist ein Passwort zugeordnet. Es scheint offensichtlich, dass das Passwort nur dem Benutzer und dem System bekannt sein muss. Um die Passwörter geheim zu halten, werden sie mit einer nicht umkehrbaren Hash-Funktion verschlüsselt, das heißt sie können leicht verschlüsselt aber nicht entschlüsselt werden. Was wir gerade als offensichtlich dargestellt haben, ist also nicht wahr: Das Betriebssystem kennt das Passwort

wirklich nicht, es kennt nur das *verschlüsselte* Passwort. Die einzige Möglichkeit, das originale Passwort herauszufinden, besteht darin, alle möglichen Passwörter auszuprobieren (*brute force* Suche).

Zu der Zeit als UNIX entstanden ist, war die einzig sichere Möglichkeit Passwörter zu verschlüsseln, leider DES (Data Encryption Standard). Für die Einwohner der USA stellte das kein Problem dar, aber da der Quellcode von DES nicht aus den USA exportiert werden durfte, musste ein Weg gefunden werden, der die Gesetze der USA nicht verletzte und gleichzeitig die Kompatibilität mit anderen UNIX Systemen, die immer noch DES benutzten, wahrte.

Die Lösung bestand darin, die Verschlüsselungsbibliotheken aufzuspalten. Benutzer in den USA konnten die DES-Bibliotheken installieren und nutzen. In der Grundeinstellung benutzt FreeBSD MD5 als Verschlüsselungsmethode, das exportiert werden durfte und damit von jedem genutzt werden konnte. Es wird davon ausgegangen, dass MD5 sicherer als DES ist, so dass DES nur aus Kompatibilitätsgründen installiert werden sollte.

15.4.1. Erkennen der Verschlüsselungsmethode

Derzeit werden DES-, MD5- und Blowfish-Hash-Funktionen unterstützt. In der Voreinstellung benutzt FreeBSD die MD5-Hash-Funktion.

Sie können leicht herausfinden, welche Verschlüsselungsmethode von FreeBSD verwendet wird. Ein Weg besteht darin, die verschlüsselten Passwörter in `/etc/master.passwd` zu untersuchen. Passwörter, die mit MD5 verschlüsselt wurden, sind länger als die mit DES verschlüsselten und beginnen mit den Zeichen `$1$`. Passwörter, die mit `$2a$` anfangen, wurden mit der Blowfish-Funktion verschlüsselt. DES Passwörter besitzen keine offensichtlichen Merkmale, an denen sie identifiziert werden könnten. Sie sind aber kürzer als MD5-Passwörter und sind in einem 64 Zeichen umfassenden Alphabet kodiert, das das `$`-Zeichen nicht enthält. Ein relativ kurzes Passwort, das nicht mit einem `$`-Zeichen anfängt, ist wahrscheinlich ein DES-Passwort.

Die Verschlüsselungsmethode für neue Passwörter wird durch `passwd_format` in `/etc/login.conf` bestimmt. Der Wert dieser Variablen kann entweder `des`, `md5` oder `blf` sein. Näheres schlagen Sie bitte in login.conf(5) nach.

15.5. Einmalpasswörter

In der Voreinstellung unterstützt FreeBSD OPIE (*One-time Passwords in Everything*), das in der Regel MD5-Hash-Funktionen einsetzt.

Im Folgenden werden drei verschiedene Passwörter verwendet. Das erste ist Ihr normales System- oder Kerberos-Passwort und wird im Folgenden “System-Passwort” genannt. Das zweite ist das Einmalpasswort, das bei OPIE von `opiekey` generiert und von `opiepasswd` und dem Login-Programm akzeptiert wird. Im Folgenden wird es “Einmalpasswort” genannt. Das dritte Passwort ist das geheime Passwort, das Sie mit `opiekey` (manchmal auch mit `opiepasswd`) zum Erstellen der Einmalpasswörter verwenden. Dieses Passwort werden wir im Folgenden “geheimes Passwort” oder schlicht “Passwort” nennen.

Das geheime Passwort steht in keiner Beziehung zu Ihrem System-Passwort, beide können gleich sein, obwohl das nicht empfohlen wird. Die geheimen Passwörter von OPIE sind nicht auf eine Länge von 8 Zeichen, wie alte UNIX Passwörter[1], beschränkt. Sie können so lang sein, wie Sie wollen. Gebräuchlich sind Passwörter, die sich aus sechs bis sieben Wörtern zusammensetzen. Das OPIE-System arbeitet größtenteils unabhängig von den auf UNIX-Systemen verwendeten Passwort-Mechanismen.

Neben dem Passwort gibt es noch zwei Werte, die für OPIE wichtig sind. Der erste ist der “Initialwert” (engl. *seed* oder *key*), der aus zwei Buchstaben und fünf Ziffern besteht. Der zweite Wert ist der “Iterationszähler”, eine Zahl zwischen 1 und 100. OPIE generiert das Einmalpasswort, indem es den Initialwert und das geheime Passwort

aneinander hängt und dann die MD5-Hash-Funktion so oft, wie durch den Iterationszähler gegeben, anwendet. Das Ergebnis wird in sechs englische Wörter umgewandelt, die Ihr Einmalpasswort sind. Das Authentifizierungssystem (meistens PAM) merkt sich das zuletzt benutzte Einmalpasswort und Sie sind authentisiert, wenn die Hash-Funktion des Passworts dem vorigen Passwort entspricht. Da nicht umkehrbare Hash-Funktionen benutzt werden, ist es unmöglich, aus einem bekannten Passwort weitere gültige Einmalpasswörter zu berechnen. Der Iterationszähler wird nach jeder erfolgreichen Anmeldung um eins verringert und stellt so die Synchronisation zwischen Benutzer und Login-Programm sicher. Wenn der Iterationszähler den Wert 1 erreicht, muss OPIE neu initialisiert werden.

In jedem System werden mehrere Programme verwendet, die weiter unten beschrieben werden. `opiekey` verlangt einen Iterationszähler, einen Initialwert und ein geheimes Passwort. Daraus generiert es ein Einmalpasswort oder eine Liste von Einmalpasswörtern. `opiepasswd` wird dazu benutzt, um OPIE zu initialisieren. Mit diesem Programm können Passwörter, Iterationszähler oder Initialwerte geändert werden. Als Parameter verlangt es entweder ein geheimes Passwort oder einen Iterationszähler oder einen Initialwert und ein Einmalpasswort. `opieinfo` hingegen gibt den momentanen Iterationszähler und Initialwert eines Benutzers aus. Diese werden aus der Datei `/etc/opiekeys` ermittelt.

Im Folgenden werden vier verschiedene Tätigkeiten beschrieben. Zuerst wird erläutert, wie `opiepasswd` über eine gesicherte Verbindung eingesetzt werden, um Einmalpasswörter das erste Mal zu konfigurieren oder das Passwort oder den Initialwert zu ändern. Als nächstes wird erklärt, wie `opiepasswd` über eine nicht gesicherte Verbindung, oder zusammen mit `opiekey` über eine gesicherte Verbindung eingesetzt werden, um dasselbe zu erreichen. Als drittes wird beschrieben, wie `opiekey` genutzt wird, um sich über eine nicht gesicherte Verbindung anzumelden. Die vierte Tätigkeit beschreibt, wie mit `opiekey` eine Reihe von Schlüsseln generiert wird, die Sie sich aufschreiben oder ausdrucken können, um sich von Orten anzumelden, die über keine gesicherten Verbindungen verfügen.

15.5.1. Einrichten über eine gesicherte Verbindung

Um OPIE erstmals zu initalisieren, rufen Sie `opiepasswd` auf:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:

ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

Nach der Aufforderung `Enter new secret pass phrase:` oder `Enter secret password:` geben Sie bitte Ihr Passwort ein. Dies ist nicht das Passwort, mit dem Sie sich anmelden, sondern es wird genutzt, um das Einmalpasswort zu generieren. Die Zeile, die mit “ID” anfängt, enthält Ihren Login-Namen, den Iterationszähler und den Initialwert. Diese Werte müssen Sie sich nicht behalten, da das System sie zeigen wird, wenn Sie sich anmelden. In der letzten Zeile steht das Einmalpasswort, das aus diesen Parametern und Ihrem geheimen Passwort ermittelt wurde. Wenn sie sich jetzt wieder anmelden wollten, dann müssten Sie dieses Passwort benutzen.

15.5.2. Einrichten über eine nicht gesicherte Verbindung

Um Einmalpasswörter über eine nicht gesicherte Verbindung einzurichten, oder das geheime Passwort zu ändern, müssen Sie über eine gesicherte Verbindung zu einer Stelle verfügen, an der Sie `opiekey` ausführen. Dies kann etwa die Eingabeaufforderung auf einer Maschine, der Sie vertrauen, sein. Zudem müssen Sie einen Iterationszähler vorgeben (100 ist ein guter Wert) und einen Initialwert wählen, wobei Sie auch einen zufällig generierten benutzen können. Benutzen Sie `opiepasswd` über die ungesicherte Verbindung zu der Maschine, die Sie einrichten wollen:

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
        otp-md5 498 to4268 ext
        Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
        otp-md5 499 to4269
        Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Drücken Sie **Return**, um die Vorgabe für den Initialwert zu akzeptieren. Bevor Sie nun das Zugriffspasswort (engl. *access password*) eingeben, rufen Sie über die gesicherte Verbindung `opikey` mit denselben Parametern auf:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Gehen Sie nun zurück zu der nicht gesicherten Verbindung und geben dort das eben generierte Einmalpasswort ein.

15.5.3. Erzeugen eines einzelnen Einmalpasswortes

Nachdem Sie OPIE eingerichtet haben, werden Sie beim nächsten Anmelden wie folgt begrüßt:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

Anmerkung: OPIE besitzt eine nützliche Eigenschaft, die hier nicht gezeigt ist. Wenn Sie an der Eingabeaufforderung **Return** eingeben, wird die echo-Funktion eingeschaltet, das heißt Sie sehen, was Sie tippen. Dies ist besonders nützlich, wenn Sie ein generiertes Passwort von einem Ausdruck abtippen müssen.

Jetzt müssen Sie Ihr Einmalpasswort generieren, um der Anmeldeaufforderung nachzukommen. Dies muss auf einem gesicherten System geschehen, auf dem Sie oder `opiekey` ausführen können. Dieses Programm gibt es übrigens auch für DOS, Windows und Mac OS. Es benötigt den Iterationszähler sowie den Initialwert als Parameter, die Sie mittels “cut-and-paste” direkt von der Login-Aufforderung nehmen können.

Auf dem sicheren System:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don’t use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Mit dem jetzt generierten Einmalpasswort können Sie die Anmeldeprozedur fortsetzen.

15.5.4. Erzeugen von mehreren Einmalpasswörtern

Manchmal müssen Sie sich an Orte begeben, an denen Sie keinen Zugriff auf eine sichere Maschine oder eine sichere Verbindung haben. In diesem Fall können Sie vorher mit `opiekey` einige Einmalpasswörter generieren, die Sie sich ausdrucken und mitnehmen können. Zum Beispiel:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don’t use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

Mit `-n 5` fordern Sie fünf Passwörter der Reihe nach an. Der letzte Iterationszähler wird durch `30` gegeben. Beachten Sie bitte, dass die Passwörter in der *umgekehrten* Reihenfolge, in der sie zu benutzen sind, ausgeben werden. Wenn Sie wirklich paranoid sind, schreiben Sie sich jetzt die Passwörter auf, ansonsten drucken Sie sie mit `lpr` aus. Beachten Sie, dass jede Zeile den Iterationszähler und das Einmalpasswort zeigt, trotzdem finden Sie es vielleicht hilfreich, eine Zeile nach Gebrauch durchzustreichen.

15.5.5. Einschränken der Benutzung von System-Passwörtern

OPIE kann die Verwendung von System-Passwörtern abhängig von der Quell-IP-Adresse einschränken. Die dazu nötigen Einstellungen werden in der Datei `/etc/opieaccess` vorgenommen, die bei der Installation des Systems automatisch erzeugt wird. Weitere Informationen über diese Datei und Sicherheitshinweise zu ihrer Verwendung entnehmen Sie bitte der Hilfeseite opieaccess(5).

Die Datei `opieaccess` könnte beispielsweise die folgende Zeile enthalten:

```
permit 192.168.0.0 255.255.0.0
```

Diese Zeile erlaubt es Benutzern, die sich von einer der angegebenen Quell-IP-Adressen anmelden, ihr System-Passwort zu verwenden. Beachten Sie bitte, dass eine Quell-IP-Adresse leicht gefälscht werden kann.

Findet sich in `opieaccess` kein passender Eintrag, muss die Anmeldung mit OPIE erfolgen.

15.6. TCP-Wrapper

Beigetragen von Tom Rhodes.

Wahrscheinlich hat jeder, der inetd(8) kennt, schon mal von den TCP-Wrappern gehört. Die wenigsten erkennen den vollen Nutzen der TCP-Wrapper in einer Netzumgebung. Es scheint, dass die meisten Leute Netzverbindungen mit einer Firewall absichern wollen. Auch wenn eine Firewall ein mächtiges Instrument ist, gibt es Sachen, die eine Firewall nicht kann. Eine Firewall kann beispielsweise keine Nachricht an den Verbindungsursprung senden. Genau das und mehr können aber die TCP-Wrapper. Im Folgenden werden die Funktionen der TCP-Wrapper und Beispiele für deren Konfiguration vorgestellt.

Die TCP-Wrapper erweitern die Steuerungsmöglichkeiten, die **inetd** über die Dienste unter seiner Kontrolle hat. Beispielsweise können Verbindungen protokolliert, Nachrichten zurückgesandt oder nur interne Verbindungen angenommen werden. Die TCP-Wrapper bieten nicht nur eine weitere Sicherheitsschicht, die teilweise auch von Firewalls geboten wird, sie bieten darüber hinaus Funktionen zur Steuerung von Verbindungen, die eine Firewall nicht bietet.

Die erweiterten Funktionen der TCP-Wrapper sind kein Firewall-Ersatz. Sie sollten zusammen mit einer Firewall und anderen Sicherheitsvorkehrungen eingesetzt werden. Die TCP-Wrapper sind eine weitere Sicherheitsschicht zum Schutz eines Systems.

Da die Wrapper die Funktion von **inetd** erweitern, wird im Folgenden vorausgesetzt, dass Sie den Abschnitt über die inetd-Konfiguration schon gelesen haben.

> **Anmerkung:** Streng genommen handelt es sich bei den von inetd(8) gestarteten Programmen nicht um “Daemonen”. Da sich diese Bezeichnung aber eingebürgert hat, wird sie auch in diesem Abschnitt verwendet.

15.6.1. TCP-Wrapper einrichten

Um die TCP-Wrapper unter FreeBSD zu benutzen, muss nur der **inetd** aus `rc.conf` mit den voreingestellten Optionen `-Ww` gestartet werden. Die Konfigurationsdatei `/etc/hosts.allow` darf keine Fehler enthalten; falls doch, werden die Fehler mit syslogd(8) protokolliert.

> **Anmerkung:** Im Gegensatz zu anderen Implementationen der TCP-Wrapper wird vom Gebrauch der Datei `hosts.deny` abgeraten. Die Konfiguration sollte sich vollständig in der Datei `/etc/hosts.allow` befinden.

In der einfachsten Konfiguration werden Dienste abhängig vom Inhalt der Datei `/etc/hosts.allow` erlaubt oder gesperrt. Unter FreeBSD wird in der Voreinstellung jeder von **inetd** gestartete Dienst erlaubt. Sehen wir uns zunächst die Grundkonfiguration an.

Eine Konfigurationszeile ist wie folgt aufgebaut: `Dienst : Adresse : Aktion`. `Dienst` ist der von **inetd** gestartete Dienst (auch Daemon genannt). Die `Adresse` kann ein gültiger Rechnername, eine IP-Adresse oder eine IPv6-Adresse in Klammern (`[ ]`) sein. Der Wert `allow` im Feld `Aktion` erlaubt Zugriffe, der Wert `deny` verbietet

Zugriffe. Die Zeilen in `hosts.allow` werden für jede Verbindung der Reihe nach abgearbeitet. Trifft eine Zeile auf eine Verbindung zu, wird die entsprechende Aktion ausgeführt und die Abarbeitung ist beendet.

Es gibt noch weitere Konfigurationsoptionen, die gleich erläutert werden. Das bisher Gesagte reicht, um eine einfache Regel aufzustellen. Wenn Sie einkommende POP3-Verbindungen für den Dienst `mail/qpopper` erlauben wollen, erweitern Sie `hosts.allow` um die nachstehende Zeile:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

Nachdem Sie die Zeile hinzugefügt haben, muss der **inetd** neu gestartet werden. Sie können dazu das Kommando kill(1) verwenden oder `/etc/rc.d/inetd restart` ausführen.

15.6.2. Erweiterte Konfiguration der TCP-Wrapper

Die TCP-Wrapper besitzen weitere Optionen, die bestimmen, wie Verbindungen behandelt werden. In einigen Fällen ist es gut, wenn bestimmten Rechnern oder Diensten eine Nachricht geschickt wird. In anderen Fällen soll vielleicht der Verbindungsaufbau protokolliert oder eine E-Mail an einen Administrator versandt werden. Oder ein Dienst soll nur für das lokale Netz bereitstehen. Dies alles ist mit so genannten Wildcards, Metazeichen und der Ausführung externer Programme möglich und wird in den nächsten zwei Abschnitten erläutert.

15.6.2.1. Externe Kommandos ausführen

Stellen Sie sich vor, eine Verbindung soll verhindert werden und gleichzeitig soll demjenigen, der die Verbindung aufgebaut hat, eine Nachricht geschickt werden. Auf welche Art müssen die TCP-Wrapper konfiguriert werden? Die Option `twist` führt beim Verbindungsaufbau ein Kommando aus. In der Datei `hosts.allow` ist ein Beispiel für diese Option enthalten:

```
# Alle anderen Dienste sind geschützt
ALL : ALL \
        : severity auth.info \
        : twist /bin/echo "You are not welcome to use %d from %h."
```

Für jeden Dienst, der nicht vorher in der Datei `hosts.allow` konfiguriert wurde, wird die Meldung “You are not allowed to use `daemon` from `hostname`.” zurückgegegeben. Dies ist besonders nützlich, wenn Sie die Gegenstelle sofort benachrichtigen wollen, nachdem die Verbindung getrennt wurde. Beachten Sie, dass der Text der Meldung in Anführungszeichen (") stehen *muss*, es gibt keine Ausnahmen zu dieser Regel.

> **Warnung:** Ein so konfigurierter Server ist anfällig für Denial-of-Service-Angriffe. Ein Angreifer kann die gesperrten Dienste mit Verbindungsanfragen überfluten.

Um einem Denial-of-Service-Angriff zu entgehen, benutzen Sie die Option `spawn`. Wie die Option `twist` verbietet die Option `spawn` die Verbindung und führt externe Kommandos aus. Allerdings sendet die Option `spawn` der Gegenstelle keine Rückmeldung. Sehen Sie sich die nachstehende Konfigurationsdatei an:

```
# Verbindungen von example.com sind gesperrt:
ALL : .example.com \
       : spawn (/bin/echo %a from %h attempted to access %d >> \
         /var/log/connections.log) \
```

```
: deny
```

Damit sind Verbindungen von der Domain `*.example.com` gesperrt. Jeder Verbindungsaufbau wird zudem in der Datei `/var/log/connections.log` protokolliert. Das Protokoll enthält den Rechnernamen, die IP-Adresse und den Dienst, der angesprochen wurde.

In der Konfigurationsdatei wurde beispielsweise das Metazeichen `%a` verwendet. Es gibt weitere Metazeichen, die in der Hilfeseite hosts_access(5) beschrieben werden.

15.6.2.2. Wildcards

Bisher verwendeten die Beispiele immer die Wildcard `ALL`. Es gibt andere Wildcards, welche die Funktionalität ein bisschen erweitern. Die Wildcard `ALL` passt beispielsweise auf jeden Dienst, jede Domain oder jede IP-Adresse. Eine andere Wildcard ist `PARANOID`. Sie passt auf jeden Rechner, dessen IP-Adresse möglicherweise gefälscht ist. Dies ist dann der Fall, wenn der Verbindungsaufbau von einer IP-Adresse erfolgt, die nicht zu dem übermittelten Rechnernamen passt. Das folgende Beispiel sollte das ganze etwas klarer werden lassen:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```

In diesem Beispiel werden alle Verbindungen zu `sendmail` verboten, die von einer IP-Adresse ausgehen, die nicht zum Rechnernamen passt.

> **Achtung:** Die Wildcard `PARANOID` kann einen Dienst unbrauchbar machen, wenn der Client oder der Server eine fehlerhafte DNS-Konfiguration besitzt. Seien Sie daher besonders vorsichtig, wenn Sie diese Wildcard in Ihre Konfiguration aufnehmen wollen.

Weiteres über Wildcards und deren Funktion lesen Sie bitte in der Hilfeseite hosts_access(5) nach.

Damit die gezeigten Beispiele funktionieren, müssen Sie die erste Konfigurationszeile in der Datei `hosts.allow` auskommentieren.

15.7. Kerberos5

Beigetragen von Tillman Hodgson. Beruht auf einem Beitrag von Mark Murray.

Kerberos ist ein Netzwerk-Protokoll, das Benutzer mithilfe eines sicheren Servers authentifiziert. Mit Risiken behaftete Dienste, wie das Anmelden an entfernten Systemen oder das Kopieren von Daten auf entfernte Systeme, werden durch **Kerberos** erheblich sicherer und lassen sich leichter steuern.

Kerberos hat eine Aufgabe: Die sichere Prüfung der Identität eines Benutzers (Authentifizierung) über das Netzwerk. Das System überprüft weder die Berechtigungen der Benutzer (Autorisierung), noch verfolgt es die durchgeführten Aktionen (Audit). Daher sollte **Kerberos** zusammen mit anderen Sicherheits-Systemen eingesetzt werden, die diese Funktionen bereitstellen. Die Daten einer Kommunikation können verschlüsselt werden, nachdem die Kommunikationspartner mit **Kerberos** ihre Identität geprüft haben.

Die folgenden Anweisungen beschreiben, wie Sie das mit FreeBSD gelieferte **Kerberos** einrichten. Eine vollständige Beschreibung des Systems entnehmen Sie bitte den entsprechenden Hilfeseiten.

Die Beschreibung der **Kerberos**-Installation benutzt folgende Namensräume:

- Die DNS-Domain (Zone) heißt example.org.
- Das **Kerberos**-Realm heißt EXAMPLE.ORG.

> **Anmerkung:** Benutzen Sie echte Domain-Namen, wenn Sie **Kerberos** einrichten. Damit vermeiden Sie DNS-Probleme und stellen die Zusammenarbeit mit anderen **Kerberos**-Realms sicher.

15.7.1. Geschichte

Das MIT entwickelte **Kerberos**, um Sicherheitsprobleme auf dem Netzwerk zu lösen. Das **Kerberos**-Protokoll verwendet starke Kryptographie, sodass ein Server die Identität eines Clients (der umgekehrte Vorgang ist auch möglich) über ein unsicheres Netzwerk feststellen kann.

Der Begriff Kerberos wird sowohl für das Protokoll als auch für Programme verwendet, die **Kerberos** benutzen (wie **Kerberos**-Telnet). Die aktuelle Protokollversion ist 5 und wird in RFC 1510 beschrieben.

Mehrere Implementierungen des Protokolls stehen frei zur Verfügung und decken viele Betriebssysteme ab. Das Massachusetts Institute of Technology (MIT), an dem **Kerberos** ursprünglich entwickelt wurde, entwickelt seine **Kerberos**-Version weiter. In den USA wird diese Version häufig eingesetzt, unterlag aber Export-Beschränkungen, da sie in den USA entwickelt wurde. Die MIT-Version von **Kerberos** befindet sich im Port `security/krb5`. Heimdal ist eine weitere Implementierung der Protokollversion 5. Sie wurde außerhalb der USA entwickelt und unterliegt daher keinen Export-Beschränkungen. Heimdal-**Kerberos** befindet sich im Port `security/heimdal` und das Basissystem von FreeBSD enthält eine minimale Installation von Heimdal.

Um möglichst viele Benutzer anzusprechen, verwenden die folgenden Beispiele die in FreeBSD enthaltene Heimdal-Distribution.

15.7.2. Das Heimdal KDC einrichten

Kerberos authentifiziert Benutzer an einer zentralen Stelle: dem Key Distribution Center (KDC). Das KDC verteilt *Tickets*, mit denen ein Dienst die Identität eines Benutzers feststellen kann. Alle Mitglieder eines **Kerberos**-Realms vertrauen dem KDC, daher gelten für das KDC erhöhte Sicherheitsanforderungen.

Obwohl das KDC wenig Ressourcen eines Rechners benötigt, sollte es wegen der Sicherheitsanforderungen auf einem separaten Rechner installiert werden.

Das KDC wird in `/etc/rc.conf` wie folgt aktiviert:

```
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
```

Danach wird die Konfigurationsdatei von **Kerberos**, `/etc/krb5.conf`, erstellt:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
        kdc = kerberos.example.org
       admin_server = kerberos.example.org
```

```
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Diese Einstellungen setzen voraus, dass der voll qualifizierte Name des KDCs `kerberos.example.org` ist. Wenn Ihr KDC einen anderen Namen hat, müssen Sie in der DNS-Zone einen Alias-Eintrag (CNAME-Record) für das KDC hinzufügen.

> **Anmerkung:** Auf großen Netzwerken mit einem ordentlich konfigurierten BIND DNS-Server kann die Datei verkürzt werden:
>
> ```
> [libdefaults]
>       default_realm = EXAMPLE.ORG
> ```
>
> Die Zonendatei von `example.org` muss dann die folgenden Zeilen enthalten:
>
> ```
> _kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
> _kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
> _kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
> _kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
> _kerberos           IN  TXT     EXAMPLE.ORG
> ```

> **Anmerkung:** Damit Klienten die **Kerberos**-Dienste benutzen können, muss die Datei `/etc/krb5.conf` entweder die vollständige Konfiguration enthalten oder eine minimale Konfiguration enthalten *und* zusätzlich ein DNS-Server richtig eingerichtet sein.

Im nächsten Schritt wird die **Kerberos**-Datenbank eingerichtet. Die Datenbank enthält die Schlüssel aller Prinzipale und ist mit einem Passwort geschützt. Dieses Passwort brauchen Sie nicht zu behalten, da ein davon abgeleiteter Schlüssel in der Datei `/var/heimdal/m-key` gespeichert wird. Den Schlüssel erstellen Sie, indem Sie das Programm `kstash` aufrufen und ein Passwort eingeben.

Nachdem Sie den Schlüssel in `/var/heimdal/m-key` erstellt haben, können Sie die Datenbank mit dem Kommando `kadmin` initialisieren. Verwenden Sie hierbei die Option `-l` (lokal). Mit dieser Option wird die Datenbank lokal modifiziert. Normal würde der `kadmind`-Dienst benutzt, der aber zu diesem Zeitpunkt noch nicht läuft. An der Eingabeaufforderung von `kadmin` können Sie mit dem Kommando `init` die Datenbank des Realms einrichten.

Zuletzt erstellen Sie mit dem Kommando `add` Ihren ersten Prinzipal. Benutzen Sie die voreingestellten Optionen; Sie können die Einstellungen später mit dem Kommando `modify` ändern. An der Eingabeaufforderung zeigt das Kommando `?` Hilfetexte an.

Zusammengefasst wird die Datenbank wie folgt eingerichtet:

```
# kstash
Master key: xxxxxxxx
Verifying password - Master key: xxxxxxxx

# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
```

```
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Jetzt kann das KDC gestartet werden. Führen Sie zum Start der Dienste die Kommandos `/etc/rc.d/kerberos start` und `/etc/rc.d/kadmind start` aus. Obwohl zu diesem Zeitpunkt noch keine kerberisierten Dienste laufen, können Sie die Funktion des KDCs schon überprüfen. Für den eben angelegten Benutzer können Sie sich vom KDC Tickets holen und diese Tickets anzeigen:

```
% kinit tillman
tillman@EXAMPLE.ORG's Password:

% klist
Credentials cache: FILE: /tmp/krb5cc_500
        Principal: tillman@EXAMPLE.ORG

  Issued           Expires          Principal
Aug 27 15:37:58  Aug 28 01:37:58  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

Dieses Ticket kann, nachdem Sie Ihre Arbeit beendet haben, zurückgezogen werden:

```
% kdestroy
```

15.7.3. Kerberos-Dienste einrichten

Alle Rechner, die kerberisierte Dienste anbieten, müssen eine Kopie der **Kerberos**-Konfigurationsdatei `/etc/krb5.conf` besitzen. Sie können die Datei einfach vom KDC kopieren.

Anschließend müssen Sie die Datei `/etc/krb5.keytab` erzeugen. Im Gegensatz zu normalen Workstations benötigt jeder Server eine `keytab`. Diese Datei enthält den Schlüssel des Servers, mit dem sich der Server und das KDC gegenseitig authentifizieren können. Die Datei muss sicher auf den Server transportiert werden (beispielsweise mit scp(1) oder einer Diskette). Unter keinen Umständen darf die Datei im Klartext, zum Beispiel mit FTP, übertragen werden, da sonst die Sicherheit des Servers gefährdet ist.

Sie können die `keytab` auch mit dem Programm `kadmin` übertragen. Da Sie mit `kadmin` sowieso einen Host-Prinzipal für den Server einrichten müssen, ist das ganz praktisch.

Sie müssen allerdings schon ein Ticket besitzen und berechtigt sein, `kadmin` auszuführen. Die Berechtigung erhalten Sie durch einen Eintrag in der Zugriffskontrollliste `kadmind.acl`. Weitere Informationen über Zugriffskontrolllisten erhalten Sie in den Heimdal-Info-Seiten (`info heimdal`) im Abschnitt "Remote administration". Wenn der Zugriff auf `kadmin` von entfernten Maschinen verboten ist, müssen Sie sich sicher auf dem KDC anmelden (lokale Konsole, ssh(1) oder kerberisiertes Telnet) und die `keytab` lokal mit `kadmin -l` erzeugen.

Nachdem Sie die Datei `/etc/krb5.conf` installiert haben, können Sie das Kommando `kadmin` benutzen. An der Eingabeaufforderung von `kadmin` erstellt das Kommando `add --random-key` den Host-Prinzipal und das Kommando `ext` extrahiert den Schlüssel des Prinzipals in eine Datei:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
```

```
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.example.org
kadmin> exit
```

Das Kommando `ext` (von *extract*) speichert den extrahierten Schlüssel in der Datei `/etc/krb5.keytab`.

Wenn auf dem KDC, vielleicht aus Sicherheitsgründen, `kadmind` nicht läuft, können Sie das Kommando `kadmin` von entfernten Rechnern nicht benutzen. In diesem Fall legen Sie den Host-Prinzipal `host/myserver.EXAMPLE.ORG` direkt auf dem KDC an. Den Schlüssel extrahieren Sie in eine temporäre Datei (damit die Datei `/etc/krb5.keytab` nicht überschrieben wird):

```
# kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

Anschließend müssen Sie die erzeugte `example.keytab` sicher auf den Server kopieren (mit `scp` oder mithilfe einer Diskette). Geben Sie auf jeden Fall einen anderen Namen für die `keytab` an, weil sonst die `keytab` des KDCs überschrieben würde.

Wegen der Datei `krb5.conf` kann der Server nun mit dem KDC kommunizieren und seine Identität mithilfe der Datei `krb5.keytab` nachweisen. Jetzt können wir kerberisierte Dienste aktivieren. Für `telnet` muss die folgende Zeile in `/etc/inetd.conf` eingefügt werden:

```
telnet    stream  tcp     nowait  root    /usr/libexec/telnetd  telnetd -a user
```

Ausschlaggebend ist, dass die Authentifizierungs-Methode mit `-a` auf `user` gesetzt wird. Weitere Details entnehmen Sie bitte der Hilfeseite telnetd(8).

Nachdem sie die Zeile in `/etc/inetd.conf` eingefügt haben, starten Sie inetd(8) mit dem Kommando `/etc/rc.d/inetd restart` durch.

15.7.4. Kerberos-Clients einrichten

Ein Client lässt sich leicht einrichten. Sie benötigen nur die **Kerberos**-Konfigurationsdatei `/etc/krb5.conf`. Kopieren Sie die Konfigurationsdatei einfach vom KDC auf den Client.

Sie können jetzt mit `kinit` Tickets anfordern, mit `klist` Tickets anzeigen und mit `kdestroy` Tickets löschen. Sie können mit **Kerberos**-Anwendungen kerberisierte Server ansprechen. Wenn das nicht funktioniert, Sie aber Tickets anfordern können, hat wahrscheinlich der kerberisierte Server ein Problem und nicht der Client oder das KDC.

Wenn Sie eine Anwendung wie `telnet` testen, können Sie mit einem Paket-Sniffer (beispielsweise tcpdump(1)) überprüfen, dass Passwörter verschlüsselt übertragen werden. Probieren Sie auch die Option `-x` von `telnet`, die den gesamten Datenverkehr verschlüsselt (analog zu `ssh`).

Zu Heimdal gehören noch weitere Anwendungen. Allerdings enthält das FreeBSD-Basissystem nur eine minimale Heimdal-Installation mit nur einer kerberisierten Anwendung: `telnet`.

Der Heimdal-Port enthält noch mehr kerberisierte Anwendungen wie `ftp`, `rsh`, `rcp` und `rlogin`. Der MIT-Port enthält ebenfalls weitere kerberisierte Anwendungen.

15.7.5. `.k5login` und `.k5users`

Normalerweise wird ein **Kerberos**-Prinzipal wie `tillman@EXAMPLE.ORG` auf ein lokales Benutzerkonto, beispielsweise `tillman`, abgebildet. Daher benötigen Client-Anwendungen (zum Beispiel `telnet`) keinen Benutzernamen.

Manchmal wird aber Zugriff auf ein lokales Benutzerkonto benötigt, zu dem es keinen passenden **Kerberos**-Prinzipal gibt. Der Prinzipal `tillman@EXAMPLE.ORG` bräuchte beispielsweise Zugriff auf das Konto `webdevelopers`. Ebenso könnten andere Prinzipale auf dieses Konto zugreifen wollen.

Die Dateien `.k5login` und `.k5users` im Heimatverzeichnis eines Benutzerkontos gewähren Zugriffe ähnlich wie die Dateien `.hosts` und `.rhosts`. Um den Prinzipalen `tillman@example.org` und `jdoe@example.org` auf das Konto `webdevelopers` zu geben, wird im Heimatverzeichnis von `webdevelopers` die Datei `.k5login` mit folgendem Inhalt angelegt:

```
tillman@example.org
jdoe@example.org
```

Die angegebenen Prinzipale haben nun ohne ein gemeinsames Passwort Zugriff auf das Konto.

Einzelheiten entnehmen Sie bitte den Hilfeseiten zu diesen Dateien. Die Datei `.k5users` wird in der Hilfeseite des Kommandos `ksu` beschrieben.

15.7.6. Tipps und Fehlersuche

- Wenn Sie den Heimdal-Port oder den MIT-Port benutzen, muss in der Umgebungsvariable `PATH` der Pfad zu den Programmen des Ports vor dem Pfad zu den **Kerberos**-Programmen des Systems stehen.
- Sind die Uhrzeiten der Systeme synchronisiert? Wenn nicht, schlägt vielleicht die Authentifizierung fehl. Abschnitt 30.10 beschreibt, wie Sie mithilfe von NTP die Uhrzeiten synchronisieren.
- Die MIT- und Heimdal-Systeme arbeiten bis auf `kadmin` gut zusammen. Für `kadmin` wurde das Protokoll nicht normiert.
- Wenn Sie den Namen eines Rechners ändern, müssen Sie auch den `host/`-Prinzipal ändern und die Datei `keytab` aktualisieren. Dies betrifft auch spezielle Einträge wie den Prinzipal für Apaches `www/mod_auth_kerb`.
- Die Rechnernamen müssen vor- und rückwärts aufgelöst werden (im DNS oder in `/etc/hosts`). CNAME-Einträge im DNS funktionieren, aber die entsprechenden A- und PTR-Einträge müssen vorhanden und richtig sein. Wenn sich Namen nicht auflösen lassen, ist die Fehlermeldung nicht gerade selbstsprechend: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.
- Einige Betriebssysteme installieren `ksu` mit falschen Zugriffsrechten; es fehlt das Set-UID-Bit für `root`. Das mag aus Sicherheitsgründen richtig sein, doch funktioniert `ksu` dann nicht. Dies ist kein Fehler des KDCs.
- Wenn Sie für einen Prinzipal unter MIT-**Kerberos** Tickets mit einer längeren Gültigkeit als der vorgegebenen zehn Stunden einrichten wollen, müssen Sie zwei Sachen ändern. Benutzen Sie das `modify_principal` von `kadmin`, um die maximale Gültigkeitsdauer für den Prinzipal selbst und den Prinzipal `krbtgt` zu erhöhen.
- Mit einem Packet-Sniffer können Sie feststellen, dass Sie sofort nach dem Aufruf von `kinit` eine Antwort vom KDC bekommen – noch bevor Sie überhaupt ein Passwort eingegeben haben! Das ist in Ordnung: Das KDC händigt ein Ticket-Granting-Ticket (TGT) auf Anfrage aus, da es durch einen vom Passwort des Benutzers abgeleiteten Schlüssel geschützt ist. Wenn das Passwort eingegeben wird, wird es nicht zum KDC gesendet, sondern zum Entschlüsseln der Antwort des KDCs benutzt, die `kinit` schon erhalten hat. Wird die Antwort

erfolgreich entschlüsselt, erhält der Benutzer einen Sitzungs-Schlüssel für die künftige verschlüsselte Kommunikation mit dem KDC und das Ticket-Granting-Ticket. Das Ticket-Granting-Ticket wiederum ist mit dem Schlüssel des KDCs verschlüsselt. Diese Verschlüsselung ist für den Benutzer völlig transparent und erlaubt dem KDC, die Echtheit jedes einzelnen TGT zu prüfen.

- Wenn Sie **OpenSSH** verwenden und Tickets mir einer langen Gültigkeit (beispielsweise einer Woche) benutzen, setzen Sie die Option `TicketCleanup` in der Datei `sshd_config` auf `no`. Ansonsten werden Ihre Tickets gelöscht, wenn Sie sich abmelden.
- Host-Prinzipale können ebenfalls Tickets mit längerer Gültigkeit besitzen. Wenn der Prinzipal eines Benutzers über ein Ticket verfügt, das eine Woche gültig ist, das Ticket des Host-Prinzipals aber nur neun Stunden gültig ist, funktioniert der Ticket-Cache nicht wie erwartet. Im Cache befindet sich dann ein abgelaufenes Ticket des Host-Prinzipals.
- Wenn Sie mit `krb5.dict` die Verwendung schlechter Passwörter verhindern wollen, geht das nur mit Prinzipalen, denen eine Passwort-Policy zugewiesen wurde. Die Hilfeseite von `kadmind` beschreibt kurz, wie `krb5.dict` verwendet wird. Das Format von `krb5.dict` ist einfach: Die Datei enthält pro Zeile ein Wort. Sie können daher einen symbolischen Link auf `/usr/share/dict/words` erstellen.

15.7.7. Unterschiede zum MIT-Port

Der Hauptunterschied zwischen MIT-**Kerberos** und Heimdal-**Kerberos** ist das Kommando `kadmin`. Die Befehlssätze des Kommandos (obwohl funktional gleichwertig) und das verwendete Protokoll unterscheiden sich in beiden Varianten. Das KDC lässt sich nur mit dem `kadmin` Kommando der passenden **Kerberos**-Variante verwalten.

Für dieselbe Funktion können auch die Client-Anwendungen leicht geänderte Kommandozeilenoptionen besitzen. Folgen Sie bitte der Anleitung auf der **Kerberos**-Seite (http://web.mit.edu/Kerberos/www/) des MITs. Achten Sie besonders auf den Suchpfad für Anwendungen. Der MIT-Port wird standardmäßig in `/usr/local/` installiert. Wenn die Umgebungsvariable `PATH` zuerst die Systemverzeichnisse enthält, werden die Systemprogramme anstelle der MIT-Programme ausgeführt.

> **Anmerkung:** Wenn Sie den MIT-Port `security/krb5` verwenden, erscheint bei der Anmeldung mit `telnetd` und `klogind` die Fehlermeldung `incorrect permissions on cache file`. Lesen Sie dazu bitte die im Port enthaltene Datei `/usr/local/share/doc/krb5/README.FreeBSD`. Wichtig ist, dass zur Authentifizierung die Binärdatei `login.krb5` verwendet wird, die für durchgereichte Berechtigungen die Eigentümer korrekt ändert.

In der Datei `rc.conf` müssen folgende Zeilen aufgenommen werden:

```
kerberos5_server="/usr/local/sbin/krb5kdc"
kadmind5_server="/usr/local/sbin/kadmind"
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
```

Diese Zeilen sind notwendig, weil die Anwendungen von MIT-Kerberos Binärdateien unterhalb von `/usr/local` installieren.

15.7.8. Beschränkungen von Kerberos

15.7.8.1. Kerberos muss ganzheitlich verwendet werden

Jeder über das Netzwerk angebotetene Dienst muss mit **Kerberos** zusammenarbeiten oder auf anderen Wegen gegen Angriffe aus dem Netzwerk geschützt sein. Andernfalls können Berechtigungen gestohlen und wiederverwendet werden. Es ist beispielsweise nicht sinnvoll, für Anmeldungen mit `rsh` und `telnet` **Kerberos** zu benutzen, dagegen aber POP3-Zugriff auf einen Mail-Server zu erlauben, da POP3 Passwörter im Klartext versendet.

15.7.8.2. Kerberos ist für Einbenutzer-Systeme gedacht

In Mehrbenutzer-Umgebungen ist **Kerberos** unsicherer als in Einbenutzer-Umgebungen, da die Tickets im für alle lesbaren Verzeichnis `/tmp` gespeichert werden. Wenn ein Rechner von mehreren Benutzern verwendet wird, ist es möglich, dass Tickets gestohlen werden.

Dieses Problem können Sie lösen, indem Sie mit der Kommandozeilenoption `-c` oder besser mit der Umgebungsvariablen `KRB5CCNAME` einen Ort für die Tickets vorgeben. Diese Vorgehensweise wird leider selten benutzt. Es reicht, die Tickets im Heimatverzeichnis eines Benutzers zu speichern und mit Zugriffsrechten zu schützen.

15.7.8.3. Das KDC ist verwundbar

Das KDC muss genauso abgesichert werden wie die auf ihm befindliche Passwort-Datenbank. Auf dem KDC dürfen keine anderen Dienste laufen und der Rechner sollte physikalisch gesichert sein. Die Gefahr ist groß, da **Kerberos** alle Passwörter mit einem Schlüssel, dem Haupt-Schlüssel, verschlüsselt. Der Haupt-Schlüssel wiederum wird in einer Datei auf dem KDC gespeichert.

Ein kompromittierter Haupt-Schlüssel ist nicht ganz so schlimm wie allgemein angenommen. Der Haupt-Schlüssel wird nur zum Verschlüsseln der Passwort-Datenbank und zum Initialisieren des Zufallsgenerators verwendet. Solange der Zugriff auf das KDC abgesichert ist, kann ein Angreifer wenig mit dem Haupt-Schlüssel anfangen.

Wenn das KDC nicht zur Verfügung steht, vielleicht wegen eines Denial-of-Service Angriffs oder wegen eines Netzwerkproblems, ist eine Authentifizierung unmöglich. Damit können die Netzwerk-Dienste nicht benutzt werden; das KDC ist also ein optimales Ziel für einen Denial-of-Service Angriff. Sie können diesem Angriff ausweichen, indem Sie mehrere KDCs (einen Master und einen oder mehrere Slaves) verwenden. Der Rückfall auf ein sekundäres KDC oder eine andere Authentifizierungs-Methode (dazu ist PAM bestens geeignet) muss sorgfältig eingerichtet werden.

15.7.8.4. Mängel von Kerberos

Mit **Kerberos** können sich Benutzer, Rechner und Dienste gegenseitig authentifizieren. Allerdings existiert kein Mechanismus, der das KDC gegenüber Benutzern, Rechnern oder Diensten authentifiziert. Ein verändertes `kinit` könnte beispielsweise alle Benutzernamen und Passwörter abfangen. Die von veränderten Programmen ausgehende Gefahr können Sie lindern, indem Sie die Integrität von Dateien mit Werkzeugen wie `security/tripwire` prüfen.

15.7.9. Weiterführende Dokumentation

- The Kerberos FAQ (http://www.faqs.org/faqs/Kerberos-faq/general/preamble.html)
- Designing an Authentication System: a Dialogue in Four Scenes (http://web.mit.edu/Kerberos/www/dialogue.html)
- RFC 1510, The **Kerberos** Network Authentication Service (V5) (http://www.ietf.org/rfc/rfc1510.txt?number=1510)
- MIT **Kerberos**-Seite (http://web.mit.edu/Kerberos/www/)
- Heimdal **Kerberos**-Seite (http://www.pdc.kth.se/heimdal/)

15.8. OpenSSL

Beigetragen von Tom Rhodes.

Es wird oft übersehen, dass **OpenSSL** Teil des FreeBSD-Basissystems ist. **OpenSSL** bietet eine verschlüsselte Transportschicht oberhalb der normalen Kommunikationsschicht und kann daher zusammen mit vielen Netzdiensten benutzt werden.

Anwendungsbeispiele für **OpenSSL** sind die verschlüsselte Authentifizierung von E-Mail-Clients oder Web-Transaktionen wie das Bezahlen mit einer Kreditkarte. **OpenSSL** kann während des Baus in viele Ports, wie `www/apache22` und `mail/claws-mail`, integriert werden.

> **Anmerkung:** Ist beim Aufruf von `make` die Variable `WITH_OPENSSL_BASE` nicht explizit auf `yes` gesetzt, baut die Ports-Sammlung meist den Port `security/openssl`.

Das **OpenSSL** von FreeBSD stellt die Protokolle Secure Sockets Layer v2/v3 (SSLv2/SSLv3) und Transport Layer Security v1 (TLSv1) zur Verfügung. Die **OpenSSL**-Bibliotheken stellen kryptographische Funktionen bereit.

> **Anmerkung:** Mit **OpenSSL** kann der IDEA-Algorithmus verwendet werden, wegen Patenten in den USA ist der Algorithmus in der Voreinstellung allerdings deaktiviert. Wenn Sie die IDEA-Lizenz akzeptieren, können Sie den IDEA-Algorithmus aktivieren, indem Sie die Variable `MAKE_IDEA` in `make.conf` setzen.

Meist wird **OpenSSL** eingesetzt, um Zertifikate für Anwendungen bereitzustellen. Die Zertifikate stellen die Identität einer Firma oder eines Einzelnen sicher. Wenn ein Zertifikat nicht von einer Zertifizierungsstelle (*Certificate Authority*, CA) gegengezeichnet wurde, erhalten Sie normalerweise eine Warnung. Eine Zertifizierungsstelle ist eine Firma wie VeriSign (http://www.verisign.com/), die Zertifikate von Personen oder Firmen gegenzeichnet und damit die Korrektheit der Zertifikate bestätigt. Diese Prozedur kostet Geld, ist aber keine Voraussetzung für den Einsatz von Zertifikaten, beruhigt aber sicherheitsbewusste Benutzer.

15.8.1. Zertifikate erzeugen

Ein Zertifikat erzeugen Sie mit dem nachstehenden Kommando:

```
# openssl req -new -nodes -out req.pem -keyout cert.pem
Generating a 1024 bit RSA private key
..............++++++
.....................................++++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SOME PASSWORD
An optional company name []:Another Name
```

Beachten Sie bitte, dass die Eingabe bei “Common Name” ein gültiger Domain-Name sein muss. Eine andere Eingabe erzeugt ein unbrauchbares Zertifikat. Das Zertifikat kann mit einer Gültigkeitsdauer und anderen Verschlüsselungsalgorithmen erzeugt werden. Die Hilfeseite openssl(1) beschreibt die zur Verfügung stehenden Optionen.

Das Verzeichnis, in dem Sie den letzten Befehl ausgeführt haben, enthält nun zwei Dateien: Die Anforderung für ein neues Zertifikat wurde in `req.pem` gespeichert. Diese Datei können Sie an eine Zertifizierungsstelle senden, wo Ihre Angaben geprüft werden. Nach erfolgreicher Prüfung wird das Zertifikat an Sie zurückgesandt. Die zweite Datei, `cert.pem`, enthält den privaten Schlüssel für Ihr Zertifikat und darf auch keine Fall in fremde Hände geraten, da ein Angreifer sonst in der Lage ist, anderen Personen oder Rechnern vorzugaukeln, dass es sich bei ihm um Sie handelt.

Wenn Sie keine Signatur einer Zertifizierungsstelle benötigen, können Sie ein selbst-signiertes Zertifikat erstellen. Erzeugen Sie dazu zuerst einen RSA-Schlüssel:

```
# openssl dsaparam -rand -genkey -out myRSA.key 1024
```

Erzeugen Sie dann den CA-Schlüssel:

```
# openssl gendsa -des3 -out myca.key myRSA.key
```

Erstellen Sie mit diesem Schlüssel das Zertifikat:

```
# openssl req -new -x509 -days 365 -key myca.key -out new.crt
```

Zwei neue Dateien befinden sich nun im Verzeichnis: Der Schlüssel der Zertifizierungsstelle `myca.key` und das Zertifikat selbst, `new.crt`. Sie sollten in einem Verzeichnis, vorzugsweise unterhalb von `/etc` abgelegt werden, das nur von `root` lesbar ist. Setzen Sie die Zugriffsrechte der Dateien mit `chmod` auf `0700`.

15.8.2. Beispiel für Zertifikate

Was fangen Sie mit einem Zertifikat an? Sie könnten damit beispielsweise die Verbindungen zu **Sendmail** verschlüsseln. Dies würde die Klartext-Authentifizierung für Benutzer des lokalen MTA überflüssig machen.

> **Anmerkung:** Das ist nicht unbedingt die beste Lösung, da einige MUAs Warnungen ausgeben, wenn ein Zertifikat nicht lokal installiert ist. Die Installation von Zertifikaten wird in der Dokumentation der MUAs beschrieben.

Ergänzen Sie die Konfigurationsdatei von **sendmail** (`.mc`) um die nachstehenden Zeilen:

```
dnl SSL Options
define(‘confCACERT_PATH’,‘/etc/certs’)dnl
define(‘confCACERT’,‘/etc/certs/new.crt’)dnl
define(‘confSERVER_CERT’,‘/etc/certs/new.crt’)dnl
define(‘confSERVER_KEY’,‘/etc/certs/myca.key’)dnl
define(‘confTLS_SRV_OPTIONS’, ‘V’)dnl
```

Im Verzeichnis `/etc/certs` befindet sich der Schlüssel und das Zertifikat. Bauen Sie danach im Verzeichnis `/etc/mail` mit dem Kommando `make install` die `.cf`-Datei und starten Sie anschließend **sendmail** mit `make restart` neu.

Wenn alles gut ging, erscheinen keine Fehlermeldungen in der Datei `/var/log/maillog` und Sie sehen **sendmail** in der Prozessliste.

Testen Sie nun den Mailserver mit dem Kommando telnet(1):

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is ’^]’.
220 example.com ESMTP Sendmail 8.12.10/8.12.10; Tue, 31 Aug 2004 03:41:22 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Wenn in einer Zeile `STARTTLS` erscheint, hat alles funktioniert.

15.9. VPNs mit IPsec

Geschrieben von Nik Clayton.

Dieser Abschnitt beschreibt, wie Sie mit FreeBSD-Gateways ein *Virtual-Private-Network* (VPN) einrichten. Als Beispiel wird ein VPN zwischen zwei Netzen verwendet, die über das Internet miteinander verbunden sind.

15.9.1. IPsec Grundlagen

Geschrieben von Hiten M. Pandya.

Dieser Abschnitt zeigt Ihnen, wie Sie IPsec einrichten. Um IPsec einzurichten, sollten Sie einen neuen Kernel bauen können (siehe Kapitel 9).

IPsec ist ein Protokoll, das auf dem Internet-Protokoll (IP) aufbaut. Mit IPsec können mehrere Systeme geschützt miteinander kommunizieren. Das in FreeBSD realisierte IPsec-Protokoll baut auf der KAME-Implementierung (http://www.kame.net/) auf und unterstützt sowohl IPv4 als auch IPv6.

IPsec besteht wiederum aus zwei Protokollen:

- *Encapsulated Security Payload (ESP)* verschlüsselt IP-Pakete mit einem symmetrischen Verfahren (beispielsweise Blowfish oder 3DES). Damit werden die Pakete vor Manipulationen Dritter geschützt.
- Der *Authentication Header (AH)* enthät eine kryptographische Prüfsumme, die sicher stellt, dass ein IP-Paket nicht verändert wurde. Der Authentication-Header folgt nach dem normalen IP-Header und erlaubt dem Empfänger eines IP-Paketes, dessen Integrität zu prüfen.

ESP und AH können, je nach Situation, zusammen oder einzeln verwendet werden.

IPsec kann in zwei Modi betrieben werden: Der *Transport-Modus* verschlüsselt die Daten zwischen zwei Systemen. Der *Tunnel-Modus* verbindet zwei Subnetze miteinander. Durch einen Tunnel können dann beispielsweise verschlüsselte Daten übertragen werden. Ein Tunnel wird auch als Virtual-Private-Network (VPN) bezeichnet. Detaillierte Informationen über das IPsec-Subsystem von FreeBSD enthält die Hilfeseite ipsec(4).

Die folgenden Optionen in der Kernelkonfiguration aktivieren IPsec:

```
options   IPSEC        #IP security
device    crypto
```

Wenn Sie zur Fehlersuche im IPsec-Subsystem Unterstützung wünschen, sollten Sie die folgende Option ebenfalls aktivieren:

```
options   IPSEC_DEBUG  #debug for IP security
```

15.9.2. Was ist ein VPN?

Es gibt keinen Standard, der festlegt, was ein Virtual-Private-Network ist. VPNs können mit verschiedenen Techniken, die jeweils eigene Vor- und Nachteile besitzen, implementiert werden. Dieser Abschnitt stellt eine Möglichkeit vor, ein VPN aufzubauen.

15.9.3. Das Szenario: Zwei Netzwerke, ein Heim- und ein Firmennetzwerk. Beide sind mit dem Internet verbunden und verhalten sich dank VPN wie ein Netzwerk.

Dieses Szenario hat die folgenden Vorausetzungen:

- Es müssen zwei Netzwerke vorhanden sein.
- Beide Netzwerke müssen intern IP benutzen.
- Beide Netzwerke sind über ein FreeBSD-Gateway mit dem Internet verbunden.
- Der Gateway jedes Netzwerks besitzt mindestens eine öffentliche IP-Adresse.
- Die intern verwendeten IP-Adressen können private oder öffentliche Adressen sein. Sie dürfen sich nicht überlappen; zum Beispiel: nicht beide sollten `192.168.1.x` benutzen.

15.9.4. Konfiguration von IPsec in FreeBSD

Geschrieben von Tom Rhodes.

Als erstes muss `security/ipsec-tools` aus der Ports-Sammlung installiert werden. Dieses Softwarepaket Dritter enthält einige Anwendungen, die ihnen bei der Konfiguration von IPsec helfen.

Als nächstes müssen zwei gif(4)-Pseudogeräte angelegt werden, um die Pakete zu tunneln und dafür zu sorgen, dass beide Netzwerke richtig miteinander kommunizieren können. Geben Sie als Benutzer `root` die folgenden Befehle ein: Austausch der *`internen`* und *`externen`* Werte durch die realen internen und externen Gateways:

```
# ifconfig gif0 create

# ifconfig gif0 internal1 internal2

# ifconfig gif0 tunnel external1 external2
```

Beispiel: Die öffentliche IP-Adresse des Firmennetzwerkes (LAN) ist: `172.16.5.4` mit einer internen IP-Adresse von `10.246.38.1`. Das Heimnetzwerk (LAN) hat die öffentliche IP-Adresse `192.168.1.12` mit der internen privaten IP-Adresse `10.0.0.5`.

Dies mag verwirrend erscheinen, schauen Sie sich deshalb das folgende Beispiel aus dem ifconfig(8)-Befehl an:

```
Gateway 1:

gif0: flags=8051 mtu 1280
tunnel inet 172.16.5.4 --> 192.168.1.12
inet6 fe80::2e0:81ff:fe02:5881%gif0 prefixlen 64 scopeid 0x6
inet 10.246.38.1 --> 10.0.0.5 netmask 0xffffff00

Gateway 2:

gif0: flags=8051 mtu 1280
tunnel inet 192.168.1.12 --> 172.16.5.4
inet 10.0.0.5 --> 10.246.38.1 netmask 0xffffff00
inet6 fe80::250:bfff:fe3a:c1f%gif0 prefixlen 64 scopeid 0x4
```

Wenn Sie fertig sind, sollten beide privaten IPs mit dem ping(8) Befehl, wie die folgende Darstellung zeigt, erreichbar sein:

```
priv-net# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=64 time=42.786 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=19.255 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=20.440 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=21.036 ms
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.255/25.879/42.786/9.782 ms

corp-net# ping 10.246.38.1
PING 10.246.38.1 (10.246.38.1): 56 data bytes
64 bytes from 10.246.38.1: icmp_seq=0 ttl=64 time=28.106 ms
64 bytes from 10.246.38.1: icmp_seq=1 ttl=64 time=42.917 ms
64 bytes from 10.246.38.1: icmp_seq=2 ttl=64 time=127.525 ms
64 bytes from 10.246.38.1: icmp_seq=3 ttl=64 time=119.896 ms
64 bytes from 10.246.38.1: icmp_seq=4 ttl=64 time=154.524 ms
--- 10.246.38.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.106/94.594/154.524/49.814 ms
```

Wie erwartet, können nun beiden Seiten ICMP-Pakete von ihren privaten Adressen senden und empfangen. Als nächstes müssen beide Gateways so konfiguriert werden, dass sie die Pakete des anderen Netzwerkes richtig routen. Mit dem folgenden Befehl erreicht man das Ziel:

```
# corp-net# route add 10.0.0.0 10.0.0.5 255.255.255.0

# corp-net# route add net 10.0.0.0: gateway 10.0.0.5

# priv-net# route add 10.246.38.0 10.246.38.1 255.255.255.0

# priv-net# route add host 10.246.38.0: gateway 10.246.38.1
```

Ab jetzt sollten die Rechner von den Gateways sowie von den Rechnern hinter den Gateways erreichbar sein. Dies wird an dem folgendem Beispiel deutlich:

```
corp-net# ping 10.0.0.8
PING 10.0.0.8 (10.0.0.8): 56 data bytes
64 bytes from 10.0.0.8: icmp_seq=0 ttl=63 time=92.391 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=63 time=21.870 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=63 time=198.022 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=63 time=22.241 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=63 time=174.705 ms
--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.870/101.846/198.022/74.001 ms

priv-net# ping 10.246.38.107
PING 10.246.38.1 (10.246.38.107): 56 data bytes
64 bytes from 10.246.38.107: icmp_seq=0 ttl=64 time=53.491 ms
64 bytes from 10.246.38.107: icmp_seq=1 ttl=64 time=23.395 ms
64 bytes from 10.246.38.107: icmp_seq=2 ttl=64 time=23.865 ms
64 bytes from 10.246.38.107: icmp_seq=3 ttl=64 time=21.145 ms
```

```
64 bytes from 10.246.38.107: icmp_seq=4 ttl=64 time=36.708 ms
--- 10.246.38.107 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.145/31.721/53.491/12.179 ms
```

Das Konfigurieren der Tunnel ist der einfache Teil. Die Konfiguration einer sicheren Verbindung geht sehr viel mehr in die Tiefe. Die folgende Konfiguration benutzt pre-shared (PSK) RSA-Schlüssel. Abgesehen von den IP-Adressen, sind beide `/usr/local/etc/racoon/racoon.conf` identisch und sehen ähnlich aus wie:

```
path    pre_shared_key  "/usr/local/etc/racoon/psk.txt"; #location of pre-shared key file
log     debug;  #log verbosity setting: set to 'notify' when testing and debugging is complete

padding # options are not to be changed
{
        maximum_length  20;
        randomize       off;
        strict_check    off;
        exclusive_tail  off;
}

timer   # timing options. change as needed
{
        counter         5;
        interval        20 sec;
        persend         1;
#       natt_keepalive  15 sec;
        phase1          30 sec;
        phase2          15 sec;
}

listen  # address [port] that racoon will listening on
{
        isakmp          172.16.5.4 [500];
        isakmp_natt     172.16.5.4 [4500];
}

remote  192.168.1.12 [500]
{
        exchange_mode   main,aggressive;
        doi             ipsec_doi;
        situation       identity_only;
        my_identifier   address 172.16.5.4;
        peers_identifier        address 192.168.1.12;
        lifetime        time 8 hour;
        passive         off;
        proposal_check  obey;
#       nat_traversal   off;
        generate_policy off;

                        proposal {
                                encryption_algorithm    blowfish;
                                hash_algorithm          md5;
                                authentication_method   pre_shared_key;
```

```
                                lifetime time           30 sec;
                                dh_group                1;
                        }
}

sainfo  (address 10.246.38.0/24 any address 10.0.0.0/24 any)    # address $network/$netmask $type
{                                                               # $network must be the two interna
        pfs_group       1;
        lifetime        time    36000 sec;
        encryption_algorithm    blowfish,3des,des;
        authentication_algorithm        hmac_md5,hmac_sha1;
        compression_algorithm   deflate;
}
```

Die Erklärung einer jeden verfügbaren Option würde den Rahmen dieses Textes sprengen. Es gibt sehr viele relevante Informationen in der **racoon**-Konfigurationsanleitung.

Die SPD-Methoden müssn noch konfiguriert werden so dass, FreeBSD und **racoon** in der Lage sind den Netzwerkverkehr zwischen den Hosts zu ver- und entschlüsseln.

Dies wird durch ein einfaches Shellscript ähnlich wie das folgende, das auf dem Firmennetzwerk-Gateway liegt, ausgeführt. Diese Datei wird während der Systeminitialisierung ausgeführt und sollte unter `/usr/local/etc/racoon/setkey.conf` abgespeichert werden.

```
flush;
spdflush;

# To the home network
spdadd 10.246.38.0/24 10.0.0.0/24 any -P out ipsec esp/tunnel/172.16.5.4-192.168.1.12/use;
spdadd 10.0.0.0/24 10.246.38.0/24 any -P in ipsec esp/tunnel/192.168.1.12-172.16.5.4/use;
```

Einmal abgespeichert, kann **racoon** durch das folgende Kommando auf beiden Gateways gestartet werden:

```
# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log
```

Die Ausgabe sollte so ähnlich aussehen:

```
corp-net# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf
Foreground mode.
2006-01-30 01:35:47: INFO: begin Identity Protection mode.
2006-01-30 01:35:48: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:35:55: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:36:04: INFO: ISAKMP-SA established 172.16.5.4[500]-192.168.1.12[500] spi:623b9b3bd2
2006-01-30 01:36:05: INFO: initiate new phase 2 negotiation: 172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]->172.16.5.4[0] spi=28
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]->192.168.1.12[0] spi=47
2006-01-30 01:36:13: INFO: respond new phase 2 negotiation: 172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]->172.16.5.4[0] spi=12
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]->192.168.1.12[0] spi=17
```

Um sicherzustellen, dass der Tunnel richtig funktioniert, wechseln Sie auf eine andere Konsole und benutzen Sie tcpdump(1) mit dem folgenden Befehl, um sich den Netzwerkverkehr anzusehen. Tauschen Sie `em0` durch die richtige Netzwerkkarte aus.

```
# tcpdump -i em0 host 172.16.5.4 and dst 192.168.1.12
```

Die Ausgabe der Konsole sollte dem hier ähneln. Wenn nicht, gibt es ein Problem und ein Debuggen der ausgegebenen Daten ist notwendig.

```
01:47:32.021683 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com: ESP(spi=0x02acbf9f,seq=
01:47:33.022442 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com: ESP(spi=0x02acbf9f,seq=
01:47:34.024218 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com: ESP(spi=0x02acbf9f,seq=
```

An diesem Punkt sollten beide Netzwerke verfügbar sein und den Anschein haben, dass sie zum selben Netzwerk gehören. Meistens sind beide Netzwerke durch eine Firewall geschützt. Um den Netzwerkverkehr zwischen den beiden Netzwerken zu erlauben, ist es notwendig Regeln zu erstellen. Für die ipfw(8) Firewall fügen Sie folgende Zeilen in ihre Firewall-Konfigurationsdatei ein:

```
ipfw add 00201 allow log esp from any to any
ipfw add 00202 allow log ah from any to any
ipfw add 00203 allow log ipencap from any to any
ipfw add 00204 allow log udp from any 500 to any
```

Anmerkung: Die Regelnummern müssen eventuell, je nach ihrer Hostkonfiguration, angepasst werden.

Für Benutzer der pf(4)- oder ipf(8)-Firewall sollte folgendes funktionieren:

```
pass in quick proto esp from any to any
pass in quick proto ah from any to any
pass in quick proto ipencap from any to any
pass in quick proto udp from any port = 500 to any port = 500
pass in quick on gif0 from any to any
pass out quick proto esp from any to any
pass out quick proto ah from any to any
pass out quick proto ipencap from any to any
pass out quick proto udp from any port = 500 to any port = 500
pass out quick on gif0 from any to any
```

Zum Ende, um dem Computer den Start vom VPN während der Systeminitialisierung zu erlauben, fügen Sie folgende Zeilen in ihre `/etc/rc.conf`: ein

```
ipsec_enable="YES"
ipsec_program="/usr/local/sbin/setkey"
ipsec_file="/usr/local/etc/racoon/setkey.conf" # allows setting up spd policies on boot
racoon_enable="yes"
```

15.10. OpenSSH

Beigetragen von Chern Lee.

OpenSSH stellt Werkzeuge bereit, um sicher auf entfernte Maschinen zuzugreifen. Die Kommandos `rlogin`, `rsh`, `rcp` und `telnet` können durch **OpenSSH** ersetzt werden. Zusätzlich können TCP/IP-Verbindungen sicher durch

SSH weitergeleitet (getunnelt) werden. Mit SSH werden alle Verbindungen verschlüsselt, dadurch wird verhindert, dass die Verbindung zum Beispiel abgehört oder übernommen (*Hijacking*) werden kann.

OpenSSH wird vom OpenBSD-Projekt gepflegt und basiert auf SSH v1.2.12 mit allen aktuellen Fixen und Aktualisierungen. **OpenSSH** ist mit den SSH-Protokollen der Versionen 1 und 2 kompatibel.

15.10.1. Vorteile von OpenSSH

Mit telnet(1) oder rlogin(1) werden Daten in einer unverschlüsselten Form über das Netzwerk gesendet. Daher besteht die Gefahr, das Benutzer/Passwort Kombinationen oder alle Daten an beliebiger Stelle zwischen dem Client und dem Server abgehört werden. Mit **OpenSSH** stehen eine Reihe von Authentifizierungs- und Verschlüsselungsmethoden zur Verfügung, um das zu verhindern.

15.10.2. Aktivieren von sshd

Unter FreeBSD entscheidet der Anwender bei einer `Standard`-Installation, ob der **sshd**-Daemon aktiviert werden soll. Um zu überprüfen, ob **sshd** auf Ihrem System aktiviert ist, suchen Sie in `rc.conf` nach der folgenden Zeile:

```
sshd_enable="YES"
```

Ist diese Zeile vorhanden, wird sshd(8), der **OpenSSH**-Dæmon, beim Systemstart automatisch aktiviert. Alternativ können Sie **OpenSSH** auch über das rc(8)-Skript `/etc/rc.d/sshd` starten:

```
# /etc/rc.d/sshd start
```

15.10.3. SSH Client

ssh(1) arbeitet ähnlich wie rlogin(1):

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *******
```

Der Anmeldevorgang wird danach, wie von `rlogin` oder `telnet` gewohnt, weiterlaufen. SSH speichert einen Fingerabdruck des Serverschlüssels. Die Aufforderung, `yes` einzugeben, erscheint nur bei der ersten Verbindung zu einem Server. Weitere Verbindungen zu dem Server werden gegen den gespeicherten Fingerabdruck des Schlüssels geprüft und der Client gibt eine Warnung aus, wenn sich der empfangene Fingerabdruck von dem gespeicherten unterscheidet. Die Fingerabdrücke der Version 1 werden in `~/.ssh/known_hosts`, die der Version 2 in `~/.ssh/known_hosts2` gespeichert.

In der Voreinstellung akzeptieren aktuelle **OpenSSH**-Server nur SSH v2 Verbindungen. Wenn möglich, wird Version 2 verwendet, ist dies nicht möglich, fällt der Server auf Version 1 zurück. Der Client kann gezwungen werden, nur eine der beiden Versionen zu verwenden, indem die Option `-1` (für die Version 1) oder `-2` (für die Version 2) übergeben wird. Die Unterstützung für Version 1 ist nur noch aus Kompatibilitätsgründen zu älteren Versionen enthalten.

15.10.4. Secure Copy

Mit scp(1) lassen sich Dateien analog wie mit rcp(1) auf entfernte Maschinen kopieren. Mit `scp` werden die Dateien allerdings in einer sicheren Weise übertragen.

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password:
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

Da der Fingerabdruck schon im vorigen Beispiel abgespeichert wurde, wird er bei der Verwendung von `scp` in diesem Beispiel überprüft. Da die Fingerabdrücke übereinstimmen, wird keine Warnung ausgegeben.

Die Argumente, die `scp` übergeben werden, gleichen denen von `cp` in der Beziehung, dass die ersten Argumente die zu kopierenden Dateien sind und das letzte Argument den Bestimmungsort angibt. Da die Dateien über das Netzwerk kopiert werden, können ein oder mehrere Argumente die Form `user@host:<path_to_remote_file>` besitzen.

15.10.5. Konfiguration

Die für das ganze System gültigen Konfigurationsdateien des **OpenSSH**-Dæmons und des Clients finden sich in dem Verzeichnis `/etc/ssh`.

Die Client-Konfiguration befindet sich in `ssh_config`, die des Servers befindet sich in `sshd_config`.

Das SSH-System lässt sich weiterhin über die Anweisungen `sshd_program` (Vorgabe ist `/usr/sbin/sshd`) und `sshd_flags` in `/etc/rc.conf` konfigurieren.

15.10.6. ssh-keygen

Mit ssh-keygen(1) können DSA- oder RSA-Schlüssel für einen Benutzer erzeugt werden, die anstelle von Passwörtern verwendet werden können:

```
% ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_dsa):
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_dsa.
Your public key has been saved in /home/user/.ssh/id_dsa.pub.
The key fingerprint is:
bb:48:db:f2:93:57:80:b6:aa:bc:f5:d5:ba:8f:79:17 user@host.example.com
```

ssh-keygen(1) erzeugt einen öffentlichen und einen privaten Schlüssel für die Authentifizierung. Der private Schlüssel wird in `~/.ssh/id_dsa` oder `~/.ssh/id_rsa` gespeichert, während sich der öffentliche Schlüssel in `~/.ssh/id_dsa.pub` oder `~/.ssh/id_rsa.pub` befindet, je nachdem, ob es sich um einen DSA- oder einen RSA-Schlüssel handelt. Der öffentliche Schlüssel muss sowohl für RSA- als auch für DSA-Schlüssel in die Datei `~/.ssh/authorized_keys` auf dem entfernten Rechner aufgenommen werden, damit der Schlüssel funktioniert.

Damit werden Verbindungen zu der entfernten Maschine über SSH-Schlüsseln anstelle von Passwörtern authentifiziert.

Wenn bei der Erstellung der Schlüssel mit ssh-keygen(1) ein Passwort angegeben wurde, wird der Benutzer bei jeder Anmeldung zur Eingabe des Passworts aufgefordert. Um den Umgang mit SSH-Schlüsseln zu erleichtern, kann ssh-agent(1) die Verwaltung dieser Schlüssel für Sie übernehmen. Lesen Sie dazu den Abschnitt 15.10.7 weiter unten.

> **Warnung:** Die Kommandozeilenoptionen und Dateinamen sind abhängig von der **OpenSSH**-Version. Die für Ihr System gültigen Optionen finden Sie in der Hilfeseite ssh-keygen(1).

15.10.7. ssh-agent und ssh-add

Mit ssh-agent(1) und ssh-add(1) ist es möglich, **SSH**-Schlüssel in den Speicher zu laden, damit die Passphrase nicht jedesmal eingegeben werden muss.

ssh-agent(1) übernimmt die Authentifizierung von ihm geladener privater Schlüssel. ssh-agent(1) sollte nur dazu verwendet werden, ein anderes Programm zu starten, beispielsweise eine Shell oder einen Window-Manager.

Um ssh-agent(1) in einer Shell zu verwenden, muss es mit einer Shell als Argument aufgerufen werden. Zusätzlich müssen die zu verwaltende Identität (durch ssh-add(1)) sowie deren Passphrase für den privaten Schlüssel übergeben werden. Nachdem dies erledigt ist, kann sich ein Benutzer über ssh(1) auf jedem Rechner anmelden, der einen entsprechenden öffentlichen Schlüssel besitzt. Dazu ein Beispiel:

```
% ssh-agent csh
% ssh-add
Enter passphrase for /home/user/.ssh/id_dsa:
Identity added: /home/user/.ssh/id_dsa (/home/user/.ssh/id_dsa)
%
```

Um ssh-agent(1) unter X11 zu verwenden, müssen Sie ssh-agent(1) in Ihre `~/.xinitrc` aufnehmen. Dadurch können alle unter X11 gestarteten Programme die Dienste von ssh-agent(1) nutzen. Ihre `~/.xinitrc` könnte dazu etwas so aussehen:

```
exec ssh-agent startxfce4
```

Dadurch wird bei jedem Start von X11 zuerst ssh-agent(1) aufgerufen, das wiederum **XFCE** startet. Nachdem Sie diese Änderung durchgeführt haben, müssen Sie X11 neu starten. Danach können Sie mit ssh-add(1) Ihre SSH-Schlüssel laden.

15.10.8. SSH-Tunnel

Mit **OpenSSH** ist es möglich, einen Tunnel zu erstellen, in dem ein anderes Protokoll verschlüsselt übertragen wird.

Das folgende Kommando erzeugt einen Tunnel für **telnet**:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Dabei wurden die folgenden Optionen von `ssh` verwendet:

`-2`

Erzwingt die Version 2 des Protokolls (Benutzen Sie die Option nicht mit langsamen **SSH**-Servern).

`-N`

Zeigt an, dass ein Tunnel erstellt werden soll. Ohne diese Option würde `ssh` eine normale Sitzung öffnen.

`-f`

Zwingt `ssh` im Hintergrund zu laufen.

`-L`

Ein lokaler Tunnel wird in der Form *`localport:remotehost:remoteport`* angegeben. Die Verbindung wird dabei von dem lokalen Port *`localport`* auf einen entfernten Rechner weitergeleitet.

`user@foo.example.com`

Gibt den entfernten SSH-Server an.

Ein SSH-Tunnel erzeugt ein Socket auf `localhost` und dem angegebenen Port. Jede Verbindung, die auf dem angegebenen Socket aufgemacht wird, wird dann auf den spezifizierten entfernten Rechner und Port weitergeleitet.

Im Beispiel wird der Port *`5023`* auf die entfernte Maschine und dort auf `localhost` Port *`23`* weitergeleitet. Da der Port *`23`* für **Telnet** reserviert ist, erzeugt das eine sichere **Telnet**-Verbindung durch einen SSH-Tunnel.

Diese Vorgehensweise kann genutzt werden, um jedes unsichere TCP-Protokoll wie SMTP, POP3, FTP, usw. weiterzuleiten.

Beispiel 15-1. Mit SSH einen sicheren Tunnel für SMTP erstellen

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

Zusammen mit ssh-keygen(1) und zusätzlichen Benutzer-Accounts können Sie leicht benutzbare SSH-Tunnel aufbauen. Anstelle von Passwörtern können Sie Schlüssel benutzen und jeder Tunnel kann unter einem eigenen Benutzer laufen.

15.10.8.1. Beispiel für SSH-Tunnel

15.10.8.1.1. Sicherer Zugriff auf einen POP3-Server

Nehmen wir an, an Ihrer Arbeitsstelle gibt es einen SSH-Server, der Verbindungen von außen akzeptiert. Auf dem Netzwerk Ihrer Arbeitsstelle soll sich zudem noch ein Mail-Server befinden, der POP3 spricht. Das Netzwerk oder die Verbindung von Ihrem Haus zu Ihrer Arbeitsstelle ist unsicher und daher müssen Sie Ihre E-Mail über eine gesicherte Verbindung abholen können. Die Lösung zu diesem Problem besteht darin, eine SSH-Verbindung von Ihrem Haus zu dem SSH-Server an Ihrer Arbeitsstelle aufzubauen, und von dort weiter zum Mail-Server zu tunneln.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

Wenn Sie den Tunnel eingerichtet haben, konfigurieren Sie Ihren Mail-Client so, dass er POP3 Anfragen zu `localhost` Port 2110 sendet. Die Verbindung wird dann sicher zu `mail.example.com` weitergeleitet.

15.10.8.1.2. Umgehen einer strengen Firewall

Einige Netzwerkadministratoren stellen sehr drakonische Firewall-Regeln auf, die nicht nur einkommende Verbindungen filtern, sondern auch ausgehende. Es kann sein, dass Sie externe Maschinen nur über die Ports 22 und 80 (SSH und Web) erreichen.

Sie wollen auf einen Dienst, der vielleicht nichts mit Ihrer Arbeit zu tun hat, wie einen Ogg Vorbis Musik-Server, zugreifen. Wenn der Ogg Vorbis Server nicht auf den Ports 22 oder 80 läuft, können Sie aber nicht auf ihn zugreifen.

Die Lösung hier ist es, eine SSH-Verbindung zu einer Maschine außerhalb der Firewall aufzumachen und durch diese zum Ogg Vorbis Server zu tunneln.

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *******
```

Konfigurieren Sie Ihren Client so, dass er `localhost` und Port 8888 benutzt. Die Verbindung wird dann zu `music.example.com` Port 8000 weitergeleitet und Sie haben die Firewall erfolgreich umgangen.

15.10.9. Die Option `AllowUsers`

Es ist in der Regel ein gute Idee, festzulegen, welche Benutzer sich von welchem Rechner aus anmelden können. Dies lässt sich beispielsweise über die Option `AllowUsers` festlegen. Soll sich etwa nur `root` vom Rechner mit der IP-Adresse `192.168.1.32` aus einwählen dürfen, würden Sie folgenden Eintrag in `/etc/ssh/sshd_config` aufnehmen:

```
AllowUsers root@192.168.1.32
```

Damit sich `admin` von jedem Rechner aus anmelden kann, geben Sie nur den Benutzernamen an:

```
AllowUsers admin
```

Sie können auch mehrere Benutzer in einer Zeile aufführen:

```
AllowUsers root@192.168.1.32 admin
```

> **Anmerkung:** Nur ein Benutzer, der in dieser Liste aufgeführt ist, darf sich auf diesem Rechner anmelden.

Nachdem Sie `/etc/ssh/sshd_config` angepasst haben, muss sshd(8) seine Konfigurationsdateien neu einlesen. Dazu geben Sie Folgendes ein:

```
# /etc/rc.d/sshd reload
```

15.10.10. Weiterführende Informationen

OpenSSH (http://www.openssh.com/)

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1) ssh_config(5)

sshd(8) sftp-server(8) sshd_config(5)

15.11. Zugriffskontrolllisten für Dateisysteme

Beigetragen von Tom Rhodes.

Zusammen mit anderen Verbesserungen des Dateisystems wie Schnappschüsse bietet FreeBSD auch *Zugriffskontrolllisten* (*access control list*, ACL).

Zugriffskontrolllisten erweitern die normalen Zugriffsrechte von UNIX Systemen auf eine kompatible (POSIX.1e) Weise und bieten feiner granulierte Sicherheitsmechanismen.

Zugriffskontrolllisten für Dateisysteme werden mit der nachstehenden Zeile in der Kernelkonfiguration aktiviert:

```
options UFS_ACL
```

Diese Option ist in der `GENERIC`-Konfiguration aktiviert. Das System gibt eine Warnung aus, wenn ein Dateisystem mit ACLs eingehangen werden soll und die Unterstützung für ACLs nicht im Kernel aktiviert ist. Das Dateisystem muss weiterhin erweiterte Attribute zur Verfügung stellen, damit ACLs verwendet werden können. Das neue UNIX-Dateisystem UFS2 stellt diese Attribute standardmäßig zur Verfügung.

> **Anmerkung:** Die Konfiguration erweiterter Attribute auf UFS1 ist mit einem höheren Aufwand als die Konfiguration erweiterter Attribute auf UFS2 verbunden. Zudem ist UFS2 mit erweiterten Attributen leistungsfähiger als UFS1. Zugriffskontrolllisten sollten daher mit UFS2 verwendet werden.

Die Angabe der Option `acl` in `/etc/fstab` aktiviert Zugriffskontrolllisten für ein Dateisystem. Die bevorzugte Möglichkeit ist die Verwendung von Zugriffskontrolllisten mit tunefs(8) (Option `-a`), im Superblock des Dateisystems festzuschreiben. Diese Möglichkeit hat mehrere Vorteile:

- Nochmaliges Einhängen eines Dateisystems (Option `-u` von mount(8)) verändert den Status der Zugriffskontrolllisten nicht. Die Verwendung von Zugriffskontrolllisten kann nur durch Abhängen und erneutes Einhängen eines Dateisystems verändert werden. Das heißt auch, dass Zugriffskontrolllisten nicht nachträglich auf dem Root-Dateisystem aktiviert werden können.
- Die Zugriffskontrolllisten auf den Dateisystemen sind, unabhängig von den Option in `/etc/fstab` oder Namensänderungen der Geräte, immer aktiv. Dies verhindert auch, dass Zugriffskontrolllisten aus Versehen auf Dateisystem ohne Zugriffskontrolllisten aktiviert werden und durch falsche Zugriffsrechte Sicherheitsprobleme entstehen.

> **Anmerkung:** Es kann sein, dass sich der Status von Zugriffskontrolllisten später durch nochmaliges Einhängen des Dateisystems (Option `-u` von mount(8)) ändern lässt. Die momentane Variante ist aber sicherer, da der Status der Zugriffskontrolllisten nicht versehentlich geändert werden kann. Allgemein sollten Zugriffskontrolllisten auf einem Dateisystem, auf dem sie einmal verwendet wurden, nicht deaktiviert werden, da danach die Zugriffsrechte falsch sein können. Werden Zugriffskontrolllisten auf einem solchen Dateisystem wieder aktiviert,

werden die Zugriffsrechte von Dateien, die sich zwischenzeitlich geändert haben, überschrieben, was zu erneuten Problemen führt.

Die Zugriffsrechte einer Datei werden durch ein + (Plus) gekennzeichnet, wenn die Datei durch Zugriffskontrolllisten geschützt ist:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Die Verzeichnisse `directory1`, `directory2` und `directory3` sind durch Zugriffskontrolllisten geschützt, das Verzeichnis `public_html` nicht.

15.11.1. Zugriffskontrolllisten benutzen

Das Werkzeug getfacl(1) zeigt Zugriffskontrolllisten an. Das folgende Kommando zeigt die ACLs auf der Datei `test`:

```
% getfacl test
        #file:test
        #owner:1001
        #group:1001
        user::rw-
        group::r--
        other::r--
```

Das Werkzeug setfacl(1) ändert oder entfernt ACLs auf Dateien. Zum Beispiel:

```
% setfacl -k test
```

Die Option `-k` entfernt alle ACLs einer Datei oder eines Dateisystems. Besser wäre es, die Option `-b` zu verwenden, da sie die erforderlichen Felder beibehält.

```
% setfacl -m u:trhodes:rwx,g:web:r--,o::--- test
```

Mit dem vorstehenden Kommando werden die eben entfernten Zugriffskontrolllisten wiederhergestellt. Der Befehl gibt die Fehlermeldung `Invalid argument` aus, wenn Sie nicht existierende Benutzer oder Gruppen als Parameter angeben.

15.12. Sicherheitsprobleme in Software Dritter überwachen

Beigetragen von Tom Rhodes.

In den letzten Jahren wurden zahlreiche Verbesserungen in der Einschätzung und dem Umgang mit Sicherheitsproblemen erzielt. Die Gefahr von Einbrüchen in ein System wird aber immer größer, da Softwarepakete von Dritten auf nahezu jedem Betriebssystem installiert und konfiguriert werden.

Die Einschätzung der Verletzlichkeit eines Systems ist ein Schlüsselfaktor für dessen Sicherheit. FreeBSD veröffentlicht zwar Sicherheitshinweise (*security advisories*) für das Basissystem, das Projekt ist allerdings nicht dazu in der Lage, dies auch für die zahlreichen Softwarepakete von Dritten zu tun. Dennoch gibt es einen Weg, auch diese Programmpakete zu überwachen. Das in der Ports-Sammlung enthaltene Programm **Portaudit** wurde gezielt dafür entwickelt.

Der Port `ports-mgmt/portaudit` fragt dazu eine Datenbank, die vom FreeBSD Security Team sowie den Ports-Entwicklern aktualisiert und gewartet wird, auf bekannte Sicherheitsprobleme ab.

Bevor Sie **Portaudit** verwenden können, müssen Sie es über die Ports-Sammlung installieren:

```
# cd /usr/ports/security/portaudit && make install clean
```

Während der Installation werden die Konfigurationsdateien für periodic(8) aktualisiert, was es **Portaudit** erlaubt, seine Ausgabe in den täglichen Sicherheitsbericht einzufügen. Stellen Sie auf jeden Fall sicher, dass diese (an das E-Mail-Konto von `root` gesendeten) Sicherheitsberichte auch gelesen werden. An dieser Stelle ist keine weitere Konfiguration nötig.

Nach der Installation kann ein Administrator die unter `/var/db/portaudit` lokal gespeicherte Datenbank aktualisieren und sich danach durch folgenden Befehl über mögliche Sicherheitslücken der von ihm installierten Softwarepakete informieren:

```
# portaudit -Fda
```

> **Anmerkung:** Die Datenbank wird automatisch aktualisiert, wenn periodic(8) ausgeführt wird. Der eben genannte Befehl ist daher optional, er wird aber für das folgende Beispiel benötigt.

Nach erfolgter Installation der Datenbank kann ein Administrator über die Ports-Sammlung installierte Softwarepakete Dritter jederzeit überprüfen. Dazu muss er lediglich folgenden Befehl eingeben:

```
# portaudit -a
```

Existiert in Ihren installierten Softwarepaketen eine Sicherheitslücke, wird **Portaudit** eine Ausgabe ähnlich der folgenden produzieren:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <http://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-0001020eed82.html>

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Wenn Sie die angegebene URL über einen Internetbrowser aufrufen, erhalten Sie weitere Informationen über die bestehende Sicherheitslücke, wie die betroffenen Versionen, die Version des FreeBSD-Ports sowie Hinweise auf weitere Seiten, die ebenfalls Sicherheitshinweise zu diesem Problem bieten.

Portaudit ist ein mächtiges Werkzeug und insbesondere in Zusammenarbeit mit dem Port **Portupgrade** äußerst hilfreich.

15.13. FreeBSD Sicherheitshinweise

Beigesteuert von Tom Rhodes.

Wie für andere hochwertige Betriebssysteme auch werden für FreeBSD Sicherheitshinweise herausgegeben. Die Hinweise werden gewöhnlich auf den Sicherheits-Mailinglisten und in den Errata veröffentlicht, nachdem das Sicherheitsproblem behoben ist. Dieser Abschnitt beschreibt den Umgang mit den Sicherheitshinweisen.

15.13.1. Wie sieht ein Sicherheitshinweis aus?

Der nachstehende Sicherheitshinweis stammt von der Mailingliste freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications):

```
=============================================================================
FreeBSD-SA-XX:XX.UTIL                                          Security Advisory
                                                               The FreeBSD Project

Topic:          denial of service due to some problem❶

Category:       core❷
Module:         sys❸
Announced:      2003-09-23❹
Credits:        Person❺
Affects:        All releases of FreeBSD❻
                FreeBSD 4-STABLE prior to the correction date
Corrected:      2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
                2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
                2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
                2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
                2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
                2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
                2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
                2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
                2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39)❼
CVE Name:       CVE-XXXX-XXXX❽

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit
http://www.FreeBSD.org/security/.

I.   Background❾


II.  Problem Description(10)


III. Impact(11)


IV.  Workaround(12)
```

```
V.   Solution(13)

VI.  Correction details(14)

VII. References(15)
```

❶ Das Feld `Topic` enthält eine Beschreibung des Sicherheitsproblems und benennt das betroffene Programm.

❷ Das Feld `Category` beschreibt den betroffenen Systemteil. Mögliche Werte für dieses Feld sind `core`, `contrib` oder `ports`. Die Kategorie `core` gilt für Kernkomponenten des FreeBSD-Betriebssystems, die Kategorie `contrib` beschreibt zum Basissystem gehörende Software Dritter beispielsweise **sendmail**. Die Kategorie `ports` beschreibt Software, die Teil der Ports-Sammlung ist.

❸ Das Feld `Module` beschreibt die betroffene Komponente. Im Beispiel ist `sys` angegeben, das heißt dieses Problem betrifft eine Komponente, die vom Kernel benutzt wird.

❹ Das Feld `Announced` gibt den Zeitpunkt der Bekanntgabe des Sicherheitshinweises an. Damit existiert das Sicherheitsproblem, ist vom Sicherheits-Team bestätigt worden und eine entsprechende Korrektur wurde in das Quellcode-Repository von FreeBSD gestellt.

❺ Das Feld `Credits` gibt die Person oder Organisation an, die das Sicherheitsproblem bemerkte und gemeldet hat.

❻ Welche FreeBSD-Releases betroffen sind, ist im Feld `Affects` angegeben. Die Version einer Datei, die zum Kernel gehört, können Sie schnell mit `ident` ermitteln. Bei Ports ist die Versionsnummer angegeben, die Sie im Verzeichnis `/var/db/pkg` finden. Wenn Sie Ihr System nicht täglich aktualisieren, ist Ihr System wahrscheinlich betroffen.

❼ Wann das Problem in welchem Release behoben wurde, steht im Feld `Corrected`.

❽ Reserviert für Informationen, über die in der *Common Vulnerabilities Database* nach Sicherheitslücken gesucht werden kann.

❾ Im Feld `Background` wird das betroffene Werkzeug beschrieben. Meist finden Sie hier warum das Werkzeug Bestandteil von FreeBSD ist, wofür es benutzt wird und eine kurze Darstellung der Herkunft des Werkzeugs.

(10) Im Feld `Problem Description` befindet sich eine genaue Darstellung des Sicherheitsproblems. Hier wird fehlerhafter Code beschrieben oder geschildert, wie ein Werkzeug ausgenutzt wird.

(11) Das Feld `Impact` beschreibt die Auswirkungen des Sicherheitsproblems auf ein System, beispielsweise erweiterte Rechte oder gar Superuser-Rechte für normale Benutzer.

(12) Im Feld `Workaround` wird eine Umgehung des Sicherheitsproblems beschrieben. Die Umgehung ist für Administratoren gedacht, die ihr System aus Zeitnot, Netzwerk-technischen oder anderen Gründen nicht aktualisieren können. Nehmen Sie Sicherheitsprobleme ernst: Auf einem betroffenen System sollte das Problem entweder behoben oder, wie hier beschrieben, umgangen werden.

(13) Im Feld `Solution` enthält eine getestete Schritt-für-Schritt Anleitung, die das Sicherheitsproblem behebt.

(14) Das Feld `Correction Details` enthält die CVS-Tags der betroffenen Dateien zusammen mit zugehörigen Revisionsnummern.

(15) Im Feld `References` finden sich Verweise auf weitere Informationsquellen. Dies können URLs zu Webseiten, Bücher, Mailinglisten und Newsgroups sein.

15.14. Prozess-Überwachung

Beigetragen von Tom Rhodes.

Prozess-Überwachung (*Process accounting*) ist ein Sicherheitsverfahren, bei dem ein Administrator verfolgt, welche Systemressourcen verwendet werden und wie sich diese auf die einzelnen Anwender verteilen. Dadurch kann das System überwacht werden und es ist sogar möglich, zu kontrollieren, welche Befehle ein Anwender eingibt.

Diese Fähigkeiten haben sowohl Vor- als auch Nachteile. Positiv ist, dass man einen Einbruchsversuch bis an den Anfang zurückverfolgen kann. Von Nachteil ist allerdings, dass durch diesen Prozess Unmengen an Protokolldateien erzeugt werden, die auch dementsprechenden Plattenplatz benötigen. Dieser Abschnitt beschreibt die Grundlagen der Prozess-Überwachung.

15.14.1. Die Prozess-Überwachung aktivieren und konfigurieren

Bevor Sie die Prozess-Überwachung verwenden können, müssen Sie diese aktivieren. Dazu führen Sie als `root` die folgenden Befehle aus:

```
# touch /var/account/acct
# accton /var/account/acct
# echo 'accounting_enable="YES"' >> /etc/rc.conf
```

Einmal aktiviert, wird sofort mit der Überwachung von CPU-Statistiken, Befehlen und anderen Vorgängen begonnen. Protokolldateien werden in einem nur von Maschinen lesbaren Format gespeichert, daher müssen Sie diese über sa(8) aufrufen. Geben Sie keine Optionen an, gibt `sa` Informationen wie die Anzahl der Aufrufe pro Anwender, die abgelaufene Zeit in Minuten, die gesamte CPU- und Anwenderzeit in Minuten, die durchschnittliche Anzahl der Ein- und Ausgabeoperationen und viel andere mehr aus.

Um Informationen über ausgeführte Befehle zu erhalten, verwenden Sie lastcomm(1). So können Sie etwa ermittlen, welche Befehle von wem auf welchem ttys(5) ausgeführt wurden:

```
# lastcomm ls
 trhodes ttyp1
```

Das Ergebnis sind alle bekannten Einsätze von `ls` durch `trhodes` auf dem Terminal `ttyp1`.

Zahlreiche weitere nützliche Optionen finden Sie in den Manualpages zu lastcomm(1), acct(5) sowie sa(8).

Fußnoten

1. Unter FreeBSD darf das System-Passwort maximal 128 Zeichen lang sein.

Kapitel 16. Jails

Beigetragen von Matteo Riondato. Übersetzt von Oliver Peter, Dirk Arlt und Johann Kois.

16.1. Übersicht

Dieses Kapitel erklärt, was FreeBSD-Jails sind und wie man sie einsetzt. Jails, manchmal als Ersatz für *chroot-Umgebungen* bezeichnet, sind ein sehr mächtiges Werkzeug für Systemadministratoren, jedoch kann deren grundlegende Verwendung auch für fortgeschrittene Anwender nützlich sein.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Wissen, was eine Jail ist und welche Verwendungszwecke es dafür unter FreeBSD gibt.
- Wissen, wie man eine Jail erstellt, startet und und anhält.
- Die Grundlagen der Jail-Administration (sowohl innerhalb als auch ausserhalb des Jails) kennen.

Weitere nützliche Informationen über Jails sind beispielsweise in folgenden Quellen zu finden:

- Der jail(8) Manualpage. Diese umfassende Referenz beschreibt, wie man unter FreeBSD eine Jail startet, anhält und kontrolliert.
- Den Mailinglisten und deren Archive. Die Archive der Mailingliste FreeBSD general questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) und anderen Mailinglisten, welche vom FreeBSD list server (http://lists.FreeBSD.org/mailman/listinfo) bereitgestellt werden, beinhalten bereits umfangreiche Informationen zu Jails. Daher ist es sinnvoll, bei Problemen mit Jails zuerst die Archive der Mailinglisten zu durchsuchen, bevor Sie eine neue Anfrage auf der Mailingliste freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) stellen.

16.2. Jails - Definitionen

Um die für den Einsatz von Jails benötigten FreeBSD-Funktionen, deren Interna sowie die Art und Weise, mit der diese mit anderen Teilen des Betriebssystems interagieren, zu erläutern, werden in diesem Kapitel folgende Definitionen verwendet:

chroot(8) (-Befehl)

: Ein Werkzeug, das den FreeBSD-Systemaufruf chroot(2) verwendet, um das Wurzelverzeichnis eines Prozesses und all seiner Nachkömmlinge zu ändern.

chroot(2) (-Umgebung)

: Die Umgebung eines Prozesses, der in einem “chroot” läuft. Diese beinhaltet Ressourcen, wie zum Beispiel sichtbare Abschnitte des Dateisystems, verfügbare Benutzer- und Gruppenkennungen, Netzwerkschnittstellen und weitere IPC-Mechanismen und so weiter.

jail(8) (-Befehl)

: Das Systemadministrationswerkzeug, welches es erlaubt, Prozesse innerhalb der Jail-Umgebung zu starten.

Host (-Benutzer, -Prozess, -System)

Das verwaltende System einer Jail-Umgebung. Das Host-System hat Zugriff auf alle verfügbaren Hardwareressourcen und kann sowohl innerhalb als auch ausserhalb der Jail-Umgebung Prozesse steuern. Einer der wichtigsten Unterschiede des Host-System einer Jails ist, dass die Einschränkungen, welche für die Superuser-Prozesse innerhalb eines Jails gelten, nicht für die Prozesse des Host-Systems gelten.

Gast (-Benutzer, -Prozess, -System)

Ein Prozess, ein Benutzer oder eine andere Instanz, deren Zugriff durch eine FreeBSD-Jail eingeschränkt ist.

16.3. Einführung

Da die Systemadministration oft eine schwierige Aufgabe ist, wurden viele mächtige Werkzeuge entwickelt, die Administratoren bei Installation, Konfiguration und Wartung ihrer Systeme unterstützen sollen. Eine wichtige Aufgabe eines Administrators ist es, Systeme so abzusichern, dass es im regulären Betrieb zu keinen Sicherheitsverstößen kommt.

Eines der Werkzeuge, mit dem die Sicherheit eines FreeBSD-Systems verbessert werden kann, sind Jails. Jails wurden schon in FreeBSD 4.X von Poul-Henning Kamp eingeführt, wurden jedoch mit FreeBSD 5.X stark verbessert, sodass sie inzwischen zu einem mächtigen und flexiblen Subsystem herangereift sind. Trotzdem geht die Entwicklung nach wie vor weiter. Wichtige Ziele sind derzeit: Bessere Zweckmäßigkeit, Leistung, Ausfallsicherheit und allgemeine Sicherheit.

16.3.1. Was ist eine Jail?

BSD-ähnliche Betriebssysteme besitzen seit den Zeiten von 4.2BSD chroot(2). Das Werkzeug chroot(2) kann dazu benutzt werden, das root-Verzeichnis einer Reihe von Prozessen zu ändern, um so eine seperate sichere Umgebung (abgeschnitten vom Rest des Systems) zu schaffen. Prozesse, die in einer chroot-Umgebung erstellt wurden, können nicht auf Dateien oder Ressourcen zugreifen, die sich ausserhalb der Umgebung befinden. Dadurch ist es einem kompromittierten Dienst nicht möglich, das gesamte System zu kompromittieren. chroot(8) eignet sich für einfache Aufgaben, die keine flexiblen, komplexen oder fortgeschrittenen Funktionen benötigen. Obwohl seit der Entwicklung des chroot-Konzepts zahlreiche Sicherheitslöcher geschlossen wurden, die es einem Prozess erlauben konnten, aus einer Jail auszubrechen, war seit langer Zeit klar, dass chroot(2) nicht die ideale Lösung ist, einen Dienst sicher zu machen.

Dies ist einer der Hauptgründe, warum *Jails* entwickelt wurden.

Jails setzen auf dem traditionellen chroot(2)-Konzept auf und verbessern es auf unterschiedlichste Art und Weise. In einer traditionellen chroot(2)-Umgebung sind Prozesse auf den Bereich des Dateisystems beschränkt, auf den sie zugreifen können. Der Rest der Systemressourcen (wie zum Beispiel eine Reihe von Systembenutzern, die laufenden Prozesse oder das Netzwerk-Subsystem) teilen sich die chroot-Prozesse mit dem Host-System. Jails dehnen dieses Modell nicht nur auf die Virtualisierung des Zugriffs auf das Dateisystem, sondern auch auf eine Reihe von Benutzern, das Netzwerk-Subsystem des FreeBSD-Kernels und weitere Bereiche aus. Eine ausführlichere Übersicht der ausgefeilten Bedienelemente zur Konfiguration einer Jail-Umgebung finden Sie im Abschnitt Abschnitt 16.5 des Handbuchs.

Eine Jail zeichnet sich durch folgende Merkmale aus:

- Einen Unterverzeichnisbaum, der die Jail enthält. Einem Prozess, der innerhalb der Jail läuft, ist es nicht mehr möglich, aus diesem auszubrechen. Von der traditionellen chroot(2)-Umgebung bekannte Sicherheitsprobleme existieren bei FreeBSD-Jails nicht mehr.
- Einen Hostname, der innerhalb der Jail verwendet wird. Jails werden vor allem dazu verwendet, Netzwerkdienste anzubieten, daher ist es für Systemadministratoren von großem Nutzen, dass jede Jail einen beschreibenden Hostname haben kann.
- Eine IP Adresse, die der Jail zugewiesen wird und nicht verändert werden kann, solange das Jail läuft. Die IP-Adresse einer Jails ist üblicherweise ein Adress-Alias auf eine existierende Netzwerkschnittstelle. Dies ist jedoch nicht zwingend erforderlich.
- Einen Befehl (genauer den Pfad einer ausführbaren Datei) der innerhalb der Jail ausgeführt werden soll. Dieser Pfad wird relativ zum root-Verzeichnis einer Jail-Umgebung angegeben und kann sehr unterschiedlich aussehen (je nachdem, wie die Jail-Umgebung konfiguriert wurde).

Unabhängig davon können Jails eine Reihe eigener Benutzer und einen eigenen Benutzer `root` haben. Selbstverständlich sind die Rechte des Benutzers `root` nur auf die Jail-Umgebung beschränkt. Aus der Sicht des Host-Systems ist der Benutzer `root` der Jail-Umgebung kein allmächtiger Benutzer, da der Benutzer `root` der Jail-Umgebung nicht dazu berechtigt ist, kritische Operationen am System ausserhalb der angebundenen jail(8)-Umgebung durchzuführen. Weitere Informationen über die Einsatzmöglichkeiten und Beschränkungen des Benutzers `root` werden im Abschnitt Abschnitt 16.5 des Handbuchs besprochen.

16.4. Einrichtung und Verwaltung von Jails

Einige Administratoren unterscheiden zwei verschiedene Jail-Arten: “Komplette” Jails, die ein echtes FreeBSD darstellen und Jails für einen bestimmten “Dienst”, die nur einer bestimmten Anwendung oder einem Dienst (der möglicherweise mit besonderen Privilegien laufen soll) gewidmet sind. Dies ist aber nur eine konzeptuelle Unterscheidung, die Einrichtung einer Jail bleibt davon gänzlich unberührt.

```
# setenv D /hier/ist/die/jail
# mkdir -p $D ❶
# cd /usr/src
# make buildworld ❷
# make installworld DESTDIR=$D ❸
# make distribution DESTDIR=$D ❹
# mount -t devfs devfs $D/dev ❺
```

❶ Das Festlegen des Installationsorts für das Jail eignet sich am besten als Startpunkt. Hier wird sich die Jail innerhalb des Host-Dateisystems befinden. Eine gute Möglichkeit wäre etwa `/usr/jail/`*`name_der_jail`*, wobei *`name_der_jail`* den Hostname darstellt, über den die Jail identifiziert werden soll. Das Dateisystem unterhalb von `/usr/` stellt normalerweise aussreichend Platz für eine Jail zur Verfügung (bedenken Sie, dass eine “komplette” Jail ein Replikat einer jeden Datei der Standardinstallation des FreeBSD-Basissystems enthält.

❷ Wenn Sie bereits ihre Systemanwendungen mittels `make world` oder `make buildworld` neu erstellt haben, können Sie diesen Schritt überspringen und die Systemanwendungen in die neue Jail installieren.

❸ Dieser Befehl wird den Verzeichnisbaum mit allen notwendigen Binärdateien, Bibliotheken, Manualpages usw. erstellen.

❹ Der `distribution`-Befehl lässt **make** alle benötigten Konfigurationsdateien installieren, es werden also alle installierbaren Dateien aus `/usr/src/etc/` in das Verzeichnis `/etc` der Jail installiert (also nach `$D/etc/`).

❺ Das Einhängen des devfs(8)-Dateisystems innerhalb der Jail ist nicht unbedingt notwendig. Allerdings benötigt fast jede Anwendung Zugriff auf wenigstens ein Gerät. Es ist daher sehr wichtig, den Zugriff auf Devices aus der Jail heraus zu kontrollieren, da unsaubere Einstellungen es einem Angreifer erlauben könnten, in das System einzudringen. Die Kontrolle über devfs(8) erfolgt durch die in den Manualpages devfs(8) und devfs.conf(5) beschriebenen Regeln.

Ist eine Jail einmal erst erstellt, kann sie durch jail(8) gestartet werden. jail(8) benötigt zwingend mindestens vier Argumente, die im Abschnitt Abschnitt 16.3.1 des Handbuchs beschrieben sind. Weitere Argumente sind möglich, um beispielsweise die Jail mit den Berechtigungen eines bestimmten Benutzers laufen zu lassen. Das Argument *`command`* hängt vom Typ der Jail ab; für ein *virtuelles System* ist `/etc/rc` eine gute Wahl, da dies dem Startvorgang eines echten FreeBSD-Systems entspricht. Bei einer *Service*-Jail hängt dieses von der Art des Dienstes ab, der in der Jail laufen soll.

Jails werden häufig mit dem Betriebssystem gestartet, da der `rc`-Mechanismus von FreeBSD dafür eine einfach zu realisierende Möglichkeit bietet.

1. Eine Liste der Jails, die mit dem Betriebssystem gestartet werden sollen, wird in die Datei rc.conf(5) geschrieben:

   ```
   jail_enable="YES"   # Set to NO to disable starting of any jails
   jail_list="www"     # Space separated list of names of jails
   ```

   Anmerkung: Die Namen der Jails in der `jail_list` sollten nur alphanumerische Zeichen enthalten.

2. Für jede Jail in der `jail_list` sollten in rc.conf(5) einige Einstellungen vorgenommen werden:

   ```
   jail_www_rootdir="/usr/jail/www"     # jail's root directory
   jail_www_hostname="www.example.org"  # jail's hostname
   jail_www_ip="192.168.0.10"           # jail's IP address
   jail_www_devfs_enable="YES"          # mount devfs in the jail
   jail_www_devfs_ruleset="www_ruleset" # devfs ruleset to apply to jail
   ```

   Beim Start einer in rc.conf(5) konfigurierten Jail wird das `/etc/rc`-Skript der Jail (das "annimmt", dass es sich in einem kompletten System befindet) aufgerufen. Für Service-Jails sollten die Startskripte der Jail durch das Setzen der Option `jail_`*`jailname`*`_exec_start` entsprechend angepasst werden.

   Anmerkung: Eine vollständige Liste der Optionen findet sich in der Manualpage zu rc.conf(5).

Das `/etc/rc.d/jail`-Skript kann zum manuellen Starten und Stoppen der Jail genutzt werden, wenn ein Eintrag in `rc.conf` angelegt wurde:

```
# /etc/rc.d/jail start www
# /etc/rc.d/jail stop www
```

Es gibt momentan keinen sauberen Weg, eine jail(8) zu stoppen. Dies liegt daran, dass die Kommandos zum sauberen Herunterfahren eines Systems innerhalb einer Jail nicht ausgeführt werden können. Der beste Weg eine Jail

zu beenden ist es daher, innerhalb der Jail den folgenden Befehl auszuführen (alternativ können Sie auch jexec(8) von außerhalb der Jail aufrufen):

```
# sh /etc/rc.shutdown
```

Weitere Informationen zu diesem Thema finden Sie in der Manualpage jail(8).

16.5. Feinabstimmung und Administration

Es gibt verschiedene Optionen, die für jede Jail gesetzt werden können und verschiedene Wege, ein FreeBSD-Host-System mit Jails zu kombinieren. Dieser Abschnitt zeigt Ihnen:

- Einige zur Verfügung stehende Optionen zur Abstimmung des Verhaltens und der Sicherheitseinstellungen, die mit einer Jail-Installation ausgeführt werden können.
- Einige der Anwendungsprogramme für das Jail-Management, die über die FreeBSD Ports-Sammlung verfügbar sind und genutzt werden können, um Jail-basierte Lösungen allumfassend umzusetzen.

16.5.1. Systemwerkzeuge zur Feinabstimmung von Jails in FreeBSD

Die Feinabstimmung einer Jail-Konfiguration erfolgt zum Großteil durch das Setzen von sysctl(8)-Variablen. Es gibt einen speziellen sysctl-Zweig, der als Basis für die Organisation aller relevanten Optionen dient: Die `security.jail.*`-Hierarchie der FreeBSD-Kerneloptionen. Die folgende Liste enthält alle jail-bezogenen sysctls (inklusiver ihrer Voreinstellungen). Die Namen sollten selbsterklärend sein, für weitergehende Informationen lesen Sie bitte die Manualpages jail(8) und sysctl(8).

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 0`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

Diese Variablen können vom Administrator des *Host-Systems* genutzt werden, um Beschränkungen hinzuzufügen oder aufzuheben, die dem Benutzer `root` als Vorgabe auferlegt sind. Beachten Sie, dass es einige Beschränkungen gibt, die nicht verändert werden können. Der Benutzer `root` darf innheralb der jail(8) keine Dateisysteme mounten und unmounten. Ebenso ist es ihm untersagt, das devfs(8)-Regelwerk zu laden oder zu entladen. Er darf weder Firewallregeln setzen, noch administrative Aufgaben erledigen, die Modifikationen am Kernel selbst erfordern (wie bespielsweise das Setzen des `Securelevels` des Kernel.

Das FreeBSD-Basissystem enthält einen Basissatz an Werkzeugen, um Informationen über aktive Jails zu erlangen und einer Jail administrative Befehle zuzuordnen. Die Befehle jls(8) und jexec(8) sind Teil des FreeBSD-Basissystems und können für folgende Aufgaben verwendet werden:

- Das Anzeigen einer Liste der aktiven Jails und ihrer zugehörigen Jail Identifier (JID), ihrer IP-Addresse, ihres Hostnames und ihres Pfades.
- Das Herstellen einer Verbindung mit einer laufenden Jail, das Starten eines Befehls aus dem gastgebenen System heraus oder das Ausführen einer administrativen Aufgabe innerhalb der Jail selbst. Dies ist insbesondere dann nützlich, wenn der Benutzer `root` die Jail sauber herunterfahren möchte. jexec(8) kann auch zum Starten einer Shell innerhalb der Jail genutzt werden, um adminstrative Aufgaben durchzuführen:

  ```
  # jexec 1 tcsh
  ```

16.5.2. High-Level-Werkzeuge zur Jail-Administration in der FreeBSD Ports-Sammlung

Unter den zahlreichen Fremdwerkzeugen für die Administration von Jails sind die `sysutils/jailutils` die vollständigsten und brauchbarsten. Dabei handelt es sich um eine Sammlung kleiner Anwendungen, die das jail(8)-Management vereinfachen. Weitere Informationen zu diesen Werkzeugen finden Sie auf den entsprechenden Internetseiten.

16.6. Anwendung von Jails

16.6.1. Service-Jails

Beigetragen von Daniel Gerzo.

Dieser Abschnitt basiert auf einer von Simon L. B. Nielsen auf http://simon.nitro.dk/service-jails.html präsentierten Idee und einem aktualisierten Artikel von Ken Tom (<`locals@gmail.com`>). Er beschreibt, wie ein FreeBSD-System durch Benutzung der jail(8)-Funktion mit zusätzlichen Sicherheitsebenen ausgestattet werden kann. Es wird dabei angenommen, dass auf Ihrem FreeBSD-System RELENG_6_0 oder neuer installiert ist und dass Sie die Informationen aus den vorangehenden Abschnitten gelesen und verstanden haben.

16.6.1.1. Design

Eines der Hauptprobleme bei Jails ist das Management ihres Upgrade-Prozesses. Dieser neigt dazu, problematisch zu sein, da jede Jail bei jedem Upgrade komplett neu gebaut werden muss. Das stellt normalerweise kein Problem dar, wenn es sich um eine einzelne Jail handelt, da der Upgrade-Prozess recht einfach ist. Verwenden Sie aber eine größere Anzahl von Jails, kann dieser Prozess sehr zeitaufwendig werden.

> **Warnung:** Diese Konfiguration erfordert fortgeschrittene Kenntnisse im Umgang mit FreeBSD sowie der Benutzung seiner Funktionen. Sollten die unten vorgestellten Schritte zu kompliziert wirken, wird empfohlen, sich einfachere Verfahren wie `sysutils/ezjail` anzusehen, da diese einfachere Methoden zur Administration von Jails verwenden und daher nicht so anspruchsvoll sind wie der hier beschriebene Aufbau.

Diese Konfiguration basiert darauf, Jails so weit als möglich gemeinsam zu verwalten. Dies passiert auf sichere Art und Weise durch den Einsatz von mount_nullfs(8)-Mounts (read-only). Dadurch werden Aktualisierungen erleichtert und das Verteilen von verschiedenen Diensten auf verschiedene Jails wird attraktiver. Außerdem bietet dieses Verfahren einen einfachen Weg, Jails hinzuzufügen, zu entfernen und zu aktualisieren.

> **Anmerkung:** Beispiele für Dienste sind in diesem Zusammenhang: Ein HTTP-Server, ein DNS-Server, ein SMTP-Server und so weiter.

Die Ziele des in diesem Abschnitt beschriebenen Aufbaus sind:

- Das Erstellen einer einfachen und gut verständlichen Struktur von Jails. Dies beinhaltet, *nicht* für jede Jail ein vollständiges installworld laufen lassen zu müssen.
- Es einfach zu machen, neue Jails zu erstellen oder alte zu entfernen.
- Es einfach zu machen, bestehende Jails zu aktualisieren.
- Es einfach zu machen, einen angepassten FreeBSD-Zweig zu nutzen.
- Paranoid bezüglich Sicherheit zu sein und Angriffsmöglickeiten weitgehend zu reduzieren.
- Soviel Platz und Inodes wie möglich einzusparen.

Wie bereits erwähnt, ist dieses Design stark darauf angewiesen, dass eine read-only-Hauptvorlage in jede Jail hinein gemountet wird (bekannt als **nullfs**), und dass jede Jail über wenigstens ein beschreibbares Gerät verfügt. Das Gerät kann hierbei eine separate physikalische Platte oder ein vnode unterstütztes md(4)-Gerät sein. Im folgenden Beispiel wird ein **nullfs**-Mount genutzt, auf den nur Lesezugriff erlaubt ist.

Das Layout des Dateisystems wird in der folgenden Liste beschrieben:

- Jede Jail wird unterhalb des `/home/j`-Verzeichnisses gemountet.
- `/home/j/mroot` ist die Vorlage für jede Jail und die nur lesbare Partition für alle Jails.
- Unterhalb von `/home/j` wird für jede Jail ein leeres Verzeichnis angelegt.
- Jede Jail bekommt ein `/s`-Verzeichnis, das zum read/write-Teilbereich des Systems verlinkt wird.
- Jede Jail bekommt ihr eigenes read/write-System, das auf `/home/j/skel` basiert.
- Jeder Jailbereich (genauer der read/write-Teilbereich jeder Jail) wird in `/home/js` erstellt.

> **Anmerkung:** Es wird angenommen, dass die Jails sich unterhalb des `/home` Verzeichnisses befinden. Dieser Ort kann von Ihnen natürlich geändert werden. Allerdings müssen die Pfade in den folgenden Beispielen dann entsprechend angepasst werden.

16.6.1.2. Erstellen der Vorlage

Dieser Abschnitt beschreibt die Schritte, die zum Erstellen der Hauptvorlage (die den nur lesbaren Bereich für alle weiteren Jails darstellt) notwendig sind.

Es ist immer eine gute Idee, FreeBSD auf den aktuellen -RELEASE-Zweig zu aktualisieren. Lesen Sie das entsprechende Kapitel (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/handbook/makeworld.html) des Handbuchs für Informationen zu diesem Thema. Selbst wenn Sie auf eine Aktualisierung des Betriebssystems verzichten, müssen Sie dennoch ein buildworld durchführen, um fortfahren zu können. Außerdem müssen Sie das Paket `sysutils/cpdup` installiert sein. In diesem Beispiel wird portsnap(8) verwendet, um die aktuelle FreeBSD Ports-Sammlung herunterzuladen. Der Abschnitt Portsnap

(http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/handbook/portsnap.html) des Handbuchs beschreibt, wie Sie dieses Werkzeug effektiv einsetzen.

1. Zuerst erstellen wir eine Verzeichnissstruktur für das read-only-Dateisystem, das die FreeBSD-Binärdateien für unsere Jails enthalten wird. Anschließend wechseln wir in den FreeBSD-Quellcodebaum und installieren das read-only-Dateisystem in die (Vorlage-)Jail.

   ```
   # mkdir /home/j /home/j/mroot
   # cd /usr/src
   # make installworld DESTDIR=/home/j/mroot
   ```

2. Als nächstes bereiten wir die Ports-Sammlung fü die Jails vor und kopieren den FreeBSD Quellcodebaum in die Jail, da dieser für **mergemaster** benötigt wird:

   ```
   # cd /home/j/mroot
   # mkdir usr/ports
   # portsnap -p /home/j/mroot/usr/ports fetch extract
   # cpdup /usr/src /home/j/mroot/usr/src
   ```

3. Danach wird die Struktur für den read/write-Bereich des Systems erstellt:

   ```
   # mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr-X11R6 /home/j/skel/distfiles
   # mv etc /home/j/skel
   # mv usr/local /home/j/skel/usr-local
   # mv tmp /home/j/skel
   # mv var /home/j/skel
   # mv root /home/j/skel
   ```

4. Nutzen Sie **mergemaster**, um fehlende Konfigurationsdateien zu installieren. Anschließend werden die von **mergemaster** erstellten Extra-Verzeichnisse entfernt:

   ```
   # mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
   # cd /home/j/skel
   # rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
   ```

5. Nun wird das read/write-Dateisystem mit dem read-only-Dateisystem verlinkt. Bitte vergewissern Sie sich, dass die symbolischen Links an den korrekten `s/` Positionen erstellt werden. Echte Verzeichnisse oder an falschen Positionen erstellte Verzeichnisse lassen die Installation fehlschlagen.

   ```
   # cd /home/j/mroot
   # mkdir s
   # ln -s s/etc etc
   # ln -s s/home home
   # ln -s s/root root
   # ln -s ../s/usr-local usr/local
   # ln -s ../s/usr-X11R6 usr/X11R6
   # ln -s ../../s/distfiles usr/ports/distfiles
   # ln -s s/tmp tmp
   # ln -s s/var var
   ```

6. Zuletzt erstellen Sie eine allgemeine `/home/j/skel/etc/make.conf` mit folgendem Inhalt:

   ```
   WRKDIRPREFIX?=  /s/portbuild
   ```

   Ein gesetztes `WRKDIRPREFIX` erlaubt es, die FreeBSD-Ports innerhalb jeder Jail zu kompilieren. Das Ports-Verzeichnis ist Teil des read-only System. Der angepasste Pfad des `WRKDIRPREFIX` macht es möglich, innerhalb des read/write-Bereichs der Jail Ports zu bauen.

16.6.1.3. Jails erstellen

Da nun eine komplette FreeBSD-Jailvorlage vorliegt, sind wir nun in der Lage, Jails einrichten und in `/etc/rc.conf` zu konfigurieren. Dieses Beispiel zeigt das Erstellen von drei Jails: “NS”, “MAIL” und “WWW”.

1. Fügen Sie die folgenden Zeilen in `/etc/fstab` ein, damit die read-only-Vorlage und der read/write-Bereich für alle Jails verfügbar sind:

   ```
   /home/j/mroot   /home/j/ns     nullfs  ro  0   0
   /home/j/mroot   /home/j/mail   nullfs  ro  0   0
   /home/j/mroot   /home/j/www    nullfs  ro  0   0
   /home/js/ns     /home/j/ns/s   nullfs  rw  0   0
   /home/js/mail   /home/j/mail/s nullfs  rw  0   0
   /home/js/www    /home/j/www/s  nullfs  rw  0   0
   ```

   Anmerkung: Mit der Pass-Nummer 0 markierte Partitionen werden beim Booten des Systems nicht von fsck(8) geprüft, mit 0 als Dump-Nummer markierte Partitonen werden von dump(8) nicht gesichert. Wir wollen nicht, dass **fsck** unsere **nullfs**-Mounts prüft oder dass **dump** die nur lesbaren nullfs-Mounts unserer Jails sichert. Deshalb werden diese Bereiche in den letzten beiden Spalten der obenstehenden `fstab` mit “0 0” markiert.

2. Konfigurieren Sie die Jails in `/etc/rc.conf`:

   ```
   jail_enable="YES"
   jail_set_hostname_allow="NO"
   jail_list="ns mail www"
   jail_ns_hostname="ns.example.org"
   jail_ns_ip="192.168.3.17"
   jail_ns_rootdir="/usr/home/j/ns"
   jail_ns_devfs_enable="YES"
   jail_mail_hostname="mail.example.org"
   jail_mail_ip="192.168.3.18"
   jail_mail_rootdir="/usr/home/j/mail"
   jail_mail_devfs_enable="YES"
   jail_www_hostname="www.example.org"
   jail_www_ip="62.123.43.14"
   jail_www_rootdir="/usr/home/j/www"
   jail_www_devfs_enable="YES"
   ```

   Warnung: Der Grund dafür, dass die Variablen `jail_`*`name`*`_rootdir` nach `/usr/home` statt nach `/home` zeigen, liegt darin, dass der physikalische Pfad des `/home`-Verzeichnisses unter FreeBSD `/usr/home` lautet. Die Variable `jail_`*`name`*`_rootdir` darf im Pfad aber *keinen symbolischen Link* enthalten, weil das Jail ansonsten nicht gestartet werden kann. Verwenden Sie realpath(1), um den korrekten Wert für diese Variable zu bestimmen. Weitere Informationen finden Sie im Security Advisory FreeBSD-SA-07:01.jail.

3. Erstellen Sie die notwendigen Mountpunkte für die nur lesbaren Bereiche jeder Jail:

   ```
   # mkdir /home/j/ns /home/j/mail /home/j/www
   ```

4. Installieren Sie die read/write-Vorlage in jede Jail. Benutzen Sie hierfür `sysutils/cpdup`, welches es erleichtert, eine korrekte Kopie jedes Verzeichnisses zu erstellen:

```
# mkdir /home/js
# cpdup /home/j/skel /home/js/ns
# cpdup /home/j/skel /home/js/mail
# cpdup /home/j/skel /home/js/www
```

5. An dieser Stelle werden die Jails erstellt und fü den Betrieb vorbereitet. Zuerst mounten Sie die notwendigen Dateisysteme für jede Jail und starten diese dann mit dem Skript `/etc/rc.d/jail`:

```
# mount -a
# /etc/rc.d/jail start
```

Die Jails sollten nun laufen. Um zu prüfen, ob sie korrekt gestartet wurden, verwenden Sie jls(8). Nach dem Aufruf dieses Befehls sollten Sie eine Ausgabe ähnlich der folgenden erhalten:

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.3.17    ns.example.org                /home/j/ns
     2  192.168.3.18    mail.example.org              /home/j/mail
     1  62.123.43.14    www.example.org               /home/j/www
```

An diesem Punkt sollte es möglich sein, sich an jeder Jail anzumelden, Benutzer anzulegen und Dienste zu konfigurieren. Die Spalte `JID` gibt die Jail-Identifikationsnummer jeder laufenden Jail an. Nutzen Sie den folgenden Befehl, um administrative Aufgaben in der Jail mit der `JID` 3 durchzuführen:

```
# jexec 3 tcsh
```

16.6.1.4. Jails aktualisieren

Mit der Zeit wird es notwendig sein, das System auf eine neuere Version von FreeBSD zu aktualisieren. Zum einen aus Sicherheitsgründen, zum anderen, um neu eingeführte Funktionen nutzen zu können, die für die bestehenden Jails sinnvoll sind. Das Design dieses Aufbaus bietet einen einfachen Weg, bestehende Jails zu aktualisieren. Zudem reduziert es die Downtime, da die Jails erst im allerletzten Schritt gestoppt werden müssen. Außerdem bietet es die Möglichkeit, zu älteren Versionen zurückzukehren, falls irgendwelche Probleme auftreten.

1. Im ersten Schritt wird das Host-System aktualisiert. Anschließend wird eine temporäre neue read-only Vorlage `/home/j/mroot2` erstellt.

```
# mkdir /home/j/mroot2
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot2
# cd /home/j/mroot2
# cpdup /usr/src usr/src
# mkdir s
```

   Der `installworld`-Durchlauf erzeugt einige unnötige Verzeichnisse, die nun entfernt werden sollten:

```
# chflags -R 0 var
# rm -R etc var root usr/local tmp
```

2. Erzeugen Sie neue symbolische Links für das Hauptdateisystem:

```
# ln -s s/etc etc
# ln -s s/root root
# ln -s s/home home
```

```
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
# ln -s s/tmp tmp
# ln -s s/var var
```

3. Nun ist es an der Zeit, die Jails zu stoppen:

   ```
   # /etc/rc.d/jail stop
   ```

4. Unmounten des originalen Dateisystems:

   ```
   # umount /home/j/ns/s
   # umount /home/j/ns
   # umount /home/j/mail/s
   # umount /home/j/mail
   # umount /home/j/www/s
   # umount /home/j/www
   ```

   > **Anmerkung:** Die read/write-Systeme sind an das read-only System angehängt (`/s`), das daher zuerst ausgehängt werden muss.

5. Verschieben Sie das alte read-only-Dateisystem und ersetzen Sie es durch das neue Dateisystem. Das alte Dateisystem kann so als Backup dienen, falls etwas schief geht. Die Namensgebung entspricht hier derjenigen bei der Erstellung eines neuen read-only-Dateisystems. Verschieben Sie die originale FreeBSD Ports-Sammlung in das neue Dateisystem, um Platz und Inodes zu sparen:

   ```
   # cd /home/j
   # mv mroot mroot.20060601
   # mv mroot2 mroot
   # mv mroot.20060601/usr/ports mroot/usr
   ```

6. Nun ist die neue read-only-Vorlage fertig. Sie müssen daher nur noch die Dateisysteme erneut mounten und die Jails starten:

   ```
   # mount -a
   # /etc/rc.d/jail start
   ```

Nutzen Sie jls(8) um zu prüfen, ob die Jails korrekt gestartet wurden. Vergessen Sie nicht, innerhalb jeder Jail `mergemaster` laufen zu lassen. Die Konfigurationsdateien müssen (ebenso wie die rc.d-Skripten) aktualisiert werden.

Kapitel 17. Verbindliche Zugriffskontrolle

Written by Tom Rhodes. Übersetzt von Benjamin Lukas.

17.1. Übersicht

In FreeBSD 5.X wurden neue Sicherheits-Erweiterungen verfügbar, die aus dem TrustedBSD-Projekt übernommen wurden und auf dem Entwurf POSIX.1e basieren. Die beiden bedeutendsten neuen Sicherheits-Mechanismen sind Berechtigungslisten (Access Control Lists, ACL) und die verbindliche Zugriffskontrolle (Mandatory Access Control, MAC). Durch die MAC können Module geladen werden, die neue Sicherheitsrichtlinien bereitstellen. Mit Hilfe einiger Module kann beispielsweise ein eng umgrenzter Bereich des Betriebssystems gesichert werden, indem die Sicherheitsfunktionen spezieller Dienste unterstützt bzw. verstärkt werden. Andere Module wiederum betreffen in ihrer Funktion das gesamte System - alle vorhandenen Subjekte und Objekte. Das "Verbindliche" in der Namensgebung erwächst aus dem Fakt, dass die Kontrolle allein Administratoren und dem System obliegt und nicht dem Ermessen der Nutzer, wie es mit Hilfe der benutzerbestimmbaren Zugriffskontrolle (Discrectionary Access Control / DAC), dem Zugriffstandard für Dateien, gar der System V IPC in FreeBSD, normalerweise umgesetzt wird.

Dieses Kapitel wird sich auf die Grundstruktur der Verbindlichen Zugriffskontrolle und eine Auswahl der Module, die verschiedenste Sicherheitsfunktionen zur Verfügung stellen, konzentrieren.

Beim Durcharbeiten dieses Kapitels erfahren Sie:

- Welche MAC Module für Sicherheitsrichtlinien derzeit in FreeBSD eingebettet sind und wie die entsprechenden Mechanismen funktionieren.
- Was die einzelnen MAC Module an Funktionen realisieren und auch, was der Unterschied zwischen einer Richtlinie, die *mit* Labels arbeitet, und einer, die *ohne* Labels arbeitet, ist.
- Wie Sie die MAC in ein System einbetten und effizient einrichten.
- Wie die verschiedenen Richtlinienmodule einer MAC konfiguriert werden.
- Wie mit einer MAC und den gezeigten Beispielen eine sicherere Umgebung erstellt werden kann.
- Wie die Konfiguration einer MAC auf korrekte Einrichtung getestet wird.

Vor dem Lesen dieses Kapitels sollten Sie bereits:

- Grundzüge von UNIX und FreeBSD verstanden haben. (Kapitel 4).
- Mit den Grundzügen der Kernelkonfiguration und -kompilierung vertraut sein (Kapitel 9).
- Einige Vorkenntnisse über Sicherheitskonzepte im Allgemeinen und deren Umsetzung in FreeBSD im Besonderen mitbringen (Kapitel 15).

> **Warnung:** Der unsachgemäße Gebrauch der in diesem Kapitel enthaltenen Informationen kann den Verlust des Systemzugriffs, Ärger mit Nutzern oder die Unfähigkeit, grundlegende Funktionen des X-Windows-Systems zu nutzen, verursachen. Wichtiger noch ist, dass man sich nicht allein auf die MAC verlassen sollte, um ein System zu sichern. Die MAC verbessert und ergänzt lediglich die schon existierenden Sicherheits-Richtlinien - ohne eine gründliche und fundierte Sicherheitspraxis und regelmäßige Sicherheitsprüfungen wird Ihr System nie vollständig sicher sein.
>
> Außerdem sollte angemerkt werden, dass die Beispiele in diesem Kapitel auch genau dasselbe sein sollen, nämlich Beispiele. Es wird nicht empfohlen, diese bestimmten Beispiele auf einem Arbeitssystem umzusetzen.

Das Einarbeiten der verschiedenen Sicherheitsmodule erfordert eine Menge Denkarbeit und viele Tests. Jemand, der nicht versteht, wie diese Module funktionieren, kann sich schnell darin wiederfinden, dass er (oder sie) das ganze System durchforsten und viele Dateien und Verzeichnisse neu konfigurieren muß.

17.1.1. Was in diesem Kapitel nicht behandelt wird

Dieses Kapitel behandelt einen großen Teil sicherheitsrelevanter Themen, bezogen auf die Verbindliche Zugriffskontrolle (MAC). Die gegenwärtige Entwicklung neuer MAC Module ist nicht abgedeckt. Einige weitere Module, die im MAC Framework enthalten sind, haben besondere Charakteristika, die zum Testen und Entwickeln neuer Module gedacht sind. Dies sind unter anderem mac_test(4), mac_stub(4) und mac_none(4). Für weitere Informationen zu diesen Modulen und den entsprechend angebotenen Funktionen lesen Sie bitte die Manpages.

17.2. Schlüsselbegriffe

Bevor Sie weiterlesen, müssen noch einige Schlüsselbegriffe geklärt werden. Dadurch soll jegliche auftretende Verwirrung von vornherein beseitigt und die plötzliche Einführung neuer Begriffe und Informationen vermieden werden.

- *Verbund*: Ein Verbund ist ist ein Satz von Programmen und Daten, die speziell und zusammen abgeschottet wurden, um Nutzern Zugriff auf diese ausgewiesenen Systembereiche zu gewähren. Man kann sagen, ein solcher Verbund ist eine Gruppierung, ähnlich einer Arbeitsgruppe, einer Abteilung, einem Projekt oder einem Thema. Durch die Nutzung von Verbünden (*compartments*) kann man Sicherheitsrichtlinien erstellen, die alles notwendige Wissen und alle Werkzeuge zusammenfassen.
- *Hochwassermarkierung*: Eine solche Richtlinie erlaubt die Erhöhung der Sicherheitsstufe in Abhängigkeit der Klassifikation der gesuchten bzw. bereitzustellenden Information. Normalerweise wird nach Abschluss des Prozesses die ursprüngliche Sicherheitsstufe wieder hergestellt. Derzeit enthält die MAC Grundstruktur keine Möglichkeit, eine solche Richtlinie umzusetzen, der Vollständigkeit halber ist die Definition hier jedoch aufgeführt.
- *Integrität*: Das Schlüsselkonzept zur Klassifizierung der Vertraulichkeit von Daten nennt man Integrität. Je weiter die Integrität erhöht wird, umso mehr kann man den entsprechenden Daten vertrauen.
- *Label*: Ein Label ist ein Sicherheitsmerkmal, welches mit Dateien, Verzeichnissen oder anderen Elementen im System verbunden wird. Man sollte es wie einen Vertraulichkeitsstempel auffassen, der Dateien angehört wie beispielsweise die Zugriffszeit, das Erstellungsdatum oder auch der Name; sobald Dateien derart gekennzeichnet werden, bezeichnen diese Label die sicherheitsrelevanten Eigenschaften. Zugriff ist nur noch dann möglich, wenn das zugreifende Subjekt eine korrespondierende Kennzeichnung trägt. Die Bedeutung und Verarbeitung der Label-Werte ist von der Einrichtung der Richtlinie abhängig: Während einige Richtlinien das Label zum Kennzeichnen der Vertraulichkeit oder Geheimhaltungsstufe eines Objekts nutzen, können andere Richtlinien an derselben Stelle Zugriffsregeln festschreiben.
- *Level*: Eine erhöhte oder verminderte Einstellung eines Sicherheitsmerkmals. Wenn das Level erhöht wird, wird auch die ensprechende Sicherheitsstufe angehoben.
- *Niedrigwassermarkierung*: Eine solche Richtlinie erlaubt das Herabstufen des Sicherheitslevels, um weniger sensible Daten verfügbar zu machen. In die meisten Fällen wird das ursprüngliche Sicherheitslevel des Nutzers

wiederhergestellt, sobald der Vorgang abgeschlossen ist. Das einzige Modul in FreeBSD, welches von dieser Richtlinie Gebrauch macht, ist mac_lomac(4).

- *Multilabel*: Die Eigenschaft `multilabel` ist eine Dateisystemoption, die entweder im Einzelbenutzermodus mit Hilfe des Werkzeugs tunefs(8), während des Bootvorgangs in der Datei fstab(5) oder aber beim Erstellen einen neues Dateisystems aktiviert werden kann. Diese Option erlaubt einem Administrator, verschiedenen Objekten unterschiedliche Labels zuzuordnen - kann jedoch nur zusammen mit Modulen angewendet werden, die auch tatsächlich mit Labels arbeiten.
- *Objekt*: Ein Objekt oder auch Systemobjekt ist theoretisch eine Einheit, durch welche Information fließt, und zwar unter der Lenkung eines *Subjektes*. Praktisch schliesst diese Definition Verzeichnisse, Dateien, Felder, Bildschirme, Tastaturen, Speicher, Bandlaufwerke, Drucker und jegliche anderen Datenspeicher- oder -verarbeitungsgeräte ein. Im Prinzip ist ein Objekt ein Datenkontainer oder eine Systemressource - Zugriff auf ein *Objekt* bedeutet, auf Daten zuzugreifen.
- *Richtlinie*: Eine Sammlung von Regeln, die definiert, wie Zielvorgaben umgesetzt werden, nennt man Richtlinie. Eine *Richtlinie* dokumentiert normalerweise, wie mit bestimmten Elementen umgegangen wird. Dieses Kapitel faßt den Begriff in diesem Kontext als *Sicherheitsrichtlinie* auf; als eine Sammlung von Regeln, die den Fluß von Daten und Informationen kontrolliert und die gleichzeitig definiert, wer auf diese Daten und Informationen zugreifen darf.
- *Anfälligkeit*: Dieser Begriff wird normalerweise verwendet, wenn man über MLS (Multi Level Security) spricht. Das Anfälligkeits-Level beschreibt, wie wichtig oder geheim die Daten sein sollen. Um so höher das Anfälligkeits-Level, um so wichtiger die Geheimhaltung bzw. Vertraulichkeit der Daten.
- *Einzel-Label*: Von einem Einzel-Label spricht man, wenn für ein ganzes Dateisystem lediglich ein einziges Label verwendet wird, um Zugriffskontrolle über den gesamten Datenfluss zu erzwingen. Sobald diese Option verwendet wird - und das ist zu jeder Zeit, wenn die Option `multilabel` nicht explizit gesetzt wurde - sind alle Dateien und Verzeichnisse mit dem gleichen Label gekennzeichnet.
- *Subjekt*: Ein Subjekt ist jedwede Einheit, die Information in Fluss zwischen Objekten bringt: Zum Beispiel ein Nutzer, ein Nutzerprozessor, ein Systemprozeß usw. In FreeBSD handelt es sich meistens um einen Thread, der als Prozeß im Namen eines Nutzers arbeitet.

17.3. Erläuterung

Mit all diesen neuen Begriffen im Kopf können wir nun überlegen, wie die Möglichkeiten der verbindlichen Zugriffskontrolle (MAC) die Sicherheit eines Betriebssystems als Ganzes erweitern. Die verschiedenen Module, die durch die MAC bereitgestellt werden, können verwendet werden, um das Netzwerk oder Dateisysteme zu schützen, Nutzern den Zugang zu bestimmten Ports oder Sockets zu verbieten und vieles mehr. Die vielleicht beste Weise, die Module zu verwenden, ist, sie miteinander zu kombinieren, indem mehrere Sicherheitsrichtlinienmodule gleichzeitig eine mehrschichtige Sicherheitsumgebung schaffen. Das ist etwas anderes als singuläre Richtlinien wie zum Beispiel die Firewall, die typischerweise Elemente eines Systems stabilisiert, das nur für einen speziellen Zweck verwendet wird. Der Verwaltungsmehraufwand ist jedoch von Nachteil, zum Beispiel durch die Verwendung von mehreren Labels oder dem eigenhändigen Erlauben von Netzwerkzugriffen für jeden einzelnen Nutzer.

Solche Nachteile sind allerdings gering im Vergleich zum bleibenden Effekt der erstellten Struktur. Die Möglichkeit zum Beispiel, für konkrete Anwendungen genau die passenden Richtlinien auszuwählen und einzurichten, senkt gleichzeitig die Arbeitskosten. Wenn man unnötige Richtlinien aussortiert, kann man die Gesamtleistung des Systems genauso steigern wie auch eine höhere Anpassungsfähigkeit gewährleisten. Eine gute Umsetzung der MAC

beinhaltet eine Prüfung der gesamten Sicherheitsanforderungen und einen wirksamen Einsatz der verschiedenen Module.

Ein System, auf dem eine MAC verwendet wird, muß zumindest garantieren, dass einem Nutzer nicht gestattet wird, Sicherheitsmerkmale nach eigenem Ermessen zu verändern; dass Arbeitswerkzeuge, Programme und Skripte, innerhalb der Beschränkungen arbeiten können, welche die Zugriffsregeln der ausgewählten Module dem System auferlegen; und dass die volle Kontrolle über die Regeln der MAC beim Administrator ist und bleibt.

Es ist die einsame Pflicht des zuständigen Administrators, die richtigen Module sorgfältig auszuwählen. Einige Umgebungen könnten eine Beschränkung der Zugriffe über die Netzwerkschnittstellen benötigen - hier wären die Module mac_portacl(4), mac_ifoff(4) und sogar mac_biba(4) ein guter Anfang. In anderen Fällen muß man sehr strenge Vertraulichkeit von Dateisystemobjekten gewährleisten - dafür könnte man mac_bsdextended(4) oder mac_mls(4) einsetzen.

Die Entscheidung, welche Richtlinien angewandt werden, kann auch anhand der Netzwerk-Konfiguration getroffen werden. Nur bestimmten Benutzern soll erlaubt werden, via ssh(1) auf das Netzwerk oder Internet zuzugreifen - mac_portacl(4) wäre eine gute Wahl. Aber für was entscheidet man sich im Falle eines Dateisystems? Soll der Zugriff auf bestimmte Verzeichnisse von spezifischen Nutzern oder Nutzergruppen separiert werden? Oder wollen wir den Zugriff durch Nutzer oder Programme auf spezielle Dateien einschränken, indem wir gewisse Objekte als geheim einstufen?

Der Zugriff auf Objekte kann einigen vertraulichen Nutzern gestattet werden, anderen wiederum verwehrt. Als Beispiel sei hierzu ein großes Entwicklerteam angeführt, das in kleine Gruppen von Mitarbeitern aufgeteilt wurde. Die Entwickler von Projekt A dürfen nicht auf Objekte zugreifen, die von den Entwicklern von Projekt B geschrieben wurden. Sie müssen aber trotzdem auf Objekte zugreifen können, die von einem dritten Entwicklerteam geschaffen wurden - alles in allem eine verzwickte Situation. Wenn man die verschiedenen Module der MAC richtig verwendet, können Anwender in solche Gruppen getrennt und ihnen der Zugriff zu den gewünschten Systemobjekten gestattet werden - ohne Angst haben zu müssen, dass Informationen in die falschen Hände geraten.

So hat jedes Modul, das eine Sicherheitsrichtlinie verfügbar macht, einen eigenen Weg, die Sicherheit des Systems zu verstärken. Die Auswahl der Module sollte auf einem gut durchdachten Sicherheitskonzept gründen. In vielen Fällen muß das gesamte Konzept eines Systems überarbeitet und neu eingepflegt werden. Ein guter Überblick über die Möglichkeiten der verschiedenen von der MAC angebotenen Module hilft einem Administrator, die besten Richtlinien für seine spezielle Situation auszuwählen.

Im FreeBSD-Standardkernel ist die Option zur Verwendung der MAC nicht enthalten. Daher muß die Zeile

```
options      MAC
```

der Kernelkonfiguration hinzugefügt und der Kernel neu übersetzt und installiert werden.

> **Achtung:** Verschiedenen Anleitungen für die MAC empfehlen, die einzelnen Module direkt in den Kernel einzuarbeiten. Dabei ist es jedoch möglich, das System aus dem Netzwerk auszusperren oder gar schlimmeres. Die Arbeit mit der MAC ist ähnlich der Arbeit mit einer Firewall - man muß, wenn man sich nicht selbst aus dem System aussperren will, genau aufpassen. Man sollte sich eine Möglichkeit zurechtlegen, wie man eine Implementation einer MAC rückgängig machen kann - genauso wie eine Ferninstallation über das Netzwerk nur mit äußerster Vorsicht vorgenommen werden sollte. Es wird daher empfohlen, die Module nicht in den Kernel einzubinden, sondern sie beim Systemstart via `/boot/loader.conf` zu laden.

17.4. MAC Labels verstehen

MAC Label sind Sicherheitsmerkmale, die, wenn sie zum Einsatz kommen, allen Subjekten und Objekten im System zugeordnet werden.

Wenn ein Administrator ein solches Merkmal bzw. Attribut setzen will, muß er/sie verstehen können, was da genau passiert. Die Attribute, die im speziellen Fall zu vergeben sind, hängen vom geladenen Modul und den darin jeweils implementierten Richtlinien ab. Jedes dieser Richtlinienmodule setzt die Arbeit mit seinen entsprechenden Attributen in individueller Weise um. Falls der Nutzer nicht versteht, was er da konfiguriert, oder auch, was seine Konfiguration für Begleiterscheinungen mit sich bringt, ergibt sich meist als Resultat ein unerwartetes, ja sogar unerwünschtes Verhalten des gesamten Systems.

Ein Label, einem Objekt verliehen, wird verwendet, um anhand einer Richtlinie eine sicherheitsrelevante Entscheidung über Zugriffsrechte zu fällen. In einigen Richtlinien enthält bereits das Label selbst alle dafür nötigen Informationen. Andere Richtlinien verwenden diese Informationen, um zunächst ein komplexes Regelwerk abzuarbeiten.

Wenn man zum Beispiel einer Datei das Attribut `biba/low` zuordnet, wird dieses durch das Biba Sicherheitsrichtlinienmodul, und zwar mit dem Wert “low”, verarbeitet.

Einige der Richtlinienmodule, die die Möglichkeit zum Vergeben von Labels unter FreeBSD unterstützen, bieten drei vordefinierte Labels an. Dieses nennen sich “high”, “low” und “equal”. Obwohl die verschiedenen Module die Zugriffskontrolle auf verschiedene Weisen regeln, kann man sich sicher sein, das das “low”-Label der untersten, unsichersten Einstellung entspricht, das “equal”-Label die Verwendung des Moduls für das jeweilige Objekt oder Subjekt deaktiviert - und das “high”-Label die höchstmögliche Einstellung erzwingt. Im Speziellen gilt diese Aussage für die Richtlinien(-module) MLS und Biba.

In den meisten Umgebungen, sogenannten Single Label Environments, wird Objekten nur ein einzelnes Label zugewiesen. Dadurch wird nur ein Regelsatz für die Zugriffskontrolle auf das gesamte System verwendet - und das ist meistens auch tatsächlich ausreichend. Es gibt wenige Fälle, in denen mehrere Labels auf Dateisystemobjekte oder -subjekte verwendet werden. In einem solchen Fall muß das Dateisystem mit der tunefs(8)-Option `multilabel` angepaßt werden, da `single label` die Standardeinstellung ist.

Bei der Verwendung von Biba oder MLS kann man numerische Labels vergeben, die genau das Level angeben, an welcher Stelle in der Hierarchie das Subjekt oder Objekt einzuordnen ist. Dieses numerische Level wird verwendet, um Informationen in verschiedene Gruppen aufzuteilen oder zu sortieren - damit zum Beispiel nur Subjekte, die zu einer gewissen Vertraulichkeitsstufe gehören, Zugang zu einer Gruppe von Objekten erhalten.

In den meisten Fällen wird ein Administrator nur ein einzelnes Label für das gesamte Dateisystem verwenden.

Moment mal, dass ist doch dasselbe wie DAC! Ich dachte, MAC würde die Kontrolle strengstens an den Administrator binden! Diese Aussage hält immer noch stand - `root` ist derjenige, der die Kontrolle ausübt und die Richtlinie konfiguriert, so dass Nutzer in die entsprechenden, angemessenen Kategorien / Zugriffsklassen eingeordnet werden. Nunja, einige Module schränken `root` selbst ein. Die Kontrolle über Objekte wird dann einer Gruppe zugewiesen, jedoch hat `root` die Möglichkeit, die Einstellungen jederzeit zu widerrufen oder zu ändern. Dies ist das Hierarchie/Freigabe-Modell, das durch Richtlinien wie MLS oder Biba bereitgestellt wird.

17.4.1. Konfigurieren der Labels

Gewissermaßen alle Aspekte der Labelkonfiguration werden durch Werkzeuge das Basissystems umgesetzt. Die entsprechenden Kommandos bieten eine einfache Schnittstelle zum Konfigurieren, Manipulieren und auch Verifizieren der gekennzeichneten Objekte.

Mit den beiden Kommandos setfmac(8) und setpmac(8) kann man eigentlich schon alles machen. Das Kommando `setfmac` wird verwendet, um ein MAC-Label auf einem Systemobjekt zu setzen, `setpmac` hingegen zum Setzen von Labels auf Systemsubjekte. Als Beispiel soll hier dienen:

```
# setfmac biba/high test
```

Wenn bei der Ausführung dieses Kommandos keine Fehler aufgetreten sind, gelangt man zur Eingabeaufforderung zurück. Nur wenn ein Fehler auftritt, verhalten sich diese Kommandos nicht still, ganz wie auch die Kommandos chmod(1) und chown(8). In einigen Fällen wird dieser Fehler `Permission denied` lauten und gewöhnlich dann auftreten, wenn ein Label an einem Objekt angebracht oder verändert werden soll, das bereits (Zugriffs-)Beschränkungen unterliegt.[1] Der Systemadministrator kann so eine Situation mit Hilfe der folgenden Kommandos überwinden:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Wie wir hier sehen, kann `setpmac` verwendet werden, um die vorhandene Einstellungen zu umgehen, indem dem gestarteten Prozeß ein anderes, valides Label zugeordnet wird. Das Werkzeug `getpmac` wird normalerweise auf gerade laufende Prozesse angewendet. Ähnlich **sendmail**: Als Argument wird statt eines Kommandos eine eine Prozeß-ID übergeben, es verbirgt sich doch dieselbe Logik dahinter. Wenn ein Nutzer versucht, eine Datei zu verändern, auf die er keinen Zugriff hat, entsprechend der Regeln eines geladenen Richtlinienmoduls, wird der Fehler `Operation not permitted` durch die Funktion `mac_set_link` angezeigt.

17.4.1.1. Übliche Typen von Labeln

Wenn man die Module mac_biba(4), mac_mls(4) und mac_lomac(4) verwendet, hat man die Möglichkeit, einfache Label zu vergeben. Diese nennen sich `high`, `low` und `equal`. Es folgt eine kurze Beschreibung, was diese Labels bedeuten:

- Das Label `low` ist definitionsgemäß das niedrigeste Label, das einem Objekt oder Subjekt verliehen werden kann. Wird es gesetzt, kann die entsprechende Entität nicht mehr auf Entitäten zugreifen, die das Label `high` tragen.
- Das Label `equal` wird Entitäten verliehen, die von der Richtlinie ausgenommen sein sollen.
- Das Label `high` verleiht einer Entität die höchstmögliche Einstellung.

Unter Beachtung jedes einzelnen Richtlinienmoduls moduliert und beschränkt jede dieser Einstellungen den Informationsfluß unterschiedlich. Genaue Erklärungen zu den Charakteristika der einfachen Labels in den verschiedenen Modulen finden sich im entsprechenden Unterabschnitt dieses Kapitels oder in den Manpages.

17.4.1.1.1. Fortgeschrittene Label-Konfiguration

Numerische klassifizierte Labels werden verwendet in der Form `Klasse:Verbund+Verbund`. Demnach ist das Label

```
biba/10:2+3+6(5:2+3-15:2+3+4+5+6)
```

folgendermaßen zu lesen:

“Biba Policy Label”/“effektive Klasse 10” :“Verbund 2,3 und 6”: (“Low-Klasse 5:...”- “High-Klasse 15:...”)

In diesem Beispiel ist die erstgenannte Klasse als “effektive Klasse” zu bezeichnen. Ihr werden die “effektiven Verbünde” zugeordnet. Die zweite Klasse ist die “Low”-Klasse und die letzte die “high”-Klasse. Die allermeisten Konfigurationen kommen ohne die Verwendungen von solchen Klassen aus, nichtsdestotrotz kann man sie für erweiterte Konfigurationen verwenden.

Sobald sie auf *Systemsubjekte* angewendet werden, haben diese eine gegenwärtige Klasse/Verbund- Konfiguration und diese muß im definierten Rahmen gegebenenfalls angepaßt (erhöht oder gesenkt) werden. Im Gegensatz dazu haben *Systemobjekte* alle eingestellten (effektive, High- und Low-Klasse) gleichzeitig. Dies ist notwendig, damit auf Sie von den *Systemsubjekten* in den verschiedenen Klassen gleichzeitig zugegriffen werden kann.

Die Klasse und und die Verbünde in einem Subjekt-Objekt-Paar werden zum Erstellen einer sogenannten Dominanz-Relation verwendet, in welcher entweder das Subjekt das Objekt, das Objekt das Subjekt, keines das andere dominiert oder sich beide gegenseitig dominieren. Der Fall, dass sich beide dominieren, tritt dann ein, wenn die beiden Labels gleich sind. Wegen der Natur des Informationsflusses in Biba kann man einem Nutzer Rechte für einen Reihe von Abteilungen zuordnen, die zum Beispiel mit entsprechenden Projekten korrespondieren. Genauso können aber auch Objekten mehrere Abteilungen zugeordnet sein. Die Nutzer müssen eventuell ihre gegenwärtigen Rechte mithilfe von `su` or `setpmac` anpassen um auf Objekte in einer Abteilung zuzugreifen, zu der sie laut ihrer effektiven Klasse nicht berechtigt sind.

17.4.1.2. Nutzer- und Label-Einstellungen

Nutzer selbst brauchen Labels damit ihre Dateien und Prozesse korrekt mit der Sicherheitsrichtlinie zusammenarbeitet, die für das System definiert wurde. Diese werden in der Datei `login.conf` durch die Verwendung von Login- Klassen zugeordnet. Jedes Richtlinienmodul, das Label verwendet, arbeitet mit diesen Login-Klassen.

Beispielhaft wird der folgende Eintrag, der für jede Richtlinie eine Einstellung enthält, gezeigt:

```
default:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5,biba/10(5-15),lomac/10[2]:
```

Die Label-Option in der letzten Zeile legt fest, welches Standard-Label für einen Nutzer erzwungen wird. Nutzern darf niemals gestattet werden, diese Werte selbst zu verändern, demnach haben Nutzer in dieser Beziehung auch keine Wahlfreiheit. In einer richtigen Konfiguration jedoch wird kein Administrator alle Richtlinienmodule aktivieren wollen. Es wird an dieser Stelle ausdrücklich empfohlen, dieses Kapitel zu Ende zu lesen, bevor irgendein Teil dieser Konfiguration ausprobiert wird.

> **Anmerkung:** Nutzer können ihr eigenes Label nach dem Loginvorgang durchaus ändern. Jedoch kann diese Änderung nur unter den Auflagen der gerade gültigen Richtlinie geschehen. Im Beispiel oben wird für die Biba-Richtlinie eine minimale Prozeßintegrität von 5, eine maximale von 15 angegeben, aber die Voreinstellung des tatsächlichen Labels ist 10. Der Nutzerprozeß läuft also mit einer Integrität von 10 bis das Label verändert wird, zum Beispiel durch eine Anwendung des Kommandos `setpmac`, welches jedoch auf den Bereich eingeschränkt wird, der zum Zeitpunkt des Logins angegeben wurde, in diesem Fall von 5 bis 15.

Nach einer Änderung der Datei `login.conf` muß in jedem Fall die Befähigungsdatenbank mit dem Kommando `cap_mkdb` neu erstellt werden - und das gilt für alle im weiteren Verlauf gezeigten Beispiele und Diskussionspunkte.

Es ist nützlich anzumerken, dass viele Einsatzorte eine große Anzahl von Nutzern haben, die wiederum viele verschiedenen Nutzerklassen angehören sollen. Hier ist eine Menge Planungsarbeit notwendig, da die Verwaltung sehr unübersichtlich und schwierig ist.

17.4.1.3. Netzwerkschnittstellen und die zugehörigen Label

Labels können auch, wenn man sie an Netzwerkschittstellen vergibt, helfen, den Datenfluß durch das Netzwerk zu kontrollieren. Das funktioniert in allen Fällen genau so wie mit Objekten. Nutzer, die in der Biba-Richtlinie das Label `high` tragen, dürfen nicht auf Schnittstellen zugreifen, die `low` markiert sind usw.

Die Option `maclabel` wird via `ifconfig` übergeben. Zum Beispiel

```
# ifconfig bge0 maclabel biba/equal
```

belegt die Schnittstelle bge(4) mit dem MAC Label `biba/equal`. Wenn eine komplexe Einstellung wie `biba/high(low-high)` verwendet wird, muß das gesamte Label in Anführungszeichen geschrieben werden, da sonst eine Fehlermeldung zurückgegeben wird.

Jedes Richtlinienmodul, das die Vergabe von Labels unterstützt, stellt einen Parameter bereit, mit dem das MAC Label für Netzwerkschnittstellen deaktiviert werden kann. Das Label der Netzwerkschnittstelle auf `equal` zu setzen, führt zum selben Ergebnis. Beachten Sie die Ausgabe von `sysctl`, die Manpages der verschiedenen Richtlinien oder eben die Informationen, die im weiteren Verlauf dieses Kapitels angeboten werden, um mehr zu diesen Parametern zu erfahren.

17.4.2. Single- oder Multilabel?

Als Standardeinstellung verwendet das System die Option `single label`. Was bedeutet das für den Administrator? Es gibt einige Unterschiede zwischen `single label` und `multilabel`. In ihrer ureigenen Weise bieten beide Vor- und Nachteile bezogen auf die Flexibilität bei der Modellierung der Systemsicherheit.

Die Option `single label` gibt jedem Subjekt oder Objekt genau ein einziges Label, zum Beispiel `biba/high`. Mit dieser Option hat man einen geringeren Verwaltungsaufwand, aber die Flexibilität beim Einsatzes von

Richtlinien ist ebenso gering. Viele Administratoren wählen daher auch die Option `multilabel` im Sicherheitsmodell, wenn die Umstände es erfordern.

Die Option `multilabel` gestattet, jedem einzelnen Subjekt oder Objekt seine eigenen unabhängigen Label zu zuzuordnen. Die Optionen `multilabel` und `singlelabel` betreffen jedoch nur die Richtlinien, die Labels als Leistungsmerkmal verwenden, einschließlich der Richtlinien Biba, Lomac, MLS und SEBSD.

Wenn Richtlinien benutzt werden sollen, die ohne Labels auskommen, wird die Option `multilabel` nicht benötigt. Dies betrifft die Richtlinien `seeotheruids`, `portacl` und `partition`.

Man sollte sich dessen bewußt sein, dass die Verwendung der Option `multilabel` auf einer Partition und die Erstellung eines Sicherheitsmodells auf der Basis der FreeBSD `multilevel` Funktionalität einen hohen Verwaltungsaufwand bedeutet, da alles im Dateisystem ein Label bekommt. Jedes Verzeichnis, jede Datei und genauso jede Schnittstelle.

Das folgende Kommando aktiviert `multilabel` für ein Dateisystem. Dies funktioniert nur im Einzelbenutzermodus:

```
# tunefs -l enable /
```

In einer Swap-Partition wird dies nicht benötigt.

> **Anmerkung:** Falls Sie Probleme beim Setzen der Option `multilabel` auf der Root-Partition bemerken, lesen Sie bitte Abschnitt 17.17 dieses Kapitels.

17.5. Planung eines Sicherheitsmodells

Wann immer eine neue Technologie eingepflegt werden soll, ist es wichtig, vorher einen Plan zu erstellen. In den verschiedenen Etappen der Planung sollte der Administrator nie das “Große Ganze” aus den Augen verlieren und mindestens die folgenden Punkte beachten:

- Die Anforderungen
- Die Ziele

Wenn Sie MAC verwenden möchten, sind das im Besonderen folgende Punkte:

- Wie werden Informationen und Ressourcen auf den Zielsystemen klassifiziert?
- Welche Arten von Informationen bzw. Ressourcen sollen im Zugang beschränkt sein und welche Art Einschränkung soll verwendet werden?
- Welche(s) MAC Modul(e) wählt man, um sein Ziel zu erreichen?

Es ist immer möglich, die Einstellungen des Systems und der Systemressourcen im Nachhinein zu “optimieren”. Es ist aber wirklich lästig, das gesamte Dateisystem zu durchsuchen, um Dateien oder Benutzerkonten zu reparieren. Eine gute Planung hilft dem Administrator, sich einer sorgenfreien und effizienten Umsetzung eines Sicherheitsmodells zu versichern. Testlauf des Sicherheitsmodells *vor* dem Einsatz in seiner richtigen Arbeitsumgebung ist auf jeden Fall empfehlenswert. Die Idee, ein System mit einer MAC einfach loslaufen zu lassen, ist wie direkt auf einen Fehlschlag hinzuarbeiten.

Jede Umgebung hat ihre eigenen Anforderungen. Ein tiefgreifendes und vollständiges Sicherheitsprofil zu erstellen spart weitere Änderungen, nachdem das System in Betrieb genommen wurde. Also werden die folgenden Abschnitte die verschiedenen Module vorstellen, die den Administratoren zur Verfügung gestellt werden, die Nutzung und Konfiguration der einzelnen Module beschreiben; und in einigen Fällen Einblicke gewähren, für welche Situationen welche Module besonders geeignet sind. Zum Beispiel ein Webserver kann von der Verwendung der mac_biba(4) oder der mac_bsdextended(4) Richtlinie profitieren. In anderen Fällen, an einem Rechner mit nur wenigen lokalen Benutzern, ist die mac_partition(4) die Richtlinie der Wahl.

17.6. Modulkonfiguration

Jedes Modul, das in der MAC enthalten ist, kann entweder direkt in den Kernel eingefügt werden oder als Kernelmodul in der Laufzeit des Systems geladen werden. Empfohlen wird, den Modulnamen in der Datei `/boot/loader.conf` anzufügen, so dass das Modul am Anfang des Bootvorgangs eingebunden wird.

Die folgenden Abschnitte werden verschiedene MAC Module und ihre jeweiligen Vor- und Nachteile vorstellen. Außerdem wird erklärt, wie sie in bestimmte Umgebungen eingearbeitet werden können. Einige Module unterstützen die Verwendung von `Labels`, das heißt Zugriffskontrolle durch hinzufügen einer Kennzeichnung in der Art von “dieses ist erlaubt, jenes aber nicht”. Eine Label-Konfigurationdatei kontrolliert unter anderem, wie auf Dateien zugegriffen oder wie über das Netzwerk kommuniziert werden darf. Im vorangehenden Abschnitt wurde bereits erläutert, wie die Option `multilabel` auf Dateisysteme angewendet wird, um eine Zugriffskontrolle auf einzelne Dateien oder ganze Dateisysteme zu konfigurieren.

Eine `single label` Konfiguration erzwingt ein einzelnes Label für das gesamte System. Daher wird die `tunefs`-Option `multilabel` genannt.

17.7. Das MAC Modul seeotheruids

Modulename: `mac_seeotheruids.ko`

Parameter in der Kernelkonfiguration: `options MAC_SEEOTHERUIDS`

Bootparameter: `mac_seeotheruids_load="YES"`

Das Modul mac_seeotheruids(4) erweitert die `sysctl`-Variablen `security.bsd.see_other_uids` und `security.bsd.see_other_gids`. Diese Optionen benötigen keine im Vorhinein zu setzenden Labels und können leicht durchschaubar mit den anderen MAC-Modulen zusammenarbeiten.

Nachdem das Modul geladen wurde, können die folgenden `sysctl` Variablen verwendet werden.

- `security.mac.seeotheruids.enabled` dient zur Aktivierung des Moduls, zunächst mit den Standardeinstellungen. Diese verhindern, dass Nutzer Prozesse und Sockets sehen können, die ihnen nicht selbst gehöen.
- `security.mac.seeotheruids.specificgid_enabled` kann eine spezifizierte Nutzergruppe von dieser Richtlinie ausnehmen. Die entsprechende Gruppe muß an den Parameter `security.mac.seeotheruids.specificgid=`*`XXX`* übergeben werden, wobei *`XXX`* die ID der Gruppe ist, die von der Richtlinie ausgenommen werden soll.
- `security.mac.seeotheruids.primarygroup_enabled` kann verwendet werden, um eine spezifische, *primäre* Nutzergruppe von der Richtlinie auszuschliessen. Dieser Parameter und `security.mac.seeotheruids.specificgid_enabled` schließen einander aus.

17.8. Das MAC Modul bsdextended

Modulname: `mac_bsdextended.ko`

Parameter in der Kernelkonfiguration: `options MAC_BSDEXTENDED`

Bootparameter: `mac_bsdextended_load="YES"`

Das Modul mac_bsdextended(4) erstellt eine Firewall für das Dateisystem und ist eine Erweiterung des sonst üblichen Rechtemodells. Es erlaubt einem Administrator einen Regelsatz zum Schutz von Dateien, Werkzeugen und Verzeichnissen in der Dateisystemhierarchie zu erstellen, der einer Firewall ähnelt. Sobald auf ein Objekt im Dateisystem zugegriffen werden soll, wird eine Liste von Regel abgearbeitet, bis eine passende Regel gefunden wird oder die Liste zu Ende ist. Das Verhalten kann durch die Änderung des sysctl(8) Parameters `security.mac.bsdextended.firstmatch_enabled` eingestellt werden. Ähnlich wie bei den anderen Firewallmodulen in FreeBSD wird eine Datei erstellt, welche die Zugriffsregeln enthält. Diese wird beim Systemstart durch eine Variable in rc.conf(5) eingebunden.

Der Regelsatz kann mit dem Programm ugidfw(8) eingepflegt werden, welches eine Syntax bereitstellt, die der von ipfw(8) gleicht. Weitere Werkzeuge können auch selbst erstellt werden, indem die Funktionen der Bibliothek libugidfw(3) verwendet werden.

Bei der Arbeit mit diesem Modul ist äußerste Vorsicht geboten - falscher Gebrauch kann den Zugriff auf Teile des Dateisystems komplett unterbinden.

17.8.1. Beispiele

Nachdem das Modul mac_bsdextended(4) erfolgreich geladen wurde, zeigt das folgende Kommando die gegenwärtig aktiven Regeln an:

```
# ugidfw list 0 slots, 0 rules
```

Wie erwartet, sind keine Regeln definiert. Das bedeutet, das auf alle Teile des Dateisystems zugegriffen werden kann. Um eine Regel zu definieren, die jeden Zugriff durch Nutzer blockiert und nur die Rechte von `root` unangetastet läßt, muß lediglich dieses Kommando ausgeführt werden:

```
# ugidfw add subject not uid root new object not uid root mode n
```

Das ist allerdings keine gute Idee, da nun allen Nutzern der Zugriff auf selbst die einfachsten Programme wie `ls` untersagt wird. Angemessener wäre etwas wie:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Diese Befehle bewirken, dass `user1` keinen Zugriff mehr auf Dateien und Programme hat, die *`user2`* gehören. Dies schließt das Auslesen von Verzeichniseinträgen ein.

Anstelle `uid user1` könnte auch `not uid` *`user2`* als Parameter übergeben werden. Dies würde diesselben Einschränkungen für alle Nutzer bewirken anstatt nur einen einzigen.

> **Anmerkung:** `root` ist von diesen Einstellungen nicht betroffen.

Dies sollte als Überblick ausreichen, um zu verstehen, wie das Modul mac_bsdextended(4) helfen kann, das Dateisystem abzuschotten. Weitere Informationen bieten die Manpages mac_bsdextended(4) und ugidfw(8).

17.9. Das MAC Modul ifoff

Modulname: `mac_ifoff.ko`

Parameter für die Kernelkonfiguration: `options MAC_IFOFF`

Bootparameter: `mac_ifoff_load="YES"`

Das Modul mac_ifoff(4) ist einzig dazu da, Netzwerkschnittstellen im laufenden Betrieb zu deaktivieren oder zu verhindern, das Netzwerkschnittstellen während der Bootphase gestartet werden. Dieses Modul benötigt für seinen Betrieb weder Labels, die auf dem System eingerichtet werden müssen, noch hat es Abhängigkeiten zu anderen MAC Modulen.

Der größte Teil der Kontrolle geschieht über die im folgenden aufgelisteten `sysctl`-Parameter:

- `security.mac.ifoff.lo_enabled` schaltet den gesamten Netzwerkverkehr auf der Loopback-Schnittstelle lo(4) an bzw. aus.
- `security.mac.ifoff.bpfrecv_enabled` macht das Gleiche für den Berkeley Paket Filter bpf(4).
- `security.mac.ifoff.other_enabled` schaltet den Verkehr für alle anderen Netzwerkschnittstellen.

Die wahrscheinlich häufigste Nutzung von mac_ifoff(4) ist die Überwachung des Netzwerks in einer Umgebung, in der kein Netzwerkverkehr während des Bootvorgangs erlaubt werden soll. Eine andere mögliche Anwendung wäre ein Script, das mit Hilfe von `security/aide` automatisch alle Schnittstellen blockiert, sobald Dateien in geschützten Verzeichnissen angelegt oder verändert werden.

17.10. Das MAC Modul portacl

Modulname: `mac_portacl.ko`

Parameter für die Kernelkonfiguration: `options MAC_PORTACL`

Bootparameter: `mac_portacl_load="YES"`

Mit Hilfe des Moduls mac_portacl(4) können die Anbindungen an die lokalen TCP und UDP Ports durch eine Vielzahl von `sysctl` Variablen beschränkt werden. Genauer gesagt ermöglicht mac_portacl(4) Nutzern ohne `root`-Rechten den Zugriff auf zu bestimmende privilegierte Ports, also denen innerhalb der ersten 1024.

Sobald das Modul geladen wurde, ist die Richtlinie für alle Sockets verfügbar. Die folgenden Variablen können für die Konfiguration verwendet werden:

- `security.mac.portacl.enabled` schaltet die Anwendung der Richtlinie ein oder aus.
- `security.mac.portacl.port_high` gibt den höchsten Port an, der von der Richtlinie mac_portacl(4) betroffen sein soll.
- `security.mac.portacl.suser_exempt` nimmt, wenn es einen Wert ungleich Null zugewiesen bekommt, `root` von der Richtlinie aus.

- `security.mac.portacl.rules` enthält als Wert die eigentliche `mac_portacl` Richtlinie.

Die eigentliche Konfiguration der `mac_portacl` Richtlinie wird der `sysctl`-Variablen `security.mac.portacl.rules` als Zeichenkette der Form `rule[,rule,...]` übergeben. Jede einzelne Regel hat die Form `idtype:id:protocol:port`. Der Parameter *`idtype`* ist entweder `uid` oder `gid` und wird verwendet, um den Parameter *`id`* als Nutzer-ID oder Gruppen-ID zu kennzeichnen. Der Parameter *`protocol`* gibt an, ob die Regel ür TCP oder UDP gelten soll (indem man den Wert auf `tcp` oder `udp` setzt). Und der letzte Parameter, *`port`*, enthält die Nummer des Ports, auf den der angegebene Nutzer bzw. die angegebene Gruppe Zugriff erhalten soll.

> **Anmerkung:** Da der Regelsatz direkt vom Kernel ausgewertet wird, können nur Zahlenwerte übergeben werden. Das heißt, Namen von Nutzern, Gruppen oder Dienstnamen aus der Datei `/etc/services` funktionieren nicht.

Auf UNIX-artigen Betriebssystemen sind die Ports kleiner 1024 privilegierten Prozessen vorbehalten, müssen also mit als/von `root` gestartet werden und weiterhin laufen. Damit mac_portacl(4) die Vergabe von Ports kleiner als 1024 an nicht privilegierte Prozesse übernehmen kann, muß die UNIX Standardeinstellung deaktiviert werden. Dazu ändert man die sysctl(8) Variablen `net.inet.ip.portrange.reservedlow` und `net.inet.ip.portrange.reservedhigh` auf den Wert “0”.

Weiterführende Informationen entnehmen Sie bitte den unten aufgeführten Beispielen oder der Man-Page mac_portacl(4)!

17.10.1. Beispiele

Die folgenden Beispiele sollten ein wenig Licht in die obige Diskussion bringen:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

Zunächst bestimmen wir, dass mac_portacl(4) für alle privilegierten Ports gelten soll und deaktivieren die normale UNIX-Beschränkung.

```
# sysctl security.mac.portacl.suser_exempt=1
```

Da `root` von dieser Richtlinie nicht beeinträchtigt werden soll, setzen wir hier `security.mac.portacl.suser_exempt` auf einen Wert ungleich Null. Das Modul mac_portacl(4) ist nun so eingerichtet, wie es UNIX-artige Betriebssysteme normal ebenfalls tun.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Nun erlauben wir dem Nutzer mit der UID 80, normalerweise dem Nutzer `www`, den Port 80 zu verwenden. Dadurch kann der Nutzer `www` einen Webserver betreiben, ohne dafür mit `root`-Privilegien ausgestattet zu sein.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Hier wird dem Nutzer mit der UID 1001 erlaubt, die TCP Ports 110 (“pop3”) und 995 (“pop3s”) zu verwenden. Dadurch kann dieser Nutzer einen Server starten, der Verbindungen an diesen beiden Ports annehmen kann.

17.11. Das MAC Modul partition

Modulname: `mac_partition.ko`

Parameter für die Kernelkonfiguration: `options MAC_PARTITION`

Bootparameter `mac_partition_load="YES"`

Die Richtlinie mac_partition(4) setzt Prozesse in spezielle “Partitionen”, entsprechend dem zugewiesenen MAC Label. Man kann sich das vorstellen wie eine spezielle Art jail(8), auch wenn das noch kein wirklich guter Vergleich ist.

Es wird empfohlen, dieses Modul durch einen Eintrag in loader.conf(5) zu aktivieren, so dass die Richtlinie während des Bootvorganges eingebunden wird.

Der Großteil der Konfiguration geschieht mit dem Kommando setpmac(8), wie gleich erklärt wird. Außerdem gibt es folgenden `sysctl` Parameter für diese Richtlinie.

- `security.mac.partition.enabled` erzwingt die Verwendung von MAC Prozeß-Partitionen.

Sobald diese Richtlinie aktiv ist, sehen Nutzer nur noch ihre eigenen Prozesse, und alle anderen Prozesse, die ebenfalls derselben Prozeß-Partition zugeordnet sind. Sie können jedoch nicht auf Prozesse oder Werkzeuge außerhalb des Anwendungsbereich dieser Partition zugreifen. Das bedeutet unter anderem, das ein Nutzer, der einer Klasse `insecure` zugeordnet ist, nicht auf das Kommando `top` zugreifen kann - wie auch auf viele anderen Befehle, die einen eigenen Prozeß erzeugen.

Um einen Befehl einer Prozeß-Partition zuzuordnen, muß dieser durch das Kommando `setpmac` mit einem Label versehen werden:

```
# setpmac partition/13 top
```

Diese Zeile fügt das Kommando `top` dem Labelsatz für Nutzer der Klasse `insecure` hinzu, sofern die Partition 13 mit der Klasse `insecure` übereinstimmt. Beachten Sie, dass alle Prozesse, die von Nutzern dieser Klasse erzeugt werden, das Label `partition/13` erhalten, und dieses auch nicht durch den Nutzer geändert werden kann.

17.11.1. Beispiele

Der folgende Befehl listet die vergebenen Label für Prozeß-Partitionen und die laufenden Prozesse auf.

```
# ps Zax
```

Das nächste Kommando liefert das Label der Prozeß-Partition eines anderen Nutzers `trhodes` und dessen gegenwärtig laufenden Prozesse zurück.

```
# ps -ZU trhodes
```

> **Anmerkung:** Jeder Nutzer kann die Prozesse in der Prozeß-Partition von `root` betrachten, solange nicht die Richtlinie mac_seeotheruids(4) geladen wurde.

Eine ausgefeilte Umsetzung dieser Richtlinie deaktiviert alle Dienste in `/etc/rc.conf` und startet diese dann später durch ein Skript, das jedem Dienst das passende Label zuordnet.

Anmerkung: Die folgenden Richtlinien verwenden Zahlenwerte anstatt der drei Standardlabels. Diese Optionen, und ihre Grenzen, werden in den zugehörigen Manpages genauer erklärt.

17.12. Das MAC Modul Multi-Level Security

Modulname: `mac_mls.ko`

Parameter für die Kernelkonfiguration: `options MAC_MLS`

Bootparameter: `mac_mls_load="YES"`

Die Richtlinie mac_mls(4) kontrolliert die Zugriffe zwischen Subjekten und Objekten, indem sie den Informationsfluß strengen Regeln unterwirft.

In MLS Umgebungen wird jedem Subjekt oder Objekt ein “Freigabe”-Level zugeordnet, und diese werden wiederum zu einzelnen Verbünden zusammengefaßt. Da diese Freigabe- oder Anfälligkeits-Level Zahlen größer 6000 erreichen können, ist es für jeden Systemadministrator eine undankbare Aufgabe, jede Entität von Grund auf zu konfigurieren. Zum Glück gibt es 3 “instant” Labels, die in der Richtlinie zur Anwendung bereit stehen.

Diese drei Labels heißen `mls/low`, `mls/equal` und `mls/high`. Da sie in den Manpages mac_mls(4) ausführlich beschrieben werden, gibt es hier nur einen kurzen Abriß:

- Das Label `mls/low` ist eine niedrige Einstellung, die von allen anderen dominiert werden darf. Alles, was mit `mls/low` versehen wird, hat ein niedriges Freigabe-Level und darf auf keine Informationen zugreifen, denen ein höheres Freigabe-Level zugeordnet wurde. Einem Objekt mit diesem Label kann außerdem keine Information durch ein Objekt höherer Freigabe übergeben werden, es kann also auch nicht durch solche Objekte editiert oder überschrieben werden.
- Das Label `mls/equal` wird an Objekte vergeben, die von dieser Richtlinie ausgenommen werden sollen.
- Das Label `mls/high` verkörpert das höchstmögliche Freigabe-Level. Objekte, denen dieses Label zugeordnet wird, dominieren alle anderen Objekte des Systems. Trotzdem können sie Objekten mit einem niedrigeren Freigabe-Level keine Informationen zuspielen.

MLS bietet:

- Eine hierarchische Sicherheitsschicht und Zuordnung nichthierarchischer Kategorien;
- Feste Regeln: kein “Read-Up”, kein “Write-Down” (ein Subjekt kann nur Objekte gleicher oder *niedrigerer* Stufe lesen, und es kann nur Objekte gleicher oder *höherer* Stufe schreiben);
- Geheimhaltung (indem unangemessene Offenlegung von Daten verhindert wird);
- Eine Basis zum Entwerfen von Systemen, die Daten verschiedener Vertraulichkeitsebenen gleichzeitig handhaben sollen (ohne das geheime und vertrauliche Informationen untereinander ausgetauscht werden können).

Nachfolgend werden die `sysctl`-Variablen vorgestellt, die für die Einrichtung spezieller Dienste und Schnittstellen vorhanden sind.

- `security.mac.mls.enabled` schaltet die Richtlinie MLS ein (oder aus).

- `security.mac.mls.ptys_equal` sorgt dafür, dass während der Initialisierung alle pty(4)-Geräte als `mls/equal` gekennzeichnet werden.
- `security.mac.mls.revocation_enabled` sorgt dafür, dass die Zugriffsrechte von Objekten wieder zurückgesetzt werden, nachdem deren Label vorübergehend auf ein niedrigeres Freigabe-Level geändert wurde.
- `security.mac.mls.max_compartments` gibt die maximale Anzahl von Verbünden an. Im Prinzip ist es die höchste Nummer eines Verbundes auf dem System.

Um die Labels der MLS Richtlinie zu bearbeiten verwendet man setfmac(8). Um ein Objekt zu kennzeichnen, benutzen Sie folgendes Kommando:

```
# setfmac mls/5 test
```

Um das MLS-Label der Datei `test` auszulesen, verwenden Sie dieses Kommando:

```
# getfmac test
```

Dies ist eine Zusammenstellung der Merkmale von `test`. Ein anderer Ansatz ist, für diese Richtlinie eine Konfigurationsdatei in `/etc` abzulegen, die alle Informationen enthält und mit der dann das Kommando `setfmac` gefüttert wird. Diese Vorgehensweise wird erklärt, nachdem alle Richtlinien vorgestellt wurden.

17.12.1. Verbindlicher Vertraulichkeit in der Planungsphase

Mit dem Richtlinienmodul `Multi-Level Security` bereitet sich ein Administrator darauf vor, den Fluß vertraulicher Informationen zu kontrollieren. Beim Starten der Richtlinie ist immer `mls/low` voreingestellt - alles kann auf alles zugreifen. Der Administrator ändert dies während der eigentlichen Konfiguration, indem er die Vertraulichkeit bestimmter Objekte erhöht.

Jenseits der drei Grundeinstellungen des Labels kann der Administrator einzelne Nutzer oder Nutzergruppen nach Bedarf zusammenschließen und den Informationsaustausch zwischen diesen gestatten oder unterbinden. Es ist sicher eine Vereinfachung, die Freigabe-Level mit Begriffen wie `vertraulich`, `geheim` oder `streng geheim` zu bezeichnen. Einige Administratoren erstellen einfach verschiedene Gruppen auf der Ebene von gegenwärtigen Projekten. Ungeachtet der Herangehensweise bei der Klassifizierung muß ein gut durchdachter Plan existieren, bevor eine derart einengende Richtlinie umgesetzt wird.

Exemplarisch für die Anwendung dieses Moduls bzw. dieser Richtlinie seien angeführt:

- Ein E-Commerce Webserver
- Ein Dateiserver, der vertrauliche Informationen einer Firma oder eines Konzerns speichert
- Umgebungen in Finanzeinrichtungen

Der unsinnigste Einsatzort für diese Richtlinie wäre ein Arbeitsplatzrechner mit nur zwei oder drei Benutzern.

17.13. Das MAC Modul Biba

Modulname: `mac_biba.ko`

Parameter für die Kernelkonfiguration: `options MAC_BIBA`

Bootparameter: `mac_biba_load="YES"`

Das Modul mac_biba(4) lädt die MAC Biba Richtlinie. Diese ähnelt stark der MLS Richtlinie, nur das die Regeln für den Informationsfluß ein wenig vertauscht sind. Es wird in diesem Fall der absteigende Fluß sicherheitskritischer Information geregelt, während die MLS Richtlinie den aufsteigenden Fluß regelt. In gewissen Sinne treffen dieses und das vorangegangene Unterkapitel also auf beide Richtlinien zu.

In einer Biba-Umgebung wird jedem Subjekt und jedem Objekt ein “Integritäts”-Label zugeordnet. Diese Labels sind in hierarchischen Klassen und nicht-hierarchischen Komponenten geordnet. Je höher die Klasse, um so höher die Integrität.

Die unterstützten Labels heißen `biba/low`, `biba/equal` und `biba/high`. Sie werden im Folgenden erklärt:

- `biba/low` ist die niedrigste Stufe der Integrität, die einem Objekt verliehen werden kann. Wenn sie einem Objekt oder Subjekt zugeordnet wird, kann dieses auf Objekte oder Subjekte, die biba/high markiert wurden, zwar lesend zugreifen, nicht jedoch schreibend.
- Das Label `biba/equal` ist, wie der aufmerksame Leser sicherlich schon ahnt, für die Ausnahmen dieser Richtlinie gedacht und sollte nur diesen Ausnahmen entsprechenden Objekten verliehen werden.
- `biba/high` markierte Subjekte und Objekte können Objekte niedrigerer Stufe schreiben , nicht jedoch lesen. Es wird empfohlen, dass dieses Label an Objekte vergeben wird, die sich auf Integrität des gesamten Systems auswirken.

Biba stellt bereit:

- Hierarchische Integritätsstufen mit einem Satz nichthierarchischer Integritätskategorien;
- Festgeschriebene Regeln: kein “Write-Up”, kein “Read-Down” (der Gegensatz zu MLS - ein Subjekt erhält schreibenden Zugriff auf Objekte gleicher oder geringerer Stufe, aber nicht bei höherer, und lesenden Zugriff bei gleicher Stufe oder höerer, aber nicht bei niedrigerer);
- Integrität (es wird die Echtheit der Daten gewährleistet, indem unangemessene Veränderungen verhindert werden);
- Eine Abstufung der Gewährleistung (im Gegensatz zu MLS, bei der eine Abstufung der Vertraulichkeit vorgenommen wird).

Folgende `sysctl` Parameter werden zur Nutzung der Biba-Richtlinie angeboten:

- `security.mac.biba.enabled` zum Aktivieren/Deaktivieren der Richtlinie auf dem Zielsystem.
- `security.mac.biba.ptys_equal` wird verwendet, um die Biba-Richtlinie auf der pty(4)-Schnittstelle zu deaktivieren.
- `security.mac.biba.revocation_enabled` erzwingt das Zurücksetzen des Labels, falls dieses zeitweise geändert wurde um ein Subjekt zu dominieren.

Um Einstellungen der Biba Richtlinie für Systemobjekte zu verändern werden die Befehle `setfmac` und `getfmac` verwendet:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

17.13.1. Verbindliche Integrität in der Planungsphase

Integrität garantiert, im Unterschied zu Sensitivität, dass Informationen nur durch vertraute Parteien verändert werden können. Dies schließt Informationen ein, die zwischen Subjekten ausgetauscht werden, zwischen Objekt, oder auch zwischen den beiden. Durch Integrität wird gesichert, das Nutzer nur Informationen verändern, oder gar nur lesen können, die sie explizit benötigen.

Das Modul mac_biba(4) eröffnet einem Administrator die Möglichkeit zu bestimmen, welche Dateien oder Programme ein Nutzer oder eine Nutzergruppe sehen bzw. aufrufen darf. Gleichzeitig kann er zusichern, dass dieselben Programme und Dateien frei von Bedrohungen sind und das System die Echtheit gewährleistet - für diesen Nutzer oder die Nutzergruppe.

Während der anfänglichen Phase der Planung muß der Administrator vorbereitet sein, Nutzer in Klassen, Stufen und Bereiche einzuteilen. Der Zugriff auf Dateien und insbesondere auch Programme wird verhindert sowohl vor als auch nachdem sie gestartet wurden. Das System selbst erhält als Voreinstellung das Label `biba/high` sobald das Modul aktiviert wird - und es liegt allein am Administrator, die verschiedenen Klassen und Stufen für die einzelnen Nutzer zu konfigurieren. Anstatt mit Freigaben zu arbeiten, wie weiter oben gezeigt wurde, könnte man auch Überbegriffe für Projekte oder Systemkomponenten entwerfen. Zum Beispiel, ausschließlich Entwicklern den Vollzugriff auf Quellcode, Compiler und Entwicklungswerkzeuge gewähren, während man andere Nutzer in Kategorien wie Tester, Designer oder einfach nur “allgemeiner Nutzer” zusammenfaßt, die für diese Bereiche lediglich lesenden Zugriff erhalten sollen.

Mit seinem ursprünglichen Sicherheits-Standpunkt ist ein Subjekt niedrigerer Integrität unfähig, ein Subjekt höherer Integrität zu verändern. Ein Subjekt höherer Integrität kann ein Subjekt niedrigerer Integrität weder beobachten noch lesen. Wenn man ein Label für die niedrigstmögliche Klasse erstellt, kann man diese allen Subjekten verwehren. Einige weitsichtig eingerichtete Umgebungen, die diese Richtlinie verwenden, sind eingeschränkte Webserver, Entwicklungs- oder Test-Rechner oder Quellcode-Sammlungen. Wenig sinnvoll ist diese Richtlinie auf einer Arbeitsstation, oder auf Rechnern die als Router oder Firewall verwendet werden.

17.14. Das MAC Modul LOMAC

Modulname: `mac_lomac.ko`

Parameter für die Kernelkonfiguration: `options MAC_LOMAC`

Bootparameter: `mac_lomac_load="YES"`

Anders als die Biba Richtlinie erlaubt die mac_lomac(4) Richtlinie den Zugriff auf Objekte niedrigerer Integrität nur, nachdem das Integritätslevel gesenkt wurde. Dadurch wird eine Störung derIntegritätsregeln verhindert.

Die MAC Version der “Low-Watermark” Richtlinie, die nicht mit der älteren lomac(4)-Implementierung verwechselt werden darf, arbeitet fast genauso wie Biba. Anders ist, dass hier “schwebende” Label verwendet werden, die ein Herunterstufen von Subjekten durch Hilfsverbünde ermöglichen. Dieser zweite Verbund wird in der Form `[auxgrade]` angegeben und sollte in etwa aussehen wie `lomac/10[2]`, wobei die Ziffer zwei (2) hier den Hilfsverbund abbildet.

Die MAC Richtlinie `LOMAC` beruht auf einer durchgängigen Etikettierung aller Systemobjekte mit Integritätslabeln, die Subjekten das Lesen von Objekten niedriger Integrität gestatten und dann das Label des Subjektes herunterstufen - um zukünftige Schreibvorgänge auf Objekte hoher Integrität zu unterbinden. Dies ist die Funktion der Option `[auxgrade]`, die eben vorgestellt wurde. Durch sie erhält diese Richtlinie eine bessere Kompatibilität und die Initialisierung ist weniger aufwändig als bei der Richtlinie Biba.

17.14.1. Beispiele

Wie schon bei den Richtlinien Biba und MLS werden die Befehle `setfmac` und `setpmac` verwendet, um die Labels an den Systemobjekten zu setzen:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Beachten Sie, dass hier der Hilfswert auf `low` gesetzt wurde - dieses Leistungsmerkmal ist nur in der MAC `LOMAC` Richtlinie enthalten.

17.15. Beispiel 1: Nagios in einer MAC Jail

Die folgende Demonstration setzt eine sichere Umgebung mithilfe verschiedener MAC Module und sorgfältig konfigurierter Richtlinien um. Es handelt sich jedoch nur um einen Test und sollte nicht als Antwort auf jedes Problem in Fragen Sicherheit gesehen werden. Eine Richtlinie nur umzusetzen und dann einfach laufen zu lassen, funktioniert nie und kann eine echte Arbeitsumgebung in eine Katastrophe stürzen.

Bevor es losgeht, muß jedes Dateisystem mit der Option `multilabel`, wie weiter oben beschrieben, markiert werden. Dies nicht zu tun, führt zu Fehlern. Außerdem müssen die Ports `net-mngt/nagios-plugins`, `net-mngt/nagios` und `www/apache13` installiert und konfiguriert sein, so dass sie ordentlich laufen.

17.15.1. Erstellen einer Nutzerklasse `insecure`

Beginnen wir die Prozedur mit dem Hinzufügen einer Nutzerklasse in der Datei `/etc/login.conf`:

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin--
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=biba/10(10-10):
```

Zusätzlich fügen wir beim Standardnutzer folgende Zeile hinzu:

```
:label=biba/high:
```

Anschließend muß die Datenbank neu erstellt werden:

```
# cap_mkdb /etc/login.conf
```

17.15.2. Boot-Konfiguration

Starten Sie den Rechner noch nicht neu. Fügen Sie zunächst noch die folgenden Zeilen in die Datei `/boot/loader.conf` ein, damit die benötigten Module während des Systemstarts geladen werden:

```
mac_biba_load="YES"
mac_seeotheruids_load="YES"
```

17.15.3. Nutzer einrichten

Ordnen Sie den Superuser `root` der Klasse `default` zu:

```
# pw usermod root -L default
```

Alle Nutzerkonten, die weder `root` noch Systemkonten sind, brauchen nun eine Loginklasse, da sie sonst keinen Zugriff auf sonst übliche Befehle erhalten, wie bspw. vi(1). Das folgende `sh` Skript wird diese Aufgabe erledigen:

```
# for x in ‘awk -F: ’($3 >= 1001) && ($3 != 65534) { print $1 }’ \
      /etc/passwd‘; do pw usermod $x -L default; done;
```

Verschieben Sie die Nutzer `nagios` und `www` in die `insecure` Klasse:

```
# pw usermod nagios -L insecure

# pw usermod www -L insecure
```

17.15.4. Die Kontextdatei erstellen

Nun muß eine Kontextdatei erstellt werden. Die folgende Beispieldatei soll dazu in `/etc/policy.contexts` gespeichert werden:

```
# This is the default BIBA policy for this system.

# System:
/var/run                        biba/equal
/var/run/*                      biba/equal

/dev                            biba/equal
/dev/*                          biba/equal

/var                           biba/equal
```

```
/var/spool                      biba/equal
/var/spool/*                    biba/equal

/var/log                        biba/equal
/var/log/*                      biba/equal

/tmp                           biba/equal
/tmp/*                         biba/equal
/var/tmp                       biba/equal
/var/tmp/*                     biba/equal

/var/spool/mqueue                      biba/equal
/var/spool/clientmqueue        biba/equal

# For Nagios:
/usr/local/etc/nagios
/usr/local/etc/nagios/*         biba/10

/var/spool/nagios               biba/10
/var/spool/nagios/*             biba/10

# For apache
/usr/local/etc/apache           biba/10
/usr/local/etc/apache/*         biba/10
```

Die Richtlinie erzwingt Sicherheit, indem der Informationsfluß Einschränkungen unterworfen wird. In der vorliegenden Konfiguration kann kein Nutzer, weder `root` noch andere, auf **Nagios** zugreifen. Konfigurationsdateien und die Prozesse, die Teil von **Nagios** sind, werden durch unsere MAC vollständig abgegrenzt.

Die Kontextdatei kann nun vom System eingelesen werden, indem folgender Befehl ausgeführt wird:

```
# setfmac -ef /etc/policy.contexts /
# setfmac -ef /etc/policy.contexts /
```

> **Anmerkung:** Das obenstehende Dateisystem-Layout kann, je nach Umgebung, sehr unterschiedlich aussehen. Außerdem muß es auf jedem einzelnen Dateisystem ausgeführt werden.

In die Datei `/etc/mac.conf` müssen nun noch diese Änderungen eingetragen werden:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

17.15.5. Netzwerke einbinden

Tragen Sie die folgende Zeile in die Datei `/boot/loader.conf` ein:

```
security.mac.biba.trust_all_interfaces=1
```

Und das Folgende gehört in Datei `rc.conf` zu den Optionen für die Netzwerkkarte. Falls die Netzwerkverbindung(-en) via DHCP konfiguriert werden, muß man dies nach jedem Systemstart eigenhändig nachtragen:

```
maclabel biba/equal
```

17.15.6. Testen der Konfiguration

Versichern Sie sich, dass der Webserver und **Nagios** nicht automatisch geladen werden und starten Sie den Rechner neu. Prüfen Sie nun, ob `root` wirklich keinen Zugriff auf die Dateien im Konfigurationsverzeichnis von **Nagios** hat. Wenn `root` den Befehl ls(1) auf `/var/spool/nagios` ausführen kann, ist irgendwas schief gelaufen. Es sollte ein `permission denied` Fehler ausgegeben werden.

Wenn alles gut aussieht, können **Nagios**, **Apache** und **Sendmail** gestartet werden - allerdings auf eine Weise, die unserer Richtlinie gerecht wird. Zum Beispiel durch die folgenden Kommandos:

```
# cd /etc/mail && make stop && \
setpmac biba/equal make start && setpmac biba/10\(10-10\) apachectl start && \
setpmac biba/10\(10-10\) /usr/local/etc/rc.d/nagios.sh forcestart
```

Versichern Sie sich lieber doppelt, dass alles ordentlich läuft. Wenn nicht, prüfen Sie die Logs und Fehlermeldungen. Verwenden Sie das sysctl(8) Werkzeug um die Sicherheitsrichtlinie sysctl(8) zu deaktivieren und versuchen Sie dann alles noch einmal zu starten.

> **Anmerkung:** Der Superuser kann den Vollzug der Richtlinie schalten und die Konfiguration ohne Furcht verändern. Folgender Befehl stuft eine neu gestartete Shell herunter:
>
> ```
> # setpmac biba/10 csh
> ```
>
> Um dies zu vermeiden, werden die Nutzer durch login.conf(5) eingeschränkt. Wenn setpmac(8) einen Befehl außerhalb der definierten Schranken ausführen soll, wird ein Fehler zurückgeliefert. In so einem Fall muß `root` auf `biba/high(high-high)` gesetzt werden.

17.16. Beispiel 2: User Lock Down

Grundlage dieses Beispiels ist ein relativ kleines System zur Datenspeicherung mit weniger als 50 Benutzern. Diese haben die Möglichkeit, sich einzuloggen und dürfen nicht nur Daten speichern, sondern auch auf andere Ressourcen zugreifen.

Die Richtlinien mac_bsdextended(4) und mac_seeotheruids(4) können gleichzeitig eingesetzt werden. Zusammen kann man mit ihnen nicht nur den Zugriff auf Systemobjekte einschränken, sondern auch Nutzerprozesse verstecken.

Beginnen Sie, indem Sie die folgende Zeile in die Datei `/boot/loader.conf` eintragen:

```
mac_seeotheruids_load="YES"
```

Die Richtlinie mac_bsdextended(4) wird durch den anschließenden Eintrag in `/etc/rc.conf` hinzugefügt:

```
ugidfw_enable="YES"
```

Die Standardregeln, welche in `/etc/rc.bsdextended` gespeichert sind, werden zum Systemstart geladen. Sie müssen aber noch angepaßt werden. Da dieser Computer nur Nutzern dienen soll und weitere Dienste gestartet werden, kann alles bis auf die beiden letzten Zeilen auskommentiert werden. Das sorgt dafür dass jeder Nutzer seine eigenen Systemobjekte erhält.

Nun fügen wir alle benötigten Nutzer auf der Maschine hinzu und starten neu. Zum Testen der Einstellungen loggen Sie sich parallel zwei mal mit unterschiedlichen Nutzernamen ein und starten Sie das Kommando `ps aux`. Dort sehen Sie, dass Sie die Prozesse des anderen Nutzers nicht sehen können. Versuchen Sie, ls(1) auf das Heimatverzeichnis eines anderen Nutzers auszuführen. Auch dieser Versuch wird fehlschlagen.

Solange nicht die speziellen `sysctl`-Variablen geändert wurden, hat der Superuser noch vollen Zugriff. Sobald auch diese Einstellungen angepaßt wurden, führen Sie ruhig auch den obigen Test als `root` aus.

> **Anmerkung:** Wenn ein neuer Benutzer hinzugefügt wird, ist für diesen zunächst keine mac_bsdextended(4) Regel im Regelsatz vorhanden. Schnelle Abhilfe schafft hier, einfach das Kernelmodul mit kldunload(8) zu entladen und mit kldload(8) erneut einzubinden.

17.17. Fehler im MAC beheben

Während der Entwicklung des Frameworks haben einige Nutzer auf Probleme hingewiesen. Einige davon werden hier aufgeführt:

17.17.1. Die Option `multilabel` greift nicht auf der `/`-Partition

Es scheint, dass etwa jedem fünfzigsten Nutzer dieses Problem widerfährt. Und in der Tat - auch wir kennen es aus der Entwicklung. Genauere Untersuchungen dieses “Bugs” machten uns glauben, dass es sich entweder um einen Fehler in oder eine fehlerhafte Interpretation der Dokumentation handelt. Warum auch immer dieser Fehler auftritt - er kann mit folgender Prozedur behoben werden:

1. Öffnen Sie die Datei `/etc/fstab` und setzen Sie die Rootpartition auf `ro` wie “read-only”.
2. Starten Sie in den Einzelnutzermodus.
3. Rufen Sie `tunefs -l enable` für `/` auf.
4. Starten Sie in den Mehrbenutzermodus.
5. Führen Sie `mount -urw /` aus und ändern Sie anschließend in der Datei `/etc/fstab` die Option `ro` zurück in `rw`. Starten Sie das System noch einmal neu.
6. Achten Sie besonders auf die Ausgabe von `mount` um sich zu versichern, dass die `multilabel` korrekt für das root-Dateisystem gesetzt wurde.

17.17.2. Mit der aktivierten MAC kann ich keinen X11 Server starten

Dies kann durch die Richtlinie `partition` oder einer fehlerhaften Verwendung einer Richtlinie, die mit Labels arbeitet, auftreten. Zum debuggen versuchen Sie folgendes:

1. Schauen Sie sich die Fehlermeldungen genau an. Wenn der Nutzer einer `insecure` Klasse angehört, ist wahrscheinlich die Richtlinie `partition` die Ursache. Versuchen Sie, die Nutzerklasse auf `default` zu stellen und danach die Datenbank mit `cap_mkdb` zu erneuern. Wenn das Problem dadurch nicht gelöst wird, gehen Sie weiter zu Schritt 2.
2. Gehen Sie die Label-Richtlinien Schritt für Schritt nocheinmal durch. Achten Sie darauf, dass für den Nutzer, bei dem das Problem auftritt, für X11 und das Verzeichnis `/dev` alle Einstellungen korrekt sind.
3. Falls all dies nicht helfen sollte, senden Sie die Fehlermeldung und eine Beschreibung ihrer Arbeitsumgebung an die (englisch-sprachige) TrustedBSD Diskussionsliste auf der TrustedBSD (http://www.TrustedBSD.org) Webseite oder an die FreeBSD general questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions) Mailingliste.

17.17.3. Error: _secure_path(3) cannot stat `.login_conf`

Wenn ich versuche, von `root` zu einem anderen Nutzer des Systems zu wechseln, erhalte ich die Fehlermeldung `_secure_path: unable to state .login_conf`.

Diese Meldung wird gewöhnlich ausgegeben, wenn der Nutzer ein höhere Label-Einstellung hat als der, dessen Identität man annehmen möchte. Ausführlich: Wenn ein Nutzer `joe` als `biba/low` gelabelt wurde, kann `root`, der `biba/high` als Voreinstellung trägt, das Heimatverzeichnis von `joe` nicht einsehen. Das passiert unabhänig davon, ob `root` vorher mit `su` die Identität von `joe` angenommen hat oder nicht, da das Label sich nicht ändert. Hier haben wir also einen Fall, in dem das Gewährleistungsmodell von Biba verhindert, das der Superuser Objekte einer niedrigeren Integrität betrachten kann.

17.17.4. Der Nutzer `root` ist kaputt!

Im normalen oder sogar im Einzelbenutzermodus wird `root` nicht anerkannt. Das Kommando `whoami` liefert 0 (null) und `su` liefert `who are you?` zurück. Was geht da vor?

Das kann passieren, wenn eine Label-Richtlinie ausgeschaltet wird - entweder durch sysctl(8) oder wenn das Richtlinienmodul entladen wurde. Wenn eine Richtlinie deaktiviert oder auch nur vorübergehen deaktiviert wird, muß die Befähigungsdatenbank neu konfiguriert werden, d.h. die `label` Option muß entfernt werden. Überprüfen Sie, ob alle `label` Einträge aus der Datei `/etc/login.conf` entfernt wurden und bauen Sie die Datenbank mit `cap_mkdb` neu.

Dieser Fehler kann auch auftreten, wenn eine Richtlinie den Zugriff auf die Datei `master.passwd` einschränkt. Normalerweise passiert das nur, wenn ein Administrator ein Label an diese Datei vergibt, das mit der allgemeingültigen Richtlinie, die das System verwendet, in Konflikt steht. In solchen Fällen werden die Nutzerinformationen vom System ausgelesen und jeder weitere Zugriff wird blockiert, sobald das neue Label greift. Wenn man die Richtlinie via sysctl(8) ausschaltet, sollte es erstmal wieder gehen.

Fußnoten

1. Andere Vorbedingungen führen natürlich zu anderen Fehlern. Zum Beispiel wenn das Objekt nicht dem Nutzer gehört, der das Label ändern möchte, das Objekt vielleicht gar nicht existiert oder es sich um ein nur lesbares Objekt handelt. Oder eine verbindliche Richtlinie erlaubt dem Prozeß die Veränderung des Labels nicht, weil die Eigenschaften der Datei, die Eigenschaften des Prozesses oder der Inhalt des neuen Labels nicht akzeptiert werden. Beispiel: Ein Anwender mit geringer Vertraulichkeit versucht, das Label einer Datei mit hoher Vertraulichkeit zu ändern. Oder er versucht, eine Datei mit geringer Vertraulichkeit zu einer Datei mit hoher Vertraulichkeit zu machen.

Kapitel 18. Security Event Auditing

Geschrieben von Tom Rhodes und Robert Watson. Übersetzt von Daniel Seuffert.

18.1. Einleitung

Das FreeBSD-Betriebssystem unterstützt ein feingranuliertes Sicherheits-Auditing. Ereignis-Auditing erlaubt die zuverlässige, feingranulierte und konfigurierbare Aufzeichnung einer Vielzahl von sicherheitsrelevanten Systemereignissen einschliesslich Benutzereingaben, Konfigurationsänderungen sowie Datei- und Netzwerkzugriffen. Diese Log-Datensätze können unschätzbar wertvoll sein für direkte Systemüberwachung, Einbruchserkennung und Post-Mortem-Analyse. FreeBSD implementiert Suns öffentlich zugängliche BSM API und Dateiformat. Die FreeBSD-Implementierung kann mit den Audit-Implementierungen von Sun Solaris und Apple Mac OS X zusammenarbeiten.

Dieses Kapitel konzentriert sich auf die Installation und Konfiguration des Ereignis-Auditings. Es erklärt Audit-Richtlinien und stellt ein Beispiel einer Audit-Konfiguration vor.

Nach dem Lesen dieses Kapitels werden Sie Folgendes wissen:

- Was Ereignis-Auditing ist und wie es arbeitet.
- Wie man Ereignis-Auditing in FreeBSD für Benutzer und Prozesse konfiguriert.
- Wie man den Audit-Pfad mittels Audit-Reduktion und Revisionswerkzeugen überprüft.

Vor dem Lesen dieses Kapitels sollten Sie:

- Sowohl UNIX als auch FreeBSD-Basismechanismen beherrschen (Kapitel 4).
- Mit den grundlegenden Mechanismen der Kernel-Konfiguration und -Kompilierung vertraut sein (Kapitel 9).
- Mit den Maßnahmen zur Sicherung von FreeBSD vertraut sein (Kapitel 15).

> **Warnung:** Die Audit-Funktionalität in FreeBSD besitzt die Einschränkungen, dass zur Zeit nicht alle sicherheitsrelevanten System-Ereignisse auditierbar sind und dass einige Anmelde-Mechanismen, wie z.B. X11-basierte Bildschirm-Manager und Daemonen von Drittanbietern, das Auditing für Benutzeranmeldungen nicht korrekt konfigurieren.
>
> Das Sicherheits-Auditing ist in der Lage, sehr detaillierte Log-Dateien von Systemaktivitäten zu erzeugen. Auf einem ausgelasteten System kann die Pfad-Datei sehr groß werden, wenn sie für hohe Auflösung konfiguriert ist, und im Extremfall pro Woche um mehrere Gigabyte anwachsen. Administratoren sollten daher den benötigten Plattenplatz in Verbindung mit umfangreichen Audit-Konfigurationen berücksichtigen. So kann es wünschenswert sein, ein eigenes Dateisystem für `/var/audit` einzusetzen, damit andere Dateisysteme nicht betoffen sind, wenn das Dateisystem des Audit voll läuft.

18.2. Schlüsselbegriffe

Vor dem Lesen dieses Kapitels müssen einige Audit-bezogene Schlüsselbegriffe erläutert werden:

- *event*: Ein auditierbares Ereignis ist ein Ereignis, das mit dem Audit-Subsystem aufgezeichnet werden kann. Beispiele für sicherheitsrelevante Systemereignisse sind etwa das Anlegen von Dateien, das Erstellen einer Netzwerkverbindung oder eine Benutzeranmeldung. Ereignisse sind entweder “attributierbar”, können also zu einen authentifizierten Benutzer zurückverfolgt werden, oder sind “nicht-attributierbar”, falls dies nicht möglich ist. Nicht-attributierbare Ereignisse erfolgen daher vor der Authentifizierung im Anmeldeprozess (beispielsweise die Eingabe eines falschen Passworts).
- *class*: Ereignisklassen sind benannte Zusammenstellungen von zusammengehörenden Ereignissen und werden in Auswahl-Ausdrücken benutzt. Häufig genutzte Klassen von Ereignissen schließen “file creation” (fc, Anlegen von Dateien), “exec” (ex, Ausführung) und “login_logout” (lo, Anmeldung-Abmeldung) ein.
- *record*: Ein Datensatz ist ein Audit-Logeintrag, welcher ein Sicherheitsereignis enthält. Jeder Datensatz enthält einen Ereignistyp, Informationen über den Gegenstand (Benutzer), welcher die Aktion durchführt, Datums- und Zeitinformationen, Informationen über jedes Objekt oder Argument sowie den Zustand hinsichtlich Erfolg oder Scheitern der Operation.
- *trail*: Ein Audit-Pfad (audit trail) oder eine Log-Datei besteht aus einer Reihe von Audit-Datensätzen, die Sicherheitsereignisse beschreiben. Normalerweise sind die Pfade in grober zeitlicher Reihenfolge bezüglich des Zeitpunktes, an welchem ein Ereignis beendet wurde. Nur authorisierte Prozesse dürfen Datensätze zum Audit-Pfad hinzufügen.
- *selection expression*: Ein Auswahlausdruck ist eine Zeichenkette, welche eine Liste von Präfixen und Audit-Ereignisklassennamen enthält, um Ereignisse abzugleichen.
- *preselection*: Die Vorauswahl ist der Prozess, durch den das System erkennt, welche Ereignisse von Interesse für den Administrator sind, um die Erzeugung von Datensätze zu verhindern, welche nicht von Belang sind. Die Konfiguration der Vorauswahl benutzt eine Reihe von Auswahl-Ausdrücken, um zu erkennen, welche Klassen von Ereignissen für welche Benutzer aufgezeichnet werden sollen sowie globale Einstellungen, welche sowohl auf authorisierte als auch unauthorisierte Prozesse angewendet werden.
- *reduction*: Die Reduzierung ist der Prozess, durch den Datensätze von bestehenden Audit-Pfaden ausgewählt werden für Speicherung, Ausdruck oder Analyse. Ebenso der Prozess, durch den unerwünschte Datensätze aus dem Audit-Pfad entfernt werden. Mittels Reduzierung können Administratoren Richtlinien für die Speicherung von Audit-Daten vorgeben. Zum Beispiel können ausführliche Audit-Pfade für einen Monat gespeichert werden, um danach den Pfad für archivarische Zwecke auf die Anmeldeinformationen zu reduzieren.

18.3. Installation der Audit-Unterstützung

Die Unterstützung des Ereignis-Auditings für den Benutzerbereich wird bereits als Teil des Basissystems installiert. Die Audit-Unterstützung ist bereits im FreeBSD-Standardkernel enthalten, jedoch müssen Sie die folgende Zeile explizit in Ihre Kernelkonfigurationsdatei aufnehmen und den Kernel neu bauen:

```
options AUDIT
```

Bauen und installieren Sie den Kernel wie in Kapitel 9 beschrieben ist.

Nachdem der Kernel mit Audit-Unterstützung gebaut und installiert ist und das System neu gestartet wurde, aktivieren Sie den Audit-Daemon durch das Einfügen der folgenden Zeile in die Datei rc.conf(5):

```
auditd_enable="YES"
```

Die Audit-Unterstützung kann nun durch einen Neustart des Systems oder durch das manuelle Starten des Audit-Daemon aktiviert werden:

```
/etc/rc.d/auditd start
```

18.4. Die Konfiguration des Audit

Alle Konfigurationsdateien für das Sicherheits-Audit finden sich unter `/etc/security`. Die folgenden Dateien müssen vorhanden sein, bevor der Audit-Daemon gestartet wird:

- `audit_class` – Enthält die Definitionen der Audit-Klassen.
- `audit_control` – Steuert Teile des Audit-Subsystems wie Audit-Klassen, minimaler Plattenplatz auf dem Audit-Log-Datenträger, maximale Größe des Audit-Pfades usw.
- `audit_event` – Wörtliche Namen und Beschreibungen von System-Audit-Ereignissen sowie eine Liste, welche Klassen welches Ereignis aufzeichnen.
- `audit_user` – Benutzerspezifische Audit-Erfordernisse, welche mit den globalen Vorgaben bei der Anmeldung kombiniert werden.
- `audit_warn` – Ein anpassbares Shell-Skript, welches von **auditd** benutzt wird, um Warnhinweise in aussergewöhnlichen Situationen zu erzeugen, z.B. wenn der Platz für die Audit-Datensätze knapp wird oder wenn die Datei des Audit-Pfades rotiert wurde.

> **Warnung:** Audit-Konfigurationsdateien sollten vorsichtig gewartet und bearbeitet werden, da Fehler in der Konfiguration zu falscher Aufzeichnung von Ereignissen führen könnten.

18.4.1. Ereignis-Auswahlausdrücke

Auswahlausdrücke werden an einigen Stellen der Audit-Konfiguration benützt, um zu bestimmen, welche Ereignisse auditiert werden sollen. Die Ausdrücke enthalten eine Liste der Ereignisklassen, welche verglichen werden sollen, jede mit einem Präfix, welches anzeigt, ob verglichene Datensätze akzeptiert oder ignoriert werden sollen und optional, um anzuzeigen, ob der Eintrag beabsichtigt, erfolgreiche oder fehlgeschlagene Operationen zu vergleichen. Auswahlausdrücke werden von links nach rechts ausgewertet und zwei Ausdrücke werden durch Aneinanderhängen miteinander kombiniert.

Die folgende Liste enthält die Standard-Ereignisklassen für das Audit und ist in `audit_class` festgehalten:

- `all` – *all* – Vergleiche alle Ereignisklassen.
- `ad` – *administrative* – Administrative Aktionen ausgeführt auf dem System als Ganzes.
- `ap` – *application* – Aktionen definiert für Applikationen.
- `cl` – *file close* – Audit-Aufrufe für den Systemaufruf `close`.
- `ex` – *exec* – Ausführung des Audit-Programms. Auditierung von Befehlszeilen-Argumenten und Umgebungsvariablen wird gesteuert durch audit_control(5) mittels der `argv` und `envv`-Parametergemäss der `Richtlinien`-Einstellungen.

- `fa` – *file attribute access* – Auditierung des Zugriffs auf Objektattribute wie stat(1), pathconf(2) und ähnlichen Ereignissen.
- `fc` – *file create* – Audit-Ereignisse, bei denen eine Datei als Ergebnis angelegt wird.
- `fd` – *file delete* – Audit-Ereignisse, bei denen Dateilöschungen vorkommen.
- `fm` – *file attribute modify* – Audit-Ereignisse, bei welchen Dateiattribute geändert werden, wie chown(8), chflags(1), flock(2) etc.
- `fr` – *file read* – Audit-Ereignisse, bei denen Daten gelesen oder Dateien zum lesen geöffnet werden usw.
- `fw` – *file write* – Audit-Ereignisse, bei welchen Daten geschrieben oder Dateien geschrieben oder verändert werden usw.
- `io` – *ioctl* – Nutzung des Systemaufrufes ioctl(2) durch Audit.
- `ip` – *ipc* – Auditierung verschiedener Formen von Inter-Prozess-Kommunikation einschliesslich POSIX-Pipes und System V IPC-Operationen.
- `lo` – *login_logout* – Audit-Ereignisse betreffend login(1) und logout(1), welche auf dem System auftreten.
- `na` – *non attributable* – Auditierung nicht-attributierbarer Ereignisse (Ereignisse, die nicht auf einen bestimmten Benutzer zurückgeführt werden können).
- `no` – *invalid class* – Kein Abgleich von Audit-Ereignissen.
- `nt` – *network* – Audit-Ereignisse in Zusammenhang mit Netzwerkaktivitäten wie z.B. connect(2) und accept(2).
- `ot` – *other* – Auditierung verschiedener Ereignisse.
- `pc` – *process* – Auditierung von Prozess-Operationen wie exec(3) und exit(3).

Diese Ereignisklassen können angepasst werden durch Modifizierung der Konfigurationsdateien `audit_class` und `audit_event`.

Jede Audit-Klasse in dieser Liste ist kombiniert mit einem Präfix, welches anzeigt, ob erfolgreiche/gescheiterte Operationen abgebildet werden, und ob der Eintrag den Abgleich hinzufügt oder entfernt für die Klasse und den Typ.

- (none) Kein Präfix, sowohl erfolgreiche als auch gescheiterte Vorkommen eines Ereignisses werden auditiert.
- `+` Auditiere nur erfolgreiche Ereignisse in dieser Klasse.
- `-` Auditiere nur gescheiterte Operationen in dieser Klasse.
- `^` Auditiere weder erfolgreiche noch gescheiterte Ereignisse in dieser Klasse.
- `^+` Auditiere keine erfolgreichen Ereignisse in dieser Klasse.
- `^-` Auditiere keine gescheiterten Ereignisse in dieser Klasse.

Das folgende Beispiel einer Auswahl-Zeichenkette wählt erfolgreiche und gescheiterte Anmelde/Abmelde-Ereignisse aus, aber nur erfolgreich beendete Ausführungs-Ereignisse:

```
lo,+ex
```

18.4.2. Konfigurationsdateien

In den meisten Fällen müssen Administratoren nur zwei Dateien ändern, wenn sie das Audit-System konfigurieren: `audit_control` und `audit_user`. Die erste Datei steuert systemweite Audit-Eigenschaften und -Richtlinien; die zweite Datei kann für die Feinanpassung der Auditierung von Benutzern verwendet werden.

18.4.2.1. Die `audit_control`-Datei

Die `audit_control`-Datei legt eine Anzahl Vorgabewerte fest. Beim Betrachten des Inhaltes der Datei sehen wir Folgendes:

```
dir:/var/audit
flags:lo
minfree:20
naflags:lo
policy:cnt
filesz:0
```

Die Option `dir` wird genutzt, um eines oder mehrere Verzeichnisse festzulegen, in welchen Audit-Protokolle gespeichert werden. Gibt es mehrere Verzeichniseinträge, werden diese in der angegebenen Reihenfolge genutzt, bis sie jeweils gefüllt sind. Es ist üblich, Audit so zu konfigurieren, dass die Audit-Logs auf einem dedizierten Dateisystem abgelegt werden, um Wechselwirkungen zwischen dem Audit-Subsystem und anderen Subsystemen zu verhindern, falls das Dateisystem voll läuft.

Das `flags`-Feld legt die systemweite Standard-Vorauswahl-Maske für attributierbare (direkt einem Benutzer zuordenbare) Ereignisse fest. Im obigen Beispiel werden alle gescheiterten und erfolgreichen Anmelde- und Abmelde-Ereignisse für alle Benutzer aufgezeichnet.

Die Option `minfree` definiert den minimalen Prozentsatz an freiem Plattenplatz für das Dateisystem, in welchem der Audit-Pfad abgespeichert wird. Wenn diese Schwelle überschritten ist, wird ein Warnhinweis erzeugt. Das obige Beispiel legt den minimalen freien Platz auf zwanzig Prozent fest.

Die `naflags`-Option bestimmt diejenigen Audit-Klassen, für die nicht-attributierbare Ereignisse aufgezeichnet werden sollen (beispielsweise Anmeldeprozesse und System-Daemonen.

Die Option `policy` legt eine durch Kommata getrennte Liste von policy-Flags fest, welche verschiedene Aspekte des Audit-Verhaltens steuern. Der vorgegebene Flag `cnt` zeigt an, dass das System trotz eines Audit-Fehlers weiterlaufen soll (dieses Flag wird dringend angeraten). Ein anderes, häufig genutztes Flag ist `argv`, welches dazu führt, dass Befehlszeilen-Argumente für den Systemauruf execve(2) als Teil der Befehlsausführung aufgezeichnet werden.

Die `filesz`-Option spezifiziert die maximale Größe in Bytes, welche eine Audit-Pfad-Datei wachsen darf, bevor sie automatisch beendet und rotiert wird. Die Standardvorgabe `0` setzt die automatische Log-Rotation ausser Kraft. Falls die angeforderte Dateigröße größer Null und gleichzeitig unterhalb des Minimums von 512K ist, dann wird die Angabe verworfen und ein Log-Hinweis wird erzeugt.

18.4.2.2. Die Datei `audit_user`

Die `audit_user`-Datei erlaubt es dem Administrator, weitere Audit-Erfordernisse für bestimmte Benutzer festzulegen. Jede Zeile konfiguriert das Auditing für einen Benutzer über zwei Felder: Das erste Feld ist `alwaysaudit`, welches eine Ansammlung von Ereignissen vorgibt, welche immer für diesen Benutzer aufgezeichnet werden. Das zweite Feld `neveraudit` legt eine Menge an Ereignissen fest, die niemals für diesen Benutzer auditiert werden sollen.

Das folgende Beispiel einer `audit_user`-Datei zeichnet Anmelde/Abmelde-Ereignisse, erfolgreiche Befehlsausführungen für den Benutzer `root`, Anlegen von Dateien und erfolgreiche Befehlsausführungen für den Benutzer `www` auf. Falls das Beispiel zusammen mit der vorstehend als Beispiel gezeigten Datei `audit_control` benutzt wird, dann ist der Eintrag `lo` für Benutzer `root` überflüssig und Anmelde/Abmelde-Ereignisse werden für den Benutzer `www` ebenfalls aufgezeichnet.

```
root:lo,+ex:no
www:fc,+ex:no
```

18.5. Administration des Audit-Subsystems

18.5.1. Audit-Pfade betrachten

Audit-Pfade werden im binären BSM-Format gespeichert, daher benötigen Sie spezielle Werkzeuge, um derartige Dateien zu ändern oder Sie in Textdateien zu konvertieren. Der Befehl praudit(1) wandelt alle Pfad-Dateien in ein einfaches Textformat um. Der Befehl auditreduce(1) kann genutzt werden, um die Pfad-Dateien für Analyse, Ausdruck, Archivierung oder andere Zwecke zu reduzieren. `auditreduce` unterstützt eine Reihe von Auswahl-Parametern einschliesslich Ereignistyp, Ereignisklasse, Benutzer, Datum oder Uhrzeit des Ereignisses und den Dateipfad oder das Objekt, mit dem gearbeitet wurde.

Das Dienstprogramm `praudit` schreibt zum Beispiel den gesamten Inhalt einer angegebenen Audit-Protokolldatei in eine simple Textdatei:

```
# praudit /var/audit/AUDITFILE
```

`AUDITFILE` ist hier die zu schreibende Protokolldatei.

Audit-Pfade bestehen aus einer Reihe von Datensätzen, die wiederum aus Kürzeln (token) gebildet werden, die von `praudit` fortlaufend zeilenweise ausgegeben werden. Jedes Kürzel ist von einem bestimmten Typ, z.B. enthält `header` einen audit-Datensatz-Header oder `path` enthält einen Dateipfad von einer Suche. Hier ein Beispiel eines `execve`-Ereignisses:

```
header,133,10,execve(2),0,Mon Sep 25 15:58:03 2006, + 384 msec
exec arg,finger,doug
path,/usr/bin/finger
attribute,555,root,wheel,90,24918,104944
subject,robert,root,wheel,root,wheel,38439,38032,42086,128.232.9.100
return,success,0
trailer,133
```

Dieser Audit stellt einen erfolgreichen `execve`-Aufruf dar, in welchem der Befehl `finger doug` ausgeführt wurde. Das Kürzel des Argumentes enthält die Befehlszeile, welche die Shell an den Kernel weiterleitet. Das Kürzel `path` enthält den Pfad zur ausführbaren Datei (wie vom Kernel wahrgenommen). Das Kürzel `attribute` beschreibt die Binärdatei (insbesondere den Datei-Modus, der genutzt werden kann, um zu bestimmen, ob setuid auf die Applikation angewendet wurde). Das Kürzel `subject` beschreibt den untergeordneten Prozess und speichert daher in Aufeinanderfolge Audit-Benutzer-ID, effektive Benutzer-ID und Gruppen-ID, wirkliche Benutzer-ID und Grppen-ID, Process-ID, Session- ID, Port-ID und Anmelde-Adresse. Beachten Sie, dass Audit-Benutzer-ID und wirkliche Benutzer-ID abweichen: Der Benutzer `robert` wurde zum Benutzer `root`, bevor er diesen Befehl

ausführte, aber er wird auditiert mit dem ursprünglich authentifizierten Benutzer. Schließlich zeigt das Kürzel `return` die erfolgreiche Ausführung an und `trailer` schließt den Datensatz ab.

`praudit` unterstützt auch die Ausgabe im XML-Format (die sie über die Option `-x` auswählen können).

18.5.2. Audit-Pfade reduzieren

Da Audit-Protokolldateien sehr groß sein können, wird ein Administrator höchstwahrscheinlich eine Auswahl an Datensätzen verwenden, wie z.B. alle Datensätze zu einem bestimmten Benutzer:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Dies wird alle Audit-Datensätze des Benutzers `trhodes` auswählen, die in der Datei *`AUDITFILE`* gespeichert sind.

18.5.3. Delegation von Rechten für Audit-Reviews

Mitglieder der Gruppe `audit` haben die Erlaubnis, Audit-Pfade in `/var/audit` zu lesen; standardmässig ist diese Gruppe leer, daher kann nur der Benutzer `root` die Audit-Pfade lesen. Benutzer können der Gruppe `audit` hinzugefügt werden, um Rechte für Audit-Reviews zu gewähren. Da die Fähigkeit, Inhalte von Audit-Protokolldateien zu verfolgen, tiefgreifende Einblicke in das Verhalten von Benutzern und Prozessen erlaubt, wird empfohlen, dass die Gewährung von Rechten für Audit-Reviews mit Bedacht erfolgt.

18.5.4. Aktive Überwachung mittles Audit-Pipes

Audit-Pipes sind nachgebildete (geklonte) Pseudo-Geräte im Dateisystem des Gerätes, welche es Applikationen erlauben, die laufenden Audit-Datensätze anzuzapfen. Dies ist vorrangig für Autoren von Intrusion Detection Software und Systemüberwachungsprogrammen von Bedeutung. Allerdings ist für den Administrator das Audit-Pipe-Gerät ein angenehmer Weg, aktive Überwachung zu gestatten, ohne Gefahr von Problemen durch Besitzerrechte der Audit-Pfad-Datei oder Unterbrechung des Stroms von Ereignissen durch Log-Rotation. Um den laufenden Audit-Ereignisstrom zu verfolgen, geben Sie bitte folgende Befehlszeile ein:

```
# praudit /dev/auditpipe
```

In der Voreinstellung kann nur der Benutzer `root` auf die Audit-Pipe-Geräte-Knotenpunkte zugreifen. Um sie allen Mitgliedern der Gruppe `audit` zugänglich zu machen, fügen Sie eine `devfs`-Regel in `devfs.rules` hinzu:

```
add path 'auditpipe*' mode 0440 group audit
```

Lesen Sie devfs.rules(5) für weitere Informationen, wie das devfs-Dateisystem konfiguriert wird.

> **Warnung:** Es ist sehr leicht, Rückmeldungszyklen von Audit-Ereignissen hervorzurufen, in welcher das Betrachten des Resultates eines Audit-Ereignisses in die Erzeugung von mehr Audit-Ereignissen mündet. Wenn zum Beispiel der gesamte Netzwerk-I/O auditiert wird, während praudit(1) in einer SSH-Sitzung gestartet wurde, dann wird ein kontinuierlicher, mächtiger Strom von Audit-Ereignissen erzeugt, da jedes ausgegebene Ereignis wiederum neue Ereignisse erzeugt. Es ist anzuraten, `praudit` an einem Audit-Pipe-Gerät nur von Sitzungen anzuwenden (ohne feingranuliertes I/O-Auditing), um dies zu vermeiden.

18.5.5. Rotation von Audit-Pfad-Dateien

Audit-Pfade werden nur vom Kernel geschrieben und nur vom Audit-Daemon **auditd** verwaltet. Administratoren sollten nicht versuchen, newsyslog.conf(5) oder andere Werkzeuge zu benutzen, um Audit-Protokolldateien direkt zu rotieren. Stattdessen sollte das `audit` Management-Werkzeug benutzt werden, um die Auditierung zu beenden, das Audit-System neu zu konfigurieren und eine Log-Rotation durchzuführen. Der folgende Befehl veranlasst den Audit-Daemon, eine neue Protokolldatei anzulegen und dem Kernel zu signalisieren, die neue Datei zu nutzen. Die alte Datei wird beendet und umbenannt. Ab diesem Zeitpunkt kann sie vom Administrator bearbeitet werden.

```
# audit -n
```

> **Warnung:** Falls der **auditd**-Daemon gegenwärtig nicht läuft, wird dieser Befehl scheitern und eine Fehlermeldung wird ausgegeben.

Das Hinzufügen der folgenden Zeile in `/etc/crontab` wird die Log-Rotation alle zwölf Stunden durch cron(8) erzwingen:

```
0     */12       *       *       *       root    /usr/sbin/audit -n
```

Die Änderung wird wirksam, sobald Sie die neue `/etc/crontab` gespeichert haben.

Die automatische Rotation der Audit-Pfad-Datei in Abhängigkeit von der Dateigröße ist möglich durch die Angabe der Option `filesz` in audit_control(5). Dieser Vorgang ist im Abschnitt Konfigurationsdateien dieses Kapitels beschrieben.

18.5.6. Komprimierung von Audit-Pfaden

Da Audit-Pfad-Dateien sehr groß werden können, ist es oft wünschenswert, Pfade zu komprimieren oder anderweitig zu archivieren, sobald sie vom Audit-Daemon geschlossen wurden. Das Skript `audit_warn` kann genutzt werden, um angepasste Aktionen für eine Vielzahl von audit-bezogenen Ereignissen auszuführen, einschliesslich der sauberen Beendigung von Audit-Pfaden, wenn diese geschlossen werden. Zum Beispiel kann man die folgenden Zeilen in das `audit_warn`-Skript aufnehmen, um Audit-Pfade beim Beenden zu komprimieren:

```
#
# Compress audit trail files on close.
#
if [ "$1" = closefile ]; then
        gzip -9 $2
fi
```

Andere Archivierungsaktivitäten können das Kopieren zu einem zentralen Server, die Löschung der alten Pfad-Dateien oder die Reduzierung des alten Audit-Pfades durch Entfernung nicht benötigter Datensätze einschliessen. Das Skript wird nur dann ausgeführt, wenn die Audit-Pfad-Dateien sauber beendet wurden, daher wird es nicht auf Pfaden laufen, welche durch ein unsauberes Herunterfahren des Systems nicht beendet wurden.

Kapitel 19. Speichermedien

Übersetzt von Bernd Warken und Martin Heinen.

19.1. Übersicht

Dieses Kapitel behandelt die Benutzung von Laufwerken unter FreeBSD. Laufwerke können speichergestützte Laufwerke, Netzwerklaufwerke oder normale SCSI/IDE-Geräte sein.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Die Begriffe, die FreeBSD verwendet, um die Organisation der Daten auf einem physikalischen Laufwerk zu beschreiben (Partitionen und Slices).
- Wie Sie ein weiteres Laufwerk zu Ihrem System hinzufügen.
- Wie virtuelle Dateisysteme, zum Beispiel RAM-Disks, eingerichtet werden.
- Wie Sie mit Quotas die Benutzung von Laufwerken einschränken können.
- Wie Sie Partitionen verschlüsseln, um Ihre Daten zu schützen.
- Wie unter FreeBSD CDs und DVDs gebrannt werden.
- Sie werden die Speichermedien, die Sie für Backups einsetzen können, kennen.
- Wie Sie die unter FreeBSD erhältlichen Backup Programme benutzen.
- Wie Sie ein Backup mit Disketten erstellen.
- Was Dateisystem-Schnappschüsse sind und wie sie eingesetzt werden.

Bevor Sie dieses Kapitel lesen,

- sollten Sie einen einen FreeBSD-Kernel installieren können (Kapitel 9).

19.2. Gerätenamen

Die folgende Tabelle zeigt die von FreeBSD unterstützten Speichergeräte und deren Gerätenamen.

Tabelle 19-1. Namenskonventionen von physikalischen Laufwerken

Laufwerkstyp	Gerätename
IDE-Festplatten	`ad`
IDE-CD-ROM Laufwerke	`acd`
SCSI-Festplatten und USB-Speichermedien	`da`
SCSI-CD-ROM Laufwerke	`cd`
Verschiedene proprietäre CD-ROM-Laufwerke	`mcd` Mitsumi CD-ROM und `scd` Sony CD-ROM
Diskettenlaufwerke	`fd`
SCSI-Bandlaufwerke	`sa`

Laufwerkstyp	Gerätename
IDE-Bandlaufwerke	`ast`
Flash-Laufwerke	`fla` für DiskOnChip® Flash-Device
RAID-Laufwerke	`aacd` für Adaptec® AdvancedRAID, `mlxd` und `mlyd` für Mylex®, `amrd` für AMI MegaRAID®, `idad` für Compaq Smart RAID, `twed` für 3ware® RAID.

19.3. Hinzufügen von Laufwerken

Im Original von David O'Brian.

Der folgende Abschnitt beschreibt, wie Sie ein neues SCSI-Laufwerk zu einer Maschine hinzufügen, die momentan nur ein Laufwerk hat. Dazu schalten Sie zuerst den Rechner aus und installieren das Laufwerk entsprechend der Anleitungen Ihres Rechners, Ihres Controllers und des Laufwerkherstellers. Den genauen Ablauf können wir wegen der großen Abweichungen leider nicht beschreiben.

Nachdem Sie das Laufwerk installiert haben, melden Sie sich als Benutzer `root` an und kontrollieren Sie `/var/run/dmesg.boot`, um sicherzustellen, dass das neue Laufwerk gefunden wurde. Das neue Laufwerk wird, um das Beispiel fortzuführen, `da1` heißen und soll unter `/1` eingehängt werden. Fügen Sie eine IDE-Platte hinzu, wird diese den Namen `ad1` erhalten.

Da FreeBSD auf IBM-PC kompatiblen Rechnern läuft, muss es die PC BIOS-Partitionen, die verschieden von den traditionellen BSD-Partitionen sind, berücksichtigen. Eine PC Platte kann bis zu vier BIOS-Partitionen enthalten. Wenn die Platte ausschließlich für FreeBSD verwendet wird, können Sie den *dedicated* Modus benutzen, ansonsten muss FreeBSD in eine der BIOS-Partitionen installiert werden. In FreeBSD heißen die PC BIOS-Partitionen *Slices*, um sie nicht mit den traditionellen BSD-Partitionen zu verwechseln. Sie können auch Slices auf einer Platte verwenden, die ausschließlich von FreeBSD benutzt wird, sich aber in einem Rechner befindet, der noch ein anderes Betriebssystem installiert hat. Dadurch stellen Sie sicher, dass Sie `fdisk` des anderen Betriebssystems noch benutzen können.

Im Fall von Slices wird die Platte als `/dev/da1s1e` hinzugefügt. Das heißt: SCSI-Platte, Einheit 1 (die zweite SCSI-Platte), Slice 1 (PC BIOS-Partition 1) und die `e` BSD-Partition. Wird die Platte ausschließlich für FreeBSD verwendet ("dangerously dedicated"), wird sie einfach als `/dev/da1e` hinzugefügt.

> **Anmerkung:** Da bsdlabel(8) zum Speichern von Sektoren 32-Bit Integer verwendet, ist das Werkzeug in den meisten Fällen auf 2^32-1 Sektoren pro Laufwerk oder 2 TB beschränkt. In fdisk(8) darf der Startsektor nicht größer als 2^32-1 sein und Partitionen sind auf eine Länge von 2^32-1 beschränkt. In den meisten Fällen beschränkt dies die Größe einer Partition auf 2 TB und die maximale Größe eines Laufwerks auf 4 TB. Das sunlabel(8)-Format ist mit 2^32-1 Sektoren pro Partition und 8 Partitionen auf 16 TB beschränkt. Mit größeren Laufwerken können gpt(8)-Partitionen benutzt werden, um GPT-Partitionen zu erstellen. GPT hat den zusätzlichen Vorteil, dass es nicht auf 4 Slices beschränkt ist.

19.3.1. Verwenden von sysinstall(8)

1. Das **sysinstall** Menü

Um ein Laufwerk zu partitionieren und zu labeln, kann das menügestützte `sysinstall` benutzt werden. Dazu melden Sie sich als `root` an oder benutzen `su`, um `root` zu werden. Starten Sie `sysinstall` und wählen das `Configure` Menü, wählen Sie dort den Punkt `Fdisk` aus.

2. Partitionieren mit **fdisk**

   Innerhalb von **fdisk** geben Sie **A** ein, um die ganze Platte für FreeBSD zu benutzen. Beantworten Sie die Frage "remain cooperative with any future possible operating systems" mit `YES`. **W** schreibt die Änderung auf die Platte, danach können Sie **fdisk** mit **Q** verlassen. Da Sie eine Platte zu einem schon laufenden System hinzugefügt haben, beantworten Sie die Frage nach dem Master Boot Record mit `None`.

3. Disk-Label-Editor

   Als nächstes müssen Sie **sysinstall** verlassen und es erneut starten. Folgen Sie dazu bitte den Anweisungen von oben, aber wählen Sie dieses Mal die Option `Label`, um in den `Disk Label Editor` zu gelangen. Hier werden die traditionellen BSD-Partitionen erstellt. Ein Laufwerk kann acht Partitionen, die mit den Buchstaben `a-h` gekennzeichnet werden, besitzen. Einige Partitionen sind für spezielle Zwecke reserviert. Die `a` Partition ist für die Root-Partition (`/`) reserviert. Deshalb sollte nur das Laufwerk, von dem gebootet wird, eine `a` Partition besitzen. Die `b` Partition wird für Swap-Partitionen benutzt, wobei Sie diese auf mehreren Platten benutzen dürfen. Im "dangerously dedicated" Modus spricht die `c` Partition die gesamte Platte an, werden Slices verwendet, wird damit die ganze Slice angesprochen. Die anderen Partitionen sind für allgemeine Zwecke verwendbar.

   Der Label Editor von **sysinstall** bevorzugt die `e` Partition für Partitionen, die weder Root-Partitionen noch Swap-Partitionen sind. Im Label Editor können Sie ein einzelnes Dateisystem mit **C** erstellen. Wählen Sie `FS`, wenn Sie gefragt werden, ob Sie ein FS (Dateisystem) oder Swap erstellen wollen, und geben Sie einen Mountpoint z.B. `/mnt` an. Wenn Sie nach einer FreeBSD-Installation ein Dateisystem mit **sysinstall** erzeugen, so werden die Einträge in `/etc/fstab` nicht erzeugt, so dass die Angabe des Mountpoints nicht wichtig ist.

   Sie können nun das Label auf das Laufwerk schreiben und das Dateisystem erstellen, indem Sie **W** drücken. Ignorieren Sie die Meldung von **sysinstall**, dass die neue Partition nicht angehangen werden konnte, und verlassen Sie den Label Editor sowie **sysinstall**.

4. Ende

   Im letzten Schritt fügen Sie noch in `/etc/fstab` den Eintrag für das neue Laufwerk ein.

19.3.2. Die Kommandozeile

19.3.2.1. Anlegen von Slices

Mit der folgenden Vorgehensweise wird eine Platte mit anderen Betriebssystemen, die vielleicht auf Ihrem Rechner installiert sind, zusammenarbeiten und nicht das `fdisk` Programm anderer Betriebssysteme stören. Bitte benutzen Sie den `dedicated` Modus nur dann, wenn Sie dazu einen guten Grund haben!

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# fdisk -BI da1 # Initialisieren der neuen Platte
# bsdlabel -B -w da1s1 auto #Labeln.
# bsdlabel -e da1s1 # Editieren des Disklabels und Hinzufügen von Partitionen
# mkdir -p /1
# newfs /dev/da1s1e # Wiederholen Sie diesen Schritt für jede Partition
# mount /dev/da1s1e /1 # Anhängen der Partitionen
```

```
# vi /etc/fstab # Ändern Sie /etc/fstab entsprechend
```

Wenn Sie ein IDE-Laufwerk besitzen, ändern Sie `da` in `ad`.

19.3.2.2. Dedicated

Wenn das neue Laufwerk nicht von anderen Betriebssystemen benutzt werden soll, können Sie es im `dedicated` Modus betreiben. Beachten Sie bitte, dass Microsoft-Betriebssysteme mit diesem Modus eventuell nicht zurechtkommen, aber es entsteht kein Schaden am Laufwerk. Im Gegensatz dazu wird IBMs OS/2 versuchen, jede ihm nicht bekannte Partition zu reparieren.

```
# dd if=/dev/zero of=/dev/da1 bs=1k count=1
# bsdlabel -Bw da1 auto
# bsdlabel -e da1  # Erstellen der ‘e’ Partition
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab # /dev/da1e hinzufügen
# mount /1
```

Eine alternative Methode:

```
# dd if=/dev/zero of=/dev/da1 count=2
# bsdlabel /dev/da1 | bsdlabel -BR da1 /dev/stdin
# newfs /dev/da1e
# mkdir -p /1
# vi /etc/fstab # /dev/da1e hinzufügen
# mount /1
```

19.4. RAID

19.4.1. Software-RAID

19.4.1.1. Concatenated-Disk (CCD) konfigurieren

Original von Christopher Shumway. Überarbeitet von Jim Brown.

Die wichtigsten Faktoren bei der Auswahl von Massenspeichern sind Geschwindigkeit, Zuverlässigkeit und Preis. Selten findet sich eine ausgewogene Mischung aller drei Faktoren. Schnelle und zuverlässige Massenspeicher sind für gewöhnlich teuer. Um die Kosten zu senken, muss entweder an der Geschwindigkeit oder an der Zuverlässigkeit gespart werden.

Das unten beschriebene System sollte vor allem preiswert sein. Der nächst wichtige Faktor war die Geschwindigkeit gefolgt von der Zuverlässigkeit. Die Geschwindigkeit war nicht so wichtig, da über das Netzwerk auf das System zugegriffen wird. Da alle Daten schon auf CD-Rs gesichert sind, war die Zuverlässigkeit, obwohl wichtig, ebenfalls nicht von entscheidender Bedeutung.

Die Bewertung der einzelnen Faktoren ist der erste Schritt bei der Auswahl von Massenspeichern. Wenn Sie vor allem ein schnelles und zuverlässiges Medium benötigen und der Preis nicht wichtig ist, werden Sie ein anderes System als das hier beschriebene zusammenstellen.

19.4.1.1.1. Installation der Hardware

Neben der IDE-Systemplatte besteht das System aus drei Western Digital IDE-Festplatten mit 5400 RPM und einer Kapazität von je 30 GB. Insgesamt stehen also 90 GB Speicherplatz zur Verfügung. Im Idealfall sollte jede Festplatte an einen eigenen Controller angeschlossen werden. Um Kosten zu sparen, wurde bei diesem System darauf verzichtet und an jeden IDE-Controller eine Master- und eine Slave-Platte angeschlossen.

Beim Reboot wurde das BIOS so konfiguriert, dass es die angeschlossenen Platten automatisch erkennt und FreeBSD erkannte die Platten ebenfalls:

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

> **Anmerkung:** Wenn FreeBSD die Platten nicht erkennt, überprüfen Sie, ob die Jumper korrekt konfiguriert sind. Die meisten IDE-Festplatten verfügen über einen “Cable Select”-Jumper. Die Master- und Slave-Platten werden mit einem anderen Jumper konfiguriert. Bestimmen Sie den richtigen Jumper mithilfe der Dokumentation Ihrer Festplatte.

Als nächstes sollten Sie überlegen, auf welche Art der Speicher zur Verfügung gestellt werden soll. Schauen Sie sich dazu vinum(4) (Kapitel 22) und ccd(4) an. Im hier beschriebenen System wird ccd(4) eingesetzt.

19.4.1.1.2. Konfiguration von CCD

Mit ccd(4) können mehrere gleiche Platten zu einem logischen Dateisystem zusammengefasst werden. Um ccd(4) zu benutzen, muss der Kernel mit der entsprechenden Unterstützung übersetzt werden. Ergänzen Sie die Kernelkonfiguration um die nachstehende Zeile. Anschließend müssen Sie den Kernel neu übersetzen und installieren.

```
pseudo-device   ccd
```

Alternativ kann ccd(4) auch als Kernelmodul geladen werden.

Um ccd(4) zu benutzen, müssen die Laufwerke zuerst mit einem Label versehen werden. Die Label werden mit bsdlabel(8) erstellt:

```
bsdlabel -w ad1 auto
bsdlabel -w ad2 auto
bsdlabel -w ad3 auto
```

Damit wurden die Label `ad1c`, `ad2c` und `ad3c` erstellt, die jeweils das gesamte Laufwerk umfassen.

Im nächsten Schritt muss der Typ des Labels geändert werden. Die Labels können Sie mit bsdlabel(8) editieren:

```
bsdlabel -e ad1
bsdlabel -e ad2
```

```
bsdlabel -e ad3
```

Für jedes Label startet dies den durch `EDITOR` gegebenen Editor, typischerweise vi(1).

Ein unverändertes Label sieht zum Beispiel wie folgt aus:

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
```

Erstellen Sie eine `e`-Partition für ccd(4). Dazu können Sie normalerweise die Zeile der `c`-Partition kopieren, allerdings muss `fstype` auf **`4.2BSD`** gesetzt werden. Das Ergebnis sollte wie folgt aussehen:

```
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  c: 60074784        0    unused        0     0     0   # (Cyl.    0 - 59597)
  e: 60074784        0    4.2BSD        0     0     0   # (Cyl.    0 - 59597)
```

19.4.1.1.3. Erstellen des Dateisystems

Nachdem alle Platten ein Label haben, kann das ccd(4)-RAID aufgebaut werden. Dies geschieht mit ccdconfig(8):

```
ccdconfig ccd0❶ 32❷ 0❸ /dev/ad1e❹ /dev/ad2e /dev/ad3e
```

Die folgende Aufstellung erklärt die verwendeten Kommandozeilenargumente:

❶ Das erste Argument gibt das zu konfigurierende Gerät, hier `/dev/ccd0c`, an. Die Angabe von `/dev/` ist dabei optional.

❷ Der Interleave für das Dateisystem. Der Interleave definiert die Größe eines Streifens in Blöcken, die normal 512 Bytes groß sind. Ein Interleave von 32 ist demnach 16384 Bytes groß.

❸ Weitere Argumente für ccdconfig(8). Wenn Sie spiegeln wollen, können Sie das hier angeben. Die gezeigte Konfiguration verwendet keine Spiegel, sodass der Wert `0` angegeben ist.

❹ Das letzte Argument gibt die Geräte des Plattenverbundes an. Benutzen Sie für jedes Gerät den kompletten Pfadnamen.

Nach Abschluß von ccdconfig(8) ist der Plattenverbund konfiguriert und es können Dateisysteme auf dem Plattenverbund angelegt werden. Das Anlegen von Dateisystemen wird in der Hilfeseite newfs(8) beschrieben. Für das Beispiel genügt der folgende Befehl:

```
newfs /dev/ccd0c
```

19.4.1.1.4. Automatisierung

Damit ccd(4) beim Start automatisch aktiviert wird, ist die Datei `/etc/ccd.conf` mit dem folgenden Kommando zu erstellen:

```
ccdconfig -g > /etc/ccd.conf
```

Wenn `/etc/ccd.conf` existiert, wird beim Reboot `ccdconfig -C` von `/etc/rc` aufgerufen. Damit wird ccd(4) eingerichtet und die darauf befindlichen Dateisysteme können angehängt werden.

Anmerkung: Wenn Sie in den Single-User Modus booten, müssen Sie den Verbund erst konfigurieren, bevor Sie darauf befindliche Dateisysteme anhängen können:

```
ccdconfig -C
```

In `/etc/fstab` ist noch ein Eintrag für das auf dem Verbund befindliche Dateisystem zu erstellen, damit dieses beim Start des Systems immer angehängt wird:

```
/dev/ccd0c              /media       ufs     rw      2       2
```

19.4.1.2. Der Vinum-Volume-Manager

Der Vinum Volume Manager ist ein Block-Gerätetreiber, der virtuelle Platten zur Verfügung stellt. Er trennt die Verbindung zwischen der Festplatte und dem zugehörigen Block-Gerät auf. Im Gegensatz zur konventionellen Aufteilung einer Platte in Slices lassen sich dadurch Daten flexibler, leistungsfähiger und zuverlässiger verwalten. vinum(4) stellt RAID-0, RAID-1 und RAID-5 sowohl einzeln wie auch in Kombination zur Verfügung.

Mehr Informationen über vinum(4) erhalten Sie in Kapitel 22.

19.4.2. Hardware-RAID

FreeBSD unterstützt eine Reihe von RAID-Controllern. Diese Geräte verwalten einen Plattenverbund; zusätzliche Software wird nicht benötigt.

Der Controller steuert mithilfe eines BIOS auf der Karte die Plattenoperationen. Wie ein RAID System eingerichtet wird, sei kurz am Beispiel des Promise IDE RAID-Controllers gezeigt. Nachdem die Karte eingebaut ist und der Rechner neu gestartet wurde, erscheint eine Eingabeaufforderung. Wenn Sie den Anweisungen auf dem Bildschirm folgen, gelangen Sie in eine Maske, in der Sie mit den vorhandenen Festplatten ein RAID-System aufbauen können. FreeBSD behandelt das RAID-System wie eine einzelne Festplatte.

19.4.3. Wiederherstellen eines ATA-RAID-1 Verbunds

Mit FreeBSD können Sie eine ausgefallene Platte in einem RAID-Verbund während des Betriebs auswechseln, vorausgesetzt Sie bemerken den Ausfall vor einem Neustart.

Einen Ausfall erkennen Sie, wenn in der Datei `/var/log/messages` oder in der Ausgabe von dmesg(8) Meldungen wie die folgenden auftauchen:

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107 tn 4 sn 11)\\
status=59 error=40
ar0: WARNING - mirror lost
```

Überprüfen Sie den RAID-Verbund mit atacontrol(8):

```
# atacontrol list
ATA channel 0:
       Master:      no device present
       Slave:   acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
       Master:      no device present
       Slave:       no device present

ATA channel 2:
       Master:  ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
       Slave:       no device present

ATA channel 3:
       Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
       Slave:       no device present

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. Damit Sie die Platte ausbauen können, muss zuerst der ATA-Channel der ausgefallenen Platte aus dem Verbund entfernt werden:

   ```
   # atacontrol detach ata3
   ```

2. Ersetzen Sie dann die Platte.

3. Nun aktivieren Sie den ATA-Channel wieder:

   ```
   # atacontrol attach ata3
   Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
   Slave:   no device present
   ```

4. Nehmen Sie die neue Platte in den Verbund auf:

   ```
   # atacontrol addspare ar0 ad6
   ```

5. Stellen Sie die Organisation des Verbunds wieder her:

   ```
   # atacontrol rebuild ar0
   ```

6. Sie können den Fortschritt des Prozesses durch folgende Befehle kontrollieren:

   ```
   # dmesg | tail -10
   [output removed]
   ad6: removed from configuration
   ad6: deleted from ar0 disk1
   ad6: inserted into ar0 disk1 as spare

   # atacontrol status ar0
   ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
   ```

7. Warten Sie bis die Wiederherstellung beendet ist.

19.5. USB Speichermedien

Beigetragen von Marc Fonvieille.

Der Universal Serial Bus (USB) wird heutzutage von vielen externen Speichern benutzt: Festplatten, USB-Thumbdrives oder CD-Brennern, die alle von FreeBSD unterstützt werden.

19.5.1. USB-Konfiguration

USB-Massenspeicher werden vom Treiber umass(4) betrieben. Wenn Sie den `GENERIC`-Kernel benutzen, brauchen Sie keine Anpassungen vorzunehmen. Benutzen Sie einen angepassten Kernel, müssen die nachstehenden Zeilen in der Kernelkonfigurationsdatei enthalten sein:

```
device scbus
device da
device pass
device uhci
device ohci
device ehci
device usb
device umass
```

Der Treiber umass(4) greift über das SCSI-Subsystem auf die USB-Geräte zu. Ihre USB-Geräte werden daher vom System als SCSI-Geräte erkannt. Abhängig vom Chipsatz Ihrer Systemplatine benötigen Sie in der Kernelkonfiguration entweder die Option `device uhci` oder die Option `device ohci` für die Unterstützung von USB 1.1. Die Kernelkonfiguration kann allerdings auch beide Optionen enthalten. Unterstützung für USB 2.0 Controller wird durch den ehci(4)-Treiber geleistet (die `device ehci` Zeile). Vergessen Sie bitte nicht, einen neuen Kernel zu bauen und zu installieren, wenn Sie die Kernelkonfiguration verändert haben.

> **Anmerkung:** Wenn es sich bei Ihrem USB-Gerät um einen CD-R- oder DVD-Brenner handelt, müssen Sie den Treiber cd(4) für SCSI-CD-ROMs in die Kernelkonfiguration aufnehmen:
>
> ```
> device cd
> ```
>
> Da der Brenner als SCSI-Laufwerk erkannt wird, sollten Sie den Treiber atapicam(4) nicht benutzen.

19.5.2. Die USB-Konfiguration testen

Sie können das USB-Gerät nun testen. Schließen Sie das Gerät an und untersuchen Sie die Systemmeldungen (dmesg(8)), Sie sehen Ausgaben wie die folgende:

```
umass0: USB Solid state disk, rev 1.10/1.00, addr 2
GEOM: create disk da0 dp=0xc2d74850
da0 at umass-sim0 bus 0 target 0 lun 0
da0: <Generic Traveling Disk 1.11> Removable Direct Access SCSI-2 device
da0: 1.000MB/s transfers
da0: 126MB (258048 512 byte sectors: 64H 32S/T 126C)
```

Die Ausgaben, wie das erkannte Gerät oder der Gerätename (`da0`) hängen natürlich von Ihrer Konfiguration ab.

Da ein USB-Gerät als SCSI-Gerät erkannt wird, können Sie USB-Massenspeicher mit dem Befehl `camcontrol` anzeigen:

```
# camcontrol devlist
<Generic Traveling Disk 1.11>      at scbus0 target 0 lun 0 (da0,pass0)
```

Wenn auf dem Laufwerk ein Dateisystem eingerichtet ist, sollten Sie das Dateisystem einhängen können. Abschnitt 19.3 beschreibt, wie Sie USB-Laufwerke formatieren und Partitionen einrichten.

> **Warnung:** Aus Sicherheitsgründen sollten Sie Benutzern, denen Sie nicht vertrauen, das Einhängen (z.B. durch die unten beschriebene Aktivierung von `vfs.usermount`) beliebiger Medien verbieten. Die meisten Dateisysteme in FreeBSD wurden nicht entwickelt, um sich vor böswilligen Geräten zu schützen.

Damit auch normale Anwender (ohne `root`-Rechte) USB-Laufwerke einhängen können, müssen Sie Ihr System erst entsprechend konfigurieren. Als erstes müssen Sie sicherstellen, dass diese Anwender auf die beim Einhängen eines USB-Laufwerks dynamisch erzeugten Gerätedateien zugreifen dürfen. Dazu können Sie beispielsweise mit pw(8) alle potentiellen Benutzer dieser Gerätedateien in die Gruppe `operator` aufnehmen. Außerdem muss sichergestellt werden, dass Mitglieder der Gruppe `operator` Schreib- und Lesezugriff auf diese Gerätedateien haben. Dazu fügen Sie die folgenden Zeilen in die Konfigurationsdatei `/etc/devfs.rules` ein:

```
[localrules=5]
add path 'da*' mode 0660 group operator
```

> **Anmerkung:** Verfügt Ihr System auch über SCSI-Laufwerke, gibt es eine Besonderheit. Haben Sie beispielsweise die SCSI-Laufwerke `da0` bis `da2` installiert, so sieht die zweite Zeile wie folgt aus:
>
> ```
> add path 'da[3-9]*' mode 0660 group operator
> ```
>
> Dadurch werden die bereits vorhandenen SCSI-Laufwerke nicht in die Gruppe `operator` aufgenommen.

Vergessen Sie nicht, die devfs.rules(5)-Regeln in der Datei `/etc/rc.conf` zu aktivieren:

```
devfs_system_ruleset="localrules"
```

Als nächstes müssen Sie Ihre Kernelkonfiguration anpassen, damit auch normale Benutzer Dateisysteme mounten dürfen. Dazu fügen Sie am besten folgende Zeile in die Konfigurationsdatei `/etc/sysctl.conf` ein:

```
vfs.usermount=1
```

Damit diese Einstellung wirksam wird, müssen Sie Ihr System neu starten. Alternativ können Sie diese Variable auch mit sysctl(8) setzen.

Zuletzt müssen Sie noch ein Verzeichnis anlegen, in das das USB-Laufwerk eingehängt werden soll. Dieses Verzeichnis muss dem Benutzer gehören, der das USB-Laufwerk in den Verzeichnisbaum einhängen will. Dazu legen Sie als `root` ein Unterverzeichnis `/mnt/`*`username`* an (wobei Sie *`username`* durch den Login des jeweiligen Benutzers sowie *`usergroup`* durch die primäre Gruppe des Benutzers ersetzen):

```
# mkdir /mnt/username
 # chown username:usergroup /mnt/username
```

Wenn Sie nun beispielsweise einen USB-Stick anschließen, wird automatisch die Gerätedatei `/dev/da0s1` erzeugt. Da derartige Geräte in der Regel mit dem FAT-Dateisystem formatiert sind, können Sie sie beispielsweise mit dem folgenden Befehl in den Verzeichnisbaum einhängen:

```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/username
```

Wenn Sie das Gerät entfernen (das Dateisystem müssen Sie vorher abhängen), sehen Sie in den Systemmeldungen Einträge wie die folgenden:

```
umass0: at uhub0 port 1 (addr 2) disconnected
(da0:umass-sim0:0:0:0): lost device
(da0:umass-sim0:0:0:0): removing device entry
GEOM: destroy disk da0 dp=0xc2d74850
umass0: detached
```

19.5.3. Weiteres zu USB

Neben den Abschnitten Hinzufügen von Laufwerken und Anhängen und Abhängen von Dateisystemen lesen Sie bitte die Hilfeseiten umass(4), camcontrol(8) für FreeBSD 8.X oder usbdevs(8) bei vorherigen Versionen.

19.6. CDs benutzen

Beigesteuert von Mike Meyer.

19.6.1. Einführung

CDs besitzen einige Eigenschaften, die sie von konventionellen Laufwerken unterscheiden. Zuerst konnten sie nicht beschrieben werden. Sie wurden so entworfen, dass sie ununterbrochen, ohne Verzögerungen durch Kopfbewegungen zwischen den Spuren, gelesen werden können. Sie konnten früher auch leichter als vergleichbar große Medien zwischen Systemen bewegt werden.

CDs besitzen Spuren, aber damit ist der Teil Daten gemeint, der ununterbrochen gelesen wird, und nicht eine physikalische Eigenschaft der CD. Um eine CD mit FreeBSD zu erstellen, werden die Daten jeder Spur der CD in Dateien vorbereitet und dann die Spuren auf die CD geschrieben.

Das ISO 9660-Dateisystem wurde entworfen, um mit diesen Unterschieden umzugehen. Leider hat es auch damals übliche Grenzen für Dateisysteme implementiert. Glücklicherweise existiert ein Erweiterungsmechanismus, der es korrekt geschriebenen CDs erlaubt, diese Grenzen zu überschreiten und dennoch auf Systemen zu funktionieren, die diese Erweiterungen nicht unterstützen.

Der Port `sysutils/cdrtools` enthält das Programm mkisofs(8), das eine Datei erstellt, die ein ISO 9660-Dateisystem enthält. Das Programm hat Optionen, um verschiedene Erweiterungen zu unterstützen, und wird unten beschrieben.

Welches Tool Sie zum Brennen von CDs benutzen, hängt davon ab, ob Ihr CD-Brenner ein ATAPI-Gerät ist oder nicht. Mit ATAPI-CD-Brennern wird `burncd` benutzt, das Teil des Basissystems ist. SCSI- und USB-CD-Brenner werden mit `cdrecord` aus `sysutils/cdrtools` benutzt. Zusätzlich ist es möglich, über das Modul ATAPI/CAM SCSI-Werkzeuge wie `cdrecord` auch für ATAPI-Geräte einzusetzen.

Wenn Sie eine Brennsoftware mit grafischer Benutzeroberfläche benötigen, sollten Sie sich **X-CD-Roast** oder **K3b** näher ansehen. Diese Werkzeuge können als Paket oder aus den Ports (`sysutils/xcdroast` und `sysutils/k3b`) installiert werden. Mit ATAPI-Hardware benötigt **K3b** das ATAPI/CAM-Modul.

19.6.2. mkisofs

Das Programm mkisofs(8) aus dem Port `sysutils/cdrtools` erstellt ein ISO 9660-Dateisystem, das ein Abbild eines Verzeichnisbaumes ist. Die einfachste Anwendung ist wie folgt:

```
# mkisofs -o Imagedatei /path/to/tree
```

Dieses Kommando erstellt eine *`Imagedatei`*, die ein ISO 9660-Dateisystem enthält, das eine Kopie des Baumes unter */path/to/tree* ist. Dabei werden die Dateinamen auf Namen abgebildet, die den Restriktionen des ISO 9660-Dateisystems entsprechen. Dateien mit Namen, die im ISO 9660-Dateisystem nicht gültig sind, bleiben unberücksichtigt.

Es einige Optionen, um diese Beschränkungen zu überwinden. Die unter UNIX Systemen üblichen Rock-Ridge-Erweiterungen werden durch `-R` aktiviert, `-J` aktiviert die von Microsoft Systemen benutzten Joliet-Erweiterungen und `-hfs` dient dazu, um das von Mac OS benutzte HFS zu erstellen.

Für CDs, die nur auf FreeBSD-Systemen verwendet werden sollen, kann `-U` genutzt werden, um alle Beschränkungen für Dateinamen aufzuheben. Zusammen mit `-R` wird ein Abbild des Dateisystems, ausgehend von dem Startpunkt im FreeBSD-Dateibaum, erstellt, obwohl dies den ISO 9660 Standard verletzen kann.

Die letzte übliche Option ist `-b`. Sie wird benutzt, um den Ort eines Bootimages einer “El Torito” bootbaren CD anzugeben. Das Argument zu dieser Option ist der Pfad zu einem Bootimage ausgehend von der Wurzel des Baumes, der auf die CD geschrieben werden soll. In der Voreinstellung erzeugt mkisofs(8) ein ISO-Image im “Diskettenemulations”-Modus. Dabei muss das Image genau 1200, 1440 oder 2880 KB groß sein. Einige Bootloader, darunter der auf den FreeBSD-Disks verwendete, kennen keinen Emulationsmodus. Daher sollten Sie in diesen Fällen die Option `-no-emul-boot` verwenden. Wenn `/tmp/myboot` ein bootbares FreeBSD-System enthält, dessen Bootimage sich in `/tmp/myboot/boot/cdboot` befindet, können Sie ein Abbild eines ISO 9660-Dateisystems in `/tmp/bootable.iso` wie folgt erstellen:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Wenn Sie `md` in Ihrem Kernel konfiguriert haben, können Sie danach das Dateisystem einhängen:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Jetzt können Sie überprüfen, dass `/mnt` und `/tmp/myboot` identisch sind.

Sie können das Verhalten von mkisofs(8) mit einer Vielzahl von Optionen beeinflussen. Insbesondere können Sie das ISO 9660-Dateisystem modifizieren und Joliet- oder HFS-Dateisysteme brennen. Details dazu entnehmen Sie bitte der Hilfeseite mkisofs(8).

19.6.3. burncd

Wenn Sie einen ATAPI-CD-Brenner besitzen, können Sie `burncd` benutzen, um ein ISO-Image auf CD zu brennen. `burncd` ist Teil des Basissystems und unter `/usr/sbin/burncd` installiert. Da es nicht viele Optionen hat, ist es leicht zu benutzen:

```
# burncd -f cddevice data imagefile.iso fixate
```

Dieses Kommando brennt eine Kopie von *imagefile.iso* auf das Gerät *cddevice*. In der Grundeinstellung wird das Gerät `/dev/acd0` benutzt. burncd(8) beschreibt, wie die Schreibgeschwindigkeit gesetzt wird, die CD ausgeworfen wird und Audiodaten geschrieben werden.

19.6.4. cdrecord

Wenn Sie keinen ATAPI-CD-Brenner besitzen, benutzen Sie `cdrecord`, um CDs zu brennen. `cdrecord` ist nicht Bestandteil des Basissystems. Sie müssen es entweder aus den Ports in `sysutils/cdrtools` oder dem passenden Paket installieren. Änderungen im Basissystem können Fehler im binären Programm verursachen und führen möglicherweise dazu, dass Sie einen “Untersetzer” brennen. Sie sollten daher den Port aktualisieren, wenn Sie Ihr System aktualisieren bzw. wenn Sie STABLE verfolgen, den Port aktualisieren, wenn es eine neue Version gibt.

Obwohl `cdrecord` viele Optionen besitzt, ist die grundlegende Anwendung einfacher als `burncd`. Ein ISO 9660-Image erstellen Sie mit:

```
# cdrecord dev=device imagefile.iso
```

Der Knackpunkt in der Benutzung von `cdrecord` besteht darin, das richtige Argument zu `dev` zu finden. Benutzen Sie dazu den Schalter `-scanbus` von `cdrecord`, der eine ähnliche Ausgabe wie die folgende produziert:

```
# cdrecord -scanbus
Cdrecord 1.9 (i386-unknown-freebsd7.0) Copyright (C) 1995-2004 Jörg Schilling
Using libscg version 'schily-0.1'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) 'YAMAHA  ' 'CRW4260         ' '1.0q' Removable CD-ROM
        1,6,0   106) 'ARTEC   ' 'AM12S           ' '1.06' Scanner
        1,7,0   107) *
```

Für die aufgeführten Geräte in der Liste wird das passende Argument zu `dev` gegeben. Benutzen Sie die drei durch Kommas separierten Zahlen, die zu Ihrem CD-Brenner angegeben sind, als Argument für `dev`. Im Beispiel ist das CDRW-Gerät 1,5,0, so dass die passende Eingabe **`dev=1,5,0`** wäre. Einfachere Wege das Argument anzugeben, sind in cdrecord(1) beschrieben. Dort sollten Sie auch nach Informationen über Audiospuren, das Einstellen der Geschwindigkeit und ähnlichem suchen.

19.6.5. Kopieren von Audio-CDs

Um eine Kopie einer Audio-CD zu erstellen, kopieren Sie die Stücke der CD in einzelne Dateien und brennen diese Dateien dann auf eine leere CD. Das genaue Verfahren hängt davon ab, ob Sie ATAPI- oder SCSI-Laufwerke verwenden.

SCSI-Laufwerke

1. Kopieren Sie die Audiodaten mit `cdda2wav`:

   ```
   % cdda2wav -vall -D2,0 -B -Owav
   ```

2. Die erzeugten `.wav` Dateien schreiben Sie mit `cdrecord` auf eine leere CD:

   ```
   % cdrecord -v dev=2,0 -dao -useinfo  *.wav
   ```

   Das Argument von `dev` gibt das verwendete Gerät an, das Sie, wie in Abschnitt 19.6.4 beschrieben, ermitteln können.

ATAPI-Laufwerke

> **Anmerkung:** Über das Modul ATAPI/CAM kann `cdda2wav` auch mit ATAPI-Laufwerken verwendet werden. Diese Methode ist für die meisten Anwender besser geeignet als die im folgenden beschriebenen Methoden (Jitter-Korrektur, Big-/Little-Endian-Probleme und anderes mehr spielen hierbei eine Rolle).

1. Der ATAPI-CD-Treiber stellt die einzelnen Stücke der CD über die Dateien `/dev/acd`*d*`t`*nn*, zur Verfügung. *d* bezeichnet die Laufwerksnummer und *nn* ist die Nummer des Stücks. Die Nummer ist immer zweistellig, das heißt es wird, wenn nötig, eine führende Null ausgegeben. Die Datei `/dev/acd0t01` ist also das erste Stück des ersten CD-Laufwerks. `/dev/acd0t02` ist das zweite Stück und `/dev/acd0t03` das dritte.

   Überprüfen Sie stets, ob die entsprechenden Dateien im Verzeichnis `/dev` auch angelegt werden. Sind die Einträge nicht vorhanden, weisen Sie Ihr System an, das Medium erneut zu testen:

   ```
   # dd if=/dev/acd0 of=/dev/null count=1
   ```

   > **Anmerkung:** Unter FreeBSD 4.X werden diese Einträge nicht mit dem Wert Null vordefiniert. Falls die entsprechenden Einträge unter `/dev` nicht vorhanden sind, müssen Sie diese hier von `MAKEDEV` anlegen lassen:
   >
   > ```
   > # cd /dev
   > # sh MAKEDEV acd0t99
   > ```

2. Die einzelnen Stücke kopieren Sie mit dd(1). Sie müssen dazu eine spezielle Blockgröße angeben:

   ```
   # dd if=/dev/acd0t01 of=track1.cdr bs=2352
   # dd if=/dev/acd0t02 of=track2.cdr bs=2352
   ...
   ```

3. Die kopierten Dateien können Sie dann mit `burncd` brennen. Auf der Kommandozeile müssen Sie angeben, dass Sie Audio-Daten brennen wollen und dass das Medium fixiert werden soll:

   ```
   # burncd -f /dev/acd0 audio track1.cdr track2.cdr ... fixate
   ```

19.6.6. Kopieren von Daten-CDs

Sie können eine Daten-CD in eine Datei kopieren, die einem Image entspricht, das mit mkisofs(8) erstellt wurde. Mit Hilfe dieses Images können Sie jede Daten-CD kopieren. Das folgende Beispiel verwendet `acd0` für das CD-ROM-Gerät. Wenn Sie ein anderes Laufwerk benutzen, setzen Sie bitte den richtigen Namen ein.

```
# dd if=/dev/acd0 of=file.iso bs=2048
```

Danach haben Sie ein Image, das Sie wie oben beschrieben, auf eine CD brennen können.

19.6.7. Einhängen von Daten-CDs

Nachdem Sie eine Daten-CD gebrannt haben, wollen Sie wahrscheinlich auch die Daten auf der CD lesen. Dazu müssen Sie die CD in den Dateibaum einhängen. Die Voreinstellung für den Typ des Dateisystems von mount(8) ist `UFS`. Das System wird die Fehlermeldung `Incorrect super block` ausgeben, wenn Sie versuchen, die CD mit dem folgenden Kommando einzuhängen:

```
# mount /dev/cd0 /mnt
```

Auf der CD befindet sich ja kein `UFS` Dateisystem, so dass der Versuch, die CD einzuhängen fehlschlägt. Sie müssen mount(8) sagen, dass es ein Dateisystem vom Typ `ISO9660` verwenden soll. Dies erreichen Sie durch die Angabe von `-t cd9660` auf der Kommandozeile. Wenn Sie also die CD-ROM `/dev/cd0` in `/mnt` einhängen wollen, führen Sie folgenden Befehl aus:

```
# mount -t cd9660 /dev/cd0c /mnt
```

Abhängig vom verwendeten CD-ROM kann der Gerätename von dem im Beispiel (`/dev/cd0`) abweichen. Die Angabe von `-t cd9660` führt mount_cd9660(8) aus, so dass das Beispiel verkürzt werden kann:

```
# mount_cd9660 /dev/cd0 /mnt
```

Auf diese Weise können Sie Daten-CDs von jedem Hersteller verwenden. Es kann allerdings zu Problemen mit CDs kommen, die verschiedene ISO9660-Erweiterungen benutzen. So speichern Joliet-CDs alle Dateinamen unter Verwendung von zwei Byte langen Unicode-Zeichen. Zwar unterstützt der FreeBSD-Kernel derzeit noch kein Unicode, der CD9660-Treiber erlaubt es aber, zur Laufzeit eine Konvertierungstabelle zu laden. Tauchen bei Ihnen also statt bestimmter Zeichen nur Fragezeichen auf, so müssen Sie über die Option `-C` den benötigten Zeichensatz angeben. Weitere Informationen zu diesem Problem finden Sie in der Manualpage mount_cd9660(8).

> **Anmerkung:** Damit der Kernel diese Zeichenkonvertierung (festgelegt durch die Option `-C`) erkennt, müssen Sie das Kernelmodul `cd9660_iconv.ko` laden. Dazu fügen Sie folgende Zeile in die Datei `loader.conf` ein:
>
> ```
> cd9660_iconv_load="YES"
> ```
>
> Danach müssen Sie allerdings Ihr System neu starten. Alternativ können Sie das Kernelmodul auch direkt über kldload(8) laden.

Manchmal werden Sie die Meldung `Device not configured` erhalten, wenn Sie versuchen, eine CD-ROM einzuhängen. Für gewöhnlich liegt das daran, dass das Laufwerk meint es sei keine CD eingelegt, oder dass das Laufwerk auf dem Bus nicht erkannt wird. Es kann einige Sekunden dauern, bevor das Laufwerk merkt, dass eine CD eingelegt wurde. Seien Sie also geduldig.

Manchmal wird ein SCSI-CD-ROM nicht erkannt, weil es keine Zeit hatte, auf das Zurücksetzen des Busses zu antworten. Wenn Sie ein SCSI-CD-ROM besitzen, sollten Sie die folgende Zeile in Ihre Kernelkonfiguration aufnehmen und einen neuen Kernel bauen:

```
options SCSI_DELAY=15000
```

Die Zeile bewirkt, dass nach dem Zurücksetzen des SCSI-Busses beim Booten 15 Sekunden gewartet wird, um dem CD-ROM-Laufwerk genügend Zeit zu geben, darauf zu antworten.

19.6.8. Brennen von rohen CDs

Sie können eine Datei auch direkt auf eine CD brennen, ohne vorher auf ihr ein ISO 9660-Dateisystem einzurichten. Einige Leute nutzen dies, um Datensicherungen durchzuführen. Diese Vorgehensweise hat den Vorteil, dass Sie schneller als das Brennen einer normalen CD ist.

```
# burncd -f /dev/acd1 -s 12 data archive.tar.gz fixate
```

Wenn Sie die Daten von einer solchen CD wieder zurückbekommen wollen, müssen Sie sie direkt von dem rohen Gerät lesen:

```
# tar xzvf /dev/acd1
```

Eine auf diese Weise gefertigte CD können Sie nicht in das Dateisystem einhängen. Sie können Sie auch nicht auf einem anderen Betriebssystem lesen. Wenn Sie die erstellten CDs in das Dateisystem einhängen oder mit anderen Betriebssystemen austauschen wollen, müssen Sie mkisofs(8) wie oben beschrieben benutzen.

19.6.9. Der ATAPI/CAM Treiber

Beigetragen von Marc Fonvieille.

Mit diesem Treiber kann auf ATAPI-Geräte (wie CD-ROM-, CD-RW- oder DVD-Laufwerke) mithilfe des SCSI-Subsystems zugegriffen werden. Damit können Sie SCSI-Werkzeuge, wie `sysutils/cdrdao` oder cdrecord(1), zusammen mit einem ATAPI-Gerät benutzen.

Wenn Sie den Treiber benutzen wollen, fügen Sie die folgende Zeile in `/boot/loader.conf` ein:

```
atapicam_load="YES"
```

Danach müssen Sie Ihr System neu starten, um den Treiber zu aktivieren.

> **Anmerkung:** Alternativ können Sie die Unterstützung für atapicam(4) auch in Ihren Kernel kompilieren. Dazu fügen Sie die folgende Zeile in Ihre Kernelkonfigurationsdatei ein:
>
> ```
> device atapicam
> ```
>
> Die folgenden Zeilen werden ebenfalls benötigt, sollten aber schon Teil der Kernelkonfiguration sein:
>
> ```
> device ata
> device scbus
> device cd
> device pass
> ```

Übersetzen und installieren Sie den neuen Kernel. Der CD-Brenner sollte nun beim Neustart des Systems erkannt werden:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at ata1-master PIO4
cd0 at ata1 bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

Über den Gerätenamen `/dev/cd0` können Sie nun auf das Laufwerk zugreifen. Wenn Sie beispielsweise eine CD-ROM in `/mnt` einhängen wollen, benutzen Sie das nachstehende Kommando:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Die SCSI-Adresse des Brenners können Sie als `root` wie folgt ermitteln:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 (pass0,cd0)
```

Die SCSI-Adresse `1,0,0` können Sie mit den SCSI-Werkzeugen, zum Beispiel cdrecord(1), verwenden.

Weitere Informationen über das ATAPI/CAM- und das SCSI-System erhalten Sie in den Hilfeseiten atapicam(4) und cam(4).

19.7. DVDs benutzen

Beigetragen von Marc Fonvieille. Mit Beiträgen von Andy Polyakov.

19.7.1. Einführung

Nach der CD ist die DVD die nächste Generation optischer Speichermedien. Auf einer DVD können mehr Daten als auf einer CD gespeichert werden. DVDs werden heutzutage als Standardmedium für Videos verwendet.

Für beschreibbare DVDs existieren fünf Medienformate:

- DVD-R: Dies war das erste verfügbare Format. Das Format wurde vom DVD-Forum (http://www.dvdforum.com/forum.shtml) festgelegt. Die Medien sind nur einmal beschreibbar.
- DVD-RW: Dies ist die wiederbeschreibbare Version des DVD-R Standards. Eine DVD-RW kann ungefähr 1000 Mal beschrieben werden.
- DVD-RAM: Dies ist ebenfalls ein wiederbeschreibbares Format, das vom DVD-Forum unterstützt wird. Eine DVD-RAM verhält sich wie eine Wechselplatte. Allerdings sind die Medien nicht kompatibel zu den meisten DVD-ROM-Laufwerken und DVD-Video-Spielern. DVD-RAM wird nur von wenigen Brennern unterstützt. Wollen Sie DVD-RAM einsetzen, sollten Sie Abschnitt 19.7.9 lesen.
- DVD+RW: Ist ein wiederbeschreibbares Format, das von der DVD+RW Alliance (http://www.dvdrw.com/) festgelegt wurde. Eine DVD+RW kann ungefähr 1000 Mal beschrieben werden.
- DVD+R: Dieses Format ist die nur einmal beschreibbare Variante des DVD+RW Formats.

Auf einer einfach beschichteten DVD können 4.700.000.000 Bytes gespeichert werden. Das sind 4,38 GB oder 4485 MB (1 Kilobyte sind 1024 Bytes).

> **Anmerkung:** Die physischen Medien sind unabhängig von der Anwendung. Ein DVD-Video ist eine spezielle Anordnung von Dateien, die auf irgendein Medium (zum Beispiel DVD-R, DVD+R oder DVD-RW) geschrieben werden kann. Bevor Sie ein Medium auswählen, müssen Sie sicherstellen, dass der Brenner und der DVD-Spieler (ein Einzelgerät oder ein DVD-ROM-Laufwerk eines Rechners) mit dem Medium umgehen können.

19.7.2. Konfiguration

Das Programm growisofs(1) beschreibt DVDs. Das Kommando ist Teil der Anwendung **dvd+rw-tools** (`sysutils/dvd+rw-tools`). **dvd+rw-tools** kann mit allen DVD-Medien umgehen.

Um die Geräte anzusprechen, brauchen die Werkzeuge das SCSI-Subsystem. Daher muss der Kernel den ATAPI/CAM-Treiber zur Verfügung stellen. Der Treiber ist mit USB-Brennern nutzlos; die Konfiguration von USB-Geräten behandelt Abschnitt 19.5.

Für ATAPI-Geräte müssen Sie ebenfalls DMA-Zugriffe aktivieren. Fügen Sie dazu die nachstehende Zeile in die Datei `/boot/loader.conf` ein:

```
hw.ata.atapi_dma="1"
```

Bevor Sie **dvd+rw-tools** mit Ihrem DVD-Brenner benutzen, lesen Sie bitte die Hardware-Informationen auf der Seite dvd+rw-tools' hardware compatibility notes (http://fy.chalmers.se/~appro/linux/DVD+RW/hcn.html).

> **Anmerkung:** Wenn Sie eine grafische Oberfläche bevorzugen, schauen Sie sich bitte den Port `sysutils/k3b` an. Der Port bietet eine leicht zu bedienende Schnittstelle zu growisofs(1) und vielen anderen Werkzeugen.

19.7.3. Daten-DVDs brennen

growisofs(1) erstellt mit dem Programm mkisofs das Dateisystem und brennt anschließend die DVD. Vor dem Brennen brauchen Sie daher kein Abbild der Daten zu erstellen.

Wenn Sie von den Daten im Verzeichnis `/path/to/data` eine DVD+R oder eine DVD-R brennen wollen, benutzen Sie das nachstehende Kommando:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

Die Optionen `-J -R` werden an mkisofs(8) durchgereicht und dienen zum Erstellen des Dateisystems (hier: ein ISO-9660-Dateisystem mit Joliet- und Rock-Ridge-Erweiterungen). Weiteres entnehmen Sie bitte der Hilfeseite mkisofs(8).

Die Option `-Z` wird für die erste Aufnahme einer Session benötigt, egal ob Sie eine Multi-Session-DVD brennen oder nicht. Für */dev/cd0* müssen Sie den Gerätenamen Ihres Brenners einsetzen. Die Option `-dvd-compat` schließt das Medium, weitere Daten können danach nicht mehr angehängt werden. Durch die Angabe dieser Option kann das Medium von mehr DVD-ROM-Laufwerken gelesen werden.

Sie können auch ein vorher erstelltes Abbild der Daten brennen. Die nachstehende Kommandozeile brennt das Abbild in der Datei *imagefile.iso*:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

Die Schreibgeschwindigkeit hängt von den verwendeten Medium sowie dem verwendeten Gerät ab und sollte automatisch gesetzt werden. Falls Sie die Schreibgeschwindigkeit vorgeben möchten, verwenden Sie den Parameter `-speed=`. Weiteres erfahren Sie in der Hilfeseite growisofs(1).

> **Anmerkung:** Um grössere Dateien als 4.38GB in ihre Sammlung aufzunehmen, ist es notwendig ein UDF/ISO-9660 Hybrid-Dateisystem zu erstellen. Dieses Dateisystem muss mit zusätzlichen Parametern `-udf -iso-level 3` bei mkisofs(8) und allen relevanten Programmen (z.B. growisofs(1)) erzeugt werden. Dies ist nur notwendig wenn Sie ein ISO-Image erstellen oder direkt auf eine DVD schreiben wollen. DVDs, die in dieser Weise hergestellt worden sind, müssen als UDF-Dateisystem mit mount_udf(8) eingehangen werden. Sie sind nur auf Betriebssystemen, die UDF unterstützen brauchbar, ansonsten sieht es so aus, als ob sie kaputte Dateien enthalten würden.
>
> Um so eine ISO Datei zu bauen, geben Sie den folgenden Befehl ein:
>
> ```
> % mkisofs -R -J -udf -iso-level 3 -o imagefile.iso /path/to/data
> ```
>
> Um Daten direkt auf eine DVD zu brennen, geben Sie den folgenden Befehl ein:
>
> ```
> # growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R /path/to/data
> ```
>
> Wenn Sie ein ISO-Image haben das bereits grosse Dateien enthält, sind keine weiteren zusätzlichen Optionen für growisofs(1) notwendig, um das Image auf die DVD zu brennen.
>
> Beachten Sie noch, dass Sie die aktuelle Version von `sysutils/cdrtools` haben (welche mkisofs(8) enthält), da die älteren Versionen nicht den Support für grosse Dateien enthalten. Wenn Sie Probleme haben sollten, können Sie auch das Entwicklerpaket von `sysutils/cdrtools-devel` einsetzen und lesen Sie die mkisofs(8) Manualpage.

19.7.4. DVD-Videos brennen

Ein DVD-Video ist eine spezielle Anordnung von Dateien, die auf den ISO-9660 und den micro-UDF (M-UDF) Spezifikationen beruht. Ein DVD-Video ist auf eine bestimmte Datei-Hierarchie angewiesen. Daher müssen Sie DVDs mit speziellen Programmen wie `multimedia/dvdauthor` erstellen.

Wenn Sie schon ein Abbild des Dateisystems eines DVD-Videos haben, brennen Sie das Abbild wie jedes andere auch. Eine passende Kommandozeile finden Sie im vorigen Abschnitt. Wenn Sie die DVD im Verzeichnis `/path/to/video` zusammengestellt haben, erstellen Sie das DVD-Video mit dem nachstehenden Kommando:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

Die Option `-dvd-video` wird an mkisofs(8) weitergereicht. Dadurch erstellt mkisofs(8) die Datei-Hierarchie für ein DVD-Video. Weiterhin bewirkt die Angabe von `-dvd-video`, dass growisofs(1) mit der Option `-dvd-compat` aufgerufen wird.

19.7.5. DVD+RW-Medien benutzen

Im Gegensatz zu CD-RW-Medien müssen Sie DVD+RW-Medien erst formatieren, bevor Sie die Medien benutzen. Sie sollten growisofs(1) einzetzen, da das Programm Medien automatisch formatiert, wenn es erforderlich ist. Sie können eine DVD+RW aber auch mit dem Kommando `dvd+rw-format` formatieren:

```
# dvd+rw-format /dev/cd0
```

Sie müssen das Kommando nur einmal mit neuen Medien laufen lassen. Anschließend können Sie DVD+RWs, wie in den vorigen Abschnitten beschrieben, brennen.

Wenn Sie auf einer DVD+RW ein neues Dateisystem erstellen wollen, brauchen Sie die DVD+RW vorher nicht zu löschen. Überschreiben Sie einfach das vorige Dateisystem indem Sie eine neue Session anlegen:

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

Mit dem DVD+RW-Format ist es leicht, Daten an eine vorherige Aufnahme anzuhängen. Dazu wird eine neue Session mit der schon bestehenden zusammengeführt. Es wird keine Multi-Session geschrieben, sondern growisofs(1) *vergrößert* das ISO-9660-Dateisystem auf dem Medium.

Das folgende Kommando fügt weitere Daten zu einer vorher erstellten DVD+RW hinzu:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Wenn Sie eine DVD+RW erweitern, verwenden Sie dieselben mkisofs(8)-Optionen wie beim Erstellen der DVD+RW.

> **Anmerkung:** Um die Kompatibilität mit DVD-ROM-Laufwerken zu gewährleisten, wollen Sie vielleicht die Option `-dvd-compat` einsetzen. Zu einem DVD+RW-Medium können Sie mit dieser Option auch weiterhin Daten hinzufügen.

Wenn Sie das Medium aus irgendwelchen Gründen doch löschen müssen, verwenden Sie den nachstehenden Befehl:

```
# growisofs -Z /dev/cd0=/dev/zero
```

19.7.6. DVD-RW-Medien benutzen

Eine DVD-RW kann mit zwei Methoden beschrieben werden: *Sequential-Recording* oder *Restricted-Overwrite*. Voreingestellt ist Sequential-Recording.

Eine neue DVD-RW kann direkt beschrieben werden; sie muss nicht vorher formatiert werden. Allerdings muss eine DVD-RW, die mit Sequential-Recording aufgenommen wurde, zuerst gelöscht werden, bevor eine neue Session aufgenommen werden kann.

Der folgende Befehl löscht eine DVD-RW im Sequential-Recording-Modus:

```
# dvd+rw-format -blank=full /dev/cd0
```

> **Anmerkung:** Das vollständige Löschen (`-blank=full`) dauert mit einem 1x Medium ungefähr eine Stunde. Wenn die DVD-RW im Disk-At-Once-Modus (DAO) aufgenommen wurde, kann Sie mit der Option `-blank`

schneller gelöscht werden. Um eine DVD-RW im DAO-Modus zu brennen, benutzen Sie das folgende Kommando:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Die Option `-use-the-force-luke=dao` sollte nicht erforderlich sein, da growisofs(1) den DAO-Modus erkennt.

Der Restricted-Overwrite-Modus sollte mit jeder DVD-RW verwendet werden, da er flexibler als der voreingestellte Sequential-Recording-Modus ist.

Um Daten auf eine DVD-RW im Sequential-Recording-Modus zu schreiben, benutzen Sie dasselbe Kommando wie für die anderen DVD-Formate:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

Wenn Sie weitere Daten zu einer Aufnahme hinzufügen wollen, benutzen Sie die Option `-M` von growisofs(1). Werden die Daten im Sequential-Recording-Modus hinzugefügt, wird eine neue Session erstellt. Das Ergebnis ist ein Multi-Session-Medium.

Eine DVD-RW im Restricted-Overwrite-Modus muss nicht gelöscht werden, um eine neue Session aufzunehmen. Sie können das Medium einfach mit der Option `-Z` überschreiben, ähnlich wie bei DVD+RW. Mit der Option `-M` können Sie das ISO-9660-Dateisystem, wie mit einer DVD+RW, vergrößern. Die DVD enthält danach eine Session.

Benutzen sie das nachstehende Kommando, um den Restricted-Overwrite-Modus einzustellen:

```
# dvd+rw-format /dev/cd0
```

Das folgende Kommando stellt den Modus wieder auf Sequential-Recording zurück:

```
# dvd+rw-format -blank=full /dev/cd0
```

19.7.7. Multi-Session

Nur wenige DVD-ROM-Laufwerke können Multi-Session-DVDs lesen. Meist lesen die Spieler nur die erste Session. Mehrere Sessions werden von DVD+R, DVD-R und DVD-RW im Sequential-Recording-Modus unterstützt. Im Modus Restricted-Overwrite gibt es nur eine Session.

Wenn das Medium noch nicht geschlossen ist, erstellt das nachstehende Kommando eine neue Session auf einer DVD+R, DVD-R oder DVD-RW im Sequential-Recording-Modus:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Wird diese Kommandozeile mit DVD+RW- oder DVD-RW-Medien im Restricted-Overwrite-Modus benutzt, werden die neuen Daten mit den Daten der bestehenden Session zusammengeführt. Das Medium enthält danach eine Session. Auf diesem Weg werden neue Daten zu einer bestehenden Session hinzugefügt.

Anmerkung: Für den Anfang und das Ende einer Session wird auf dem Medium zusätzlicher Platz verbraucht. Um den Speicherplatz auf dem Medium optimal auszunutzen, sollten Sie daher Sessions mit vielen Daten hinzufügen. Auf ein DVD+R-Medium passen maximal 154 Sessions, 2000 Sessions auf ein DVD-R-Medium und 127 Sessions auf eine DVD+R Double Layer.

19.7.8. Weiterführendes

Das Kommando `dvd+rw-mediainfo /dev/cd0` zeigt Informationen über eine im Laufwerk liegende DVD an.

Weiteres zu den **dvd+rw-tools** lesen Sie bitte in der Hilfeseite growisofs(1), auf der dvd+rw-tools Web-Seite (http://fy.chalmers.se/~appro/linux/DVD+RW/) oder in den Archiven der cdwrite-Mailingliste (http://lists.debian.org/cdwrite/).

19.7.9. DVD-RAM

19.7.9.1. Konfiguration

DVD-RAM-fähige Brenner werden sowohl mit SCSI- als auch mit ATAPI-Schnittstelle angeboten. Verwenden Sie ein ATAPI-Gerät, müssen Sie den DMA-Modus aktivieren. Dazu fügen Sie die folgende Zeile in `/boot/loader.conf` ein:

```
hw.ata.atapi_dma="1"
```

19.7.9.2. Das Medium vorbereiten

Wie weiter oben in diesem Kapitel bereits erwähnt, kann man eine DVD-RAM mit einer Wechselplatte vergleichen. Wie diese muss auch eine DVD-RAM vor dem ersten Einsatz “vorbereitet” werden. In unserem Beispiel wird das gesamte Medium mit dem Standard-UFS2-Dateisystem formatiert.

Dazu geben Sie als `root` bei eingelegter DVD-RAM die folgenden Befehle ein:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

Denken Sie dabei daran, dass Sie gegebenenfalls die Gerätedatei (hier `acd0`) an Ihre Konfiguration anpassen müssen.

19.7.9.3. Das Medium einsetzen

Nachdem Sie das Medium vorbereitet haben, können Sie das DVD-RAM-Medium in Ihren Verzeichnisbaum einhängen:

```
# mount /dev/acd0 /mnt
```

Danach können Sie schreibend und lesend auf das Medium zugreifen.

19.8. Disketten benutzen

Original von Julio Merino. Umgeschrieben von Martin Karlsson.

Disketten sind nützlich, wenn kein anderes bewegliches Speichermedium vorhanden ist oder wenn nur kleine Datenmengen transferiert werden sollen.

Dieser Abschnitt beschreibt die Handhabung von Disketten unter FreeBSD. Hauptsächlich geht es um die Formatierung und Benutzung von 3,5 Zoll Disketten, doch lassen sich die Konzepte leicht auf Disketten anderer Formate übertragen.

19.8.1. Disketten formatieren

19.8.1.1. Die Gerätedateien

Wie auf jedes andere Gerät auch, greifen Sie auf Disketten über Einträge im Verzeichnis `/dev` zu. Verwenden Sie dazu die Einträge `/dev/fdN`.

19.8.1.2. Formatierung

Bevor eine Diskette benutzt werden kann, muss Sie (low-level) formatiert werden, was normalerweise der Hersteller schon gemacht hat. Sie können die Diskette allerdings noch einmal formatieren, um das Medium zu überprüfen. Es ist möglich, die Kapazität der Diskette zu verändern, allerdings sind die meisten Disketten auf 1440 kB ausgelegt.

Mit fdformat(1) formatieren Sie eine Diskette. Das Kommando erwartet die Angabe eines Gerätenamens.

Achten Sie bei der Formatierung auf Fehlermeldungen, die schlechte Speichermedien anzeigen.

19.8.1.2.1. Disketten formatieren

Die Disketten werden mithilfe der Gerätedatei `/dev/fdN` formatiert. Legen Sie eine 3,5 Zoll Diskette in Ihr Laufwerk ein und führen das folgende Kommando aus:

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

19.8.2. Das Disklabel

Nach dem Formatieren muss auf der Diskette ein Disklabel erstellt werden. Das Disklabel wird später zerstört, ist aber notwendig, um die Größe und Geometrie der Diskette zu erkennen.

Das Disklabel gilt für die ganze Diskette und enthält alle Informationen über die Geometrie der Diskette. Eine Liste der möglichen Geometrien finden Sie in `/etc/disktab`.

Erstellen Sie nun das Label mit bsdlabel(8):

```
# /sbin/bsdlabel -B -w /dev/fd0 fd1440
```

19.8.3. Das Dateisystem

Auf der Diskette muss nun ein Dateisystem erstellt werden (high-level Formatierung), damit FreeBSD von der Diskette lesen und auf sie schreiben kann. Das Disklabel wird durch das Anlegen eines Dateisystems zerstört. Falls Sie die Diskette später erneut formatieren wollen, müssen Sie dann auch ein neues Disklabel anlegen.

Sie können entweder UFS oder FAT als Dateisystem verwenden. Für Disketten ist FAT das beste Dateisystem.

Das folgende Kommando legt ein Dateisystem auf der Diskette an:

```
# /sbin/newfs_msdos /dev/fd0
```

Die Diskette kann nun benutzt werden.

19.8.4. Verwenden der Diskette

Zum Einhängen der Diskette in das Dateisystem verwenden Sie den Befehl mount_msdosfs(8). Sie können auch den Port `emulators/mtools` verwenden, um mit der Diskette zu arbeiten.

19.9. Bandmedien benutzen

Die wichtigsten Bandmedien sind 4mm, 8mm, QIC, Mini-Cartridge und DLT.

19.9.1. 4mm (DDS: Digital Data Storage)

Die 4mm-Bänder ersetzen mehr und mehr das QIC-Format als Backupmedium der Wahl für Workstations. Dieser Trend nahm stark zu, als Conner die Firma Archive, einen führenden Hersteller von QIC-Laufwerken, aufkaufte und die Produktion von QIC-Laufwerken stoppte. 4mm-Laufwerke sind klein und ruhig, haben aber nicht den gleichen Ruf der Zuverlässigkeit, den die 8mm-Laufwerke genießen. Die 4mm-Kassetten sind preiswerter und mit den Maßen 76,2 x 50,8 x 12,7 mm (3 x 2 x 0,5 Inch) kleiner als die 8mm-Kassetten. Sowohl die 4mm- als auch die 8mm-Magnetköpfe haben eine relativ kurze Lebensdauer, weil beide die gleiche Helical-Scan-Technik benutzen.

Der Datendurchsatz dieser Laufwerke beginnt bei etwa 150 kByte/s, Spitzenwerte liegen bei etwa 500 kByte/s. Die Datenkapazität liegt zwischen 1,3 GB und 2 GB. Die meisten Geräte haben eine Hardwarekompression eingebaut, die die Kapazität ungefähr verdoppelt. Es gibt Multi-Drive-Einheiten für Bandbibliotheken mit bis zu 6 Laufwerken in einem Gehäuse und automatischem Bandwechsel. Die Kapazität einer solchen Bibliothek liegt bei 240 GB.

Der Standard DDS-3 unterstützt nun Bandkapazitäten bis zu 12 GB (oder komprimiert 24 GB).

4mm-Laufwerke, ebenso wie 8mm-Laufwerke, verwenden Helical-Scan. Alle Vor- und Nachteile von Helical-Scan gelten sowohl für 4mm- als auch für 8mm-Laufwerke.

Bänder sollten nach 2.000 Banddurchläufen oder 100 vollen Backups ersetzt werden.

19.9.2. 8mm (Exabyte)

8mm-Bänder sind die verbreitetsten SCSI-Bandlaufwerke; sie sind das geeignetste Bandformat zum Austausch von Bändern. Fast an jedem Standort gibt es ein 8mm-Bandlaufwerk mit 2 GB. 8mm-Bänder sind zuverlässig, gut zu handhaben und arbeiten leise. Bandkassetten sind preiswert und klein mit 122 x 84 x 15 mm (4,8 x 3,3 x 0,6 Inch). Ein Nachteil der 8mm-Technik ist die relativ kurze Lebensdauer des Schreib-/Lesekopfs und der Bänder auf Grund der hohen Relativgeschwindigkeit des Bandes über die Köpfe hinweg.

Der Datendurchsatz liegt ungefähr zwischen 250 kByte/s und 500 kByte/s. Die Datenkapazität beginnt bei 300 MB und erreicht bis zu 7 GB bei den Spitzengeräten. Die meisten Geräte haben eine Hardwarekompression eingebaut, die die Kapazität ungefähr verdoppelt. Diese Laufwerke sind erhältlich in Form von Einzelgeräten oder als

Multi-Drive-Bandbibliotheken mit 6 Laufwerken und 120 Bändern in einem Gehäuse. Die Bänder werden von der Geräteeinheit automatisch gewechselt. Die Kapazität einer solchen Bibliothek liegt bei 840 GB und mehr.

Das Exabyte-Modell “Mammoth” unterstützt 12 GB auf einem Band (24 GB mit Kompression) und kostet etwa doppelt so viel wie ein konventionelles Bandlaufwerk.

Die Daten werden mittels Helical-Scan auf das Band aufgezeichnet, die Köpfe sind leicht schräg zum Medium angebracht (mit einem Winkel von etwa 6 Grad). Das Band wickelt sich 270 Grad um die Spule, die die Köpfe trägt. Die Spule dreht sich, während das Band darüber läuft. Das Resultat ist eine hohe Datendichte und eng gepackte Spuren, die von einem Rand des Bands zum gegenüberliegenden quer über das Band abgewinkelt verlaufen.

19.9.3. QIC

QIC-150-Bänder und -Laufwerke sind wohl der am weitesten verbreitete Bandtyp überhaupt. QIC-Bandlaufwerke sind die preiswertesten “seriösen” Backupgeräte, die angeboten werden. Der Nachteil dabei ist der hohe Preis der Bänder. QIC-Bänder sind im Vergleich zu 8mm- oder 4mm-Bändern bis zu fünf Mal teurer, wenn man den Preis auf 1 GB Datenkapazität umrechnet. Aber wenn Ihr Bedarf mit einem halben Dutzend Bänder abgedeckt werden kann, mag QIC die richtige Wahl sein.

QIC ist der *gängigste* Bandlaufwerkstyp. Jeder Standort hat ein QIC-Laufwerk der einen oder anderen Dichte. Aber gerade das ist der Haken an der Sache, QIC bietet eine große Anzahl verschiedener Datendichten auf physikalisch ähnlichen (manchmal gleichen) Bändern. QIC-Laufwerke sind nicht leise. Diese Laufwerke suchen lautstark die richtige Bandstelle, bevor sie mit der Datenaufzeichnung beginnen. Sie sind während des Lesens, Schreibens und Suchens deutlich hörbar.

Die Abmessungen der QIC-Kassetten betragen 152 x 102 x 17 mm (6 x 4 x 0,7 Inch).

Der Datendurchsatz liegt ungefähr zwischen 150 kByte/s und 500 kByte/s. Die Datenkapazität reicht von 40 MB bis zu 15 GB. Hardwarekompression ist in vielen der neueren QIC-Laufwerke eingebaut. QIC-Laufwerke werden heute seltener eingesetzt; sie werden von den DAT-Laufwerken abgelöst.

Die Daten werden auf dem Band in Spuren aufgezeichnet. Die Spuren verlaufen entlang der Längsachse des Bandmediums von einem Ende zum anderen. Die Anzahl der Spuren, und damit auch die Breite einer Spur, variiert mit der Kapazität des Laufwerks. Die meisten, wenn nicht alle neueren Laufwerke sind rückwärtskompatibel, zumindest zum Lesen (aber oft auch zum Schreiben). QIC hat einen guten Ruf bezüglich der Datensicherheit (die Mechanik ist einfacher und robuster als diejenige der Helical-Scan-Laufwerke).

Bänder sollten nach 5,000 Backups ersetzt werden.

19.9.4. DLT

DLT hat die schnellste Datentransferrate von allen hier aufgelisteten Gerätetypen. Das 1/2-Inch-Band (12,7 mm) befindet sich in einer Spulkassette mit den Abmessungen 101,6 x 101,6 x 25,4 mm (4 x 4 x 1 Inch). Die eine Seite der Kassette hat eine bewegliche Abdeckung. Der Laufwerksmechanismus öffnet diese Abdeckung und zieht die Bandführung heraus. Die Bandführung trägt ein ovales Loch, die das Laufwerk zum “Einhängen” des Bandes benutzt. Die Aufwickelspule befindet sich im Innern des Bandlaufwerks. Bei allen anderen hier besprochenen Bandkassetten (9-Spur-Bänder sind die einzige Ausnahme) befinden sich sowohl die Auf- als auch die Abwickelspule im Inneren der Bandkassette.

Der Datendurchsatz liegt bei etwa 1,5 MBytes/s, der dreifache Durchsatz der 4mm-, 8mm- oder QIC-Bandlaufwerke. Die Datenkapazität reicht von 10 GB bis 20 GB für Einfachlaufwerke. Auch Mehrfachbandgeräte sind erhältlich,

sowohl als Bandwechsler wie auch als Multi-Drive-Bandbibliotheken, die Platz für 5 bis 900 Bänder verteilt auf 1 bis 20 Laufwerke enthalten, mit einer Speicherkapazität von 50 GB bis 9 TB.

Mit Kompression unterstützt das Format DLT Type IV bis zu 70 GB Kapazität.

Die Daten werden auf dem Band in Spuren aufgezeichnet, die parallel zur Bewegungsrichtung verlaufen (gerade so wie bei den QIC-Bändern). Zwei Spuren werden dabei gleichzeitig beschrieben. Die Lebenszeit der Lese- und Schreibköpfe sind relativ lang; denn sobald das Band anhält, gibt es keine Relativbewegung mehr zwischen den Köpfen und dem Band.

19.9.5. AIT

AIT ist ein neues Format von Sony, das (mit Kompression) bis zu 50 GB pro Band speichern kann. Die Bänder haben einen Speicherchip, der einen Index mit dem Inhalt des Bandes anlegt. Dieser Index kann vom Bandlaufwerk zur schnellen Bestimmung der Lage von Dateien auf dem Band benutzt werden, während andere Bänder einige Minuten zur Lokalisierung benötigen.

Entsprechende Software wie etwa **SAMS:Alexandria** können 40 oder mehr AIT-Bandbibliotheken verarbeiten, indem sie direkt mit dem Speicherchip des Bandes kommunizieren, wenn der Bandinhalt am Bildschirm dargestellt werden soll oder bestimmt werden soll, welche Dateien auf welchem Band gespeichert sind, oder um das richtige Band zu lokalisieren, zu laden und Daten vom Band zurückzuspielen. Bibliotheken dieser Art liegen in der Preiskategorie von $20,000, womit sie etwas aus dem Hobbymarkt herausfallen.

19.9.6. Die erste Benutzung eines neuen Bands

Der Versuch ein neues, vollkommen leeres Band ohne weiteres zu lesen oder zu beschreiben wird schief gehen. Auf der Konsole werden dann Meldungen ähnlich wie folgt ausgegeben:

```
sa0(ncr1:4:0): NOT READY asc:4,1
0(ncr1:4:0): Logical unit is in process of becoming ready
```

Das Band enthält nämlich keinen Identifier-Block (Blocknummer 0). Alle QIC-Bandlaufwerke seit der Einführung des QIC-525-Standards schreiben einen Identifier-Block auf das Band. Es gibt zwei Lösungen:

- `mt fsf 1` veranlasst das Bandlaufwerk einen Identifier-Block auf das Band zu schreiben.
- Das Band durch Drücken des Bandauswurfknopfs an der Vorderseite des Bandgeräts auswerfen.

  Danach das Band wieder einlegen und mit `dump` Daten auf das Band übertragen.

  Das Kommando `dump` gibt die Meldung `DUMP: End of tape detected` zurück und die Konsole zeigt: `HARDWARE FAILURE info:280 asc:80,96`.

  Das Band zurückspulen mit dem Kommando: `mt rewind`.

  Nachfolgende Bandoperationen werden dann erfolgreich ausgeführt.

19.10. Was ist mit Backups auf Disketten?

19.10.1. Kann ich Disketten zum Backup meiner Daten verwenden?

Disketten sind kein wirklich geeignetes Medium für Backups aus folgenden Gründen:

- Disketten sind unzuverlässig, besonders langfristig.
- Speichern und Wiederherstellen ist sehr langsam.
- Sie haben eine sehr eingeschränkte Kapazität (Die Zeiten sind längst vorbei, wo eine ganze Festplatte auf ein Dutzend Disketten oder so gespeichert werden konnte).

Wenn jedoch keine andere Möglichkeit zum Datenbackup vorhanden ist, dann sind Disketten immer noch besser als gar kein Backup.

Wenn man gezwungen ist Disketten zu verwenden, dann sollte man auf eine gute Qualität achten. Disketten, die schon einige Jahre im Büro herumgelegen haben, sind eine schlechte Wahl. Ideal sind neue Disketten von einem renommierten Hersteller.

19.10.2. Wie mache ich ein Backup auf Disketten?

Die beste Art eines Diskettenbackups ist der Befehl tar(1) mit der Mehrfachband-Option `-M`, die es ermöglicht ein Backup über mehrere Disketten zu verteilen.

Ein Backup aller Dateien im aktuellen Verzeichnis einschließlich aller Unterverzeichnisse wird durch den folgenden Befehl veranlasst (als `root`):

```
# tar Mcvf /dev/fd0 *
```

Wenn die erste Diskette voll ist, meldet sich tar(1) und verlangt einen Diskettenwechsel (weil tar(1) unabhängig vom Medium arbeitet, wird das nächste Band (Volume) verlangt, was in diesem Zusammenhang eine Diskette bedeutet), in etwa wie folgt:

```
Prepare volume #2 for /dev/fd0 and hit return:
```

Dies wird mit steigender Volumenzahl wiederholt, bis alle angegebenen Dateien archiviert sind.

19.10.3. Können Diskettenbackups komprimiert werden?

Leider erlaubt es tar(1) nicht, die Option `-z` für Multi-Volume-Archive zu verwenden. Man kann natürlich alle Dateien mit gzip(1) komprimieren, sie mit tar(1) auf die Disketten aufspielen, und dann die Dateien wieder gunzip(1) dekomprimieren!

19.10.4. Wie werden Diskettenbackups wieder hergestellt?

Zur Wiederherstellung des gesamten Archivs verwendet man:

```
# tar Mxvf /dev/fd0
```

Eine Methode um nur bestimmte Dateien wieder her zu stellen ist mit der ersten Diskette den folgenden Befehl auszuführen:

```
# tar Mxvf /dev/fd0 filename
```

tar(1) wird dann die folgenden Disketten anfordern, bis die benötigte Datei gefunden ist.

Wenn man die Diskette kennt, auf der sich die Datei befindet, kann man alternativ diese Diskette auch direkt einlegen und den gleichen Befehl wie oben verwenden. Man beachte, dass, falls die erste Datei eine Fortsetzung einer Datei von einer der vorigen Disketten ist, tar(1) die Warnung ausgibt, dass diese Datei nicht wiederhergestellt werden kann, selbst dann, wenn dies gar nicht verlangt wurde!

19.11. Backup-Strategien

Beigetragen von Lowell Gilbert.

Wenn Sie eine eigene Backup-Strategie planen, müssen Sie darauf achten, dass jedes der folgenden Probleme von Ihrer Strategie abgedeckt wird:

- Plattendefekte.
- Versehentliches Löschen von Dateien.
- Eine nicht vorhersehbare Korrumpierung von Dateien.
- Die vollständige Zerstörung Ihres Systems, etwa durch ein Feuer. Dazu gehört auch die Zerstörung von Backups, die am gleichen Ort aufbewahrt werden.

Es ist nicht nur möglich, dass ein System für jedes dieser Probleme eine eigene (oft völlig unterschiedliche) Strategie benötigt. Es ist vielmehr unwahrscheinlich (sieht man von Systemen ab, die keine wichtigen Daten enthalten), dass eine Technik alle Problembereiche abdecken kann.

Häufig verwendeten Techniken sind unter anderen:

- Die Archivierung des kompletten Systems auf externen Datenträgern, die an einem gesonderten Ort aufbewahrt werden. Dieser Ansatz schützt zwar vor allen oben angeführten Problemen, ist aber zeitaufwändig. Auch eine Wiederherstellung des Systems ist nicht ohne weiteres möglich. Zwar können Sie Kopien Ihrer Backups auch vor Ort und/oder auf online zugängigen Systemen aufbewahren, was aber nichts daran ändert, dass eine Wiederherstellung, insbesondere für nicht privilegierte Benutzer, nach wie vor nicht ohne weiteres möglich ist.
- Dateisystem-Snapshots. Diese Technik hilft zwar nur gegen das versehentliche Löschen von Dateien, in einem solchen Fall ist sie aber *äußerst* hilfreich. Vorteile dieser Technik sind außerdem die leichte und schnelle Implementierung und Handhabung.
- Das Erstellen von Kopien ganzer Dateisysteme und/oder Platten (etwa durch einen periodischen rsync(1)-Transfer des kompletten Systems). Diese Technik ist insbesondere in Netzwerken mit besonderen Anforderungen nützlich. Der Schutz vor Plattendefekten ist allerdings schlechter als beim Einsatz von RAID. Die Fähigkeiten zur Wiederherstellung gelöschter Dateien sind mit denen von UFS-Snapshots vergleichbar. Ob diese Technik für Sie geeignet ist, hängt also letztlich von Ihren Anforderungen ab.

- RAID. Minimiert oder vermeidet Ausfallzeiten, die durch einen Plattendefekt verursacht werden könnten. Zwar können Plattendefekte (aufgrund der höheren Anzahl verwendeter Platten) häufiger auftreten, sie stellen aber dann kein so akutes Problem dar.
- Das Überprüfen von Datei-Fingerprints durch mtree(8). Dabei handelt es sich zwar um keine Backup-Technik im eigentlichen Sinne, Sie werden durch den Einsatz dieses Werkzeugs aber informiert, dass Sie auf Ihre Backups zurückgreifen müssen. Dies ist insbesondere beim Einsatz von Offline-Backups von großer Bedeutung. Daher sollte diese Technik regelmäßig eingesetzt werden.

Es gibt noch zahlreiche weitere Techniken, von denen aber viele nur Variationen der eben beschriebenen Techniken sind. Spezielle Anforderungen erfordern dabei in der Regel auch spezielle Backup-Techniken (so erfordert das Backup einer aktiven Datenbank in der Regel ein auf die eingesetzte Datenbank-Software abgestimmtes Verfahren). Entscheidend ist daher immer, gegen welche Gefahren Sie sich schützen und wie Sie diesen Schutz realisieren wollen.

19.12. Datensicherung

Die drei wichtigsten Programme zur Sicherung von Daten sind dump(8), tar(1) und cpio(1).

19.12.1. Sichern und Wiederherstellen

`dump` und `restore` sind die traditionellen Backup-Programme in UNIX Systemen. Sie betrachten das Laufwerk als eine Ansammlung von Blöcken, operieren also unterhalb des Abstraktionslevels von Dateien, Links und Verzeichnissen, die die Grundlage des Dateisystemkonzepts bilden. Im Gegensatz zu anderen Backup-Programmen sichert `dump` ein ganzes Dateisystem auf einem Gerät. Es ist nicht möglich nur einen Teil des Dateisystems, oder einen Verzeichnisbaum, der mehr als ein Dateisystem umfasst, zu sichern. Das `dump`-Kommando schreibt keine Dateien oder Verzeichnisse auf das Band, sondern die Blöcke, aus denen Dateien und Verzeichnisse bestehen. Wenn `restore` für das Extrahieren von Daten verwendet wird, werden temporäre Dateien standardmäßig in `/tmp/` abgelegt - wenn Sie von einer Platte mit einem kleinen `/tmp`-Verzeichnis zurücksichern, müssen Sie möglicherweise die Umgebungsvariable `TMPDIR` auf ein Verzeichnis mit mehr freiem Speicherplatz setzen, damit die Wiederherstellung gelingt.

> **Anmerkung:** Wenn Sie mit `dump` das Root-Verzeichnis sichern, werden `/home`, `/usr` und viele andere Verzeichnisse nicht gesichert, da dies normalerweise Mountpunkte für andere Dateisysteme oder symbolische Links zu diesen Dateisystemen sind.

`dump` hat einige Eigenarten, die noch aus den frühen Tagen der Version 6 von AT&T UNIX (ca. 1975) stammen. Die Parameter sind für 9-Spur-Bänder (6250 bpi) voreingestellt, nicht auf die heute üblichen Medien hoher Dichte (bis zu 62.182 ftpi). Bei der Verwendung der Kapazitäten moderner Bandlaufwerke muss diese Voreinstellung auf der Kommandozeile überschrieben werden.

`rdump` und `rrestore` können Daten über Netzwerk auf ein Band, das sich in einem Laufwerk eines anderen Computers befindet, überspielen. Beide Programme benutzen die Funktionen rcmd(3) und ruserok(3) zum Zugriff auf das entfernte Bandlaufwerk. Daher muss der Anwender, der das Backup durchführt, auf dem entfernten Rechner in `.rhosts` eingetragen sein.

Die Argumente zu `rdump` und `rrestore` müssen zur Verwendung auf dem entfernten Computer geeignet sein. Wenn Sie zum Beispiel mit `rdump` von einem FreeBSD-Rechner aus auf ein Exabyte-Bandlaufwerk einer Sun mit Namen `komodo` zugreifen möchten, setzen Sie das folgende Kommando ab:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Zum Ausführen dieses Kommandos müssen Sie auf dem entfernten Rechner in `.rhosts` eingetragen sein. Die r-Kommandos sind ein großes Sicherheitsrisiko, daher sollten Sie deren Verwendung sorgfältig abwägen.

Es ist auch möglich, `dump` und `restore` über eine gesicherte Verbindung mit `ssh` einzusetzen:

Beispiel 19-1. dump mit ssh benutzen

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
   targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-l0.gz
```

Sie können ebenfalls mit der internen Methode von `dump` auf entfernte Rechner zugreifen, indem Sie die Umgebungsvariable `RSH` setzen:

Beispiel 19-2. dump über ssh mit gesetzter `RSH` benutzen

```
# RSH=/usr/bin/ssh /sbin/dump -0uan -f tatargetuser@targetmachine.example.com:/dev/sa0 /usr
```

19.12.2. `tar`

tar(1) stammt ebenfalls aus Version 6 von AT&T UNIX (ca. 1975). `tar` arbeitet mit dem Dateisystem, denn es schreibt Dateien und Verzeichnisse auf das Band. `tar` unterstützt zwar nicht alle Optionen, die bei cpio(1) zur Verfügung stehen, aber dafür erfordert es auch nicht die ungewöhnliche Kommando-Pipeline, die von `cpio` verwendet wird.

Um Daten mit `tar` auf ein an einer Sun-Workstation (namens `komodo`) angeschlossenes Exabyte-Bandlaufwerk zu archivieren, geben Sie Folgendes ein:

```
# tar cf - . | rsh komodo dd of=tape-device obs=20b
```

Wenn Sie Bedenken bezüglich der Sicherheit beim Backup über das Netz haben, sollten Sie `ssh` anstatt `rsh` benutzen.

19.12.3. Cpio

cpio(1) ist das ursprüngliche Programm von UNIX Systemen zum Dateitransfer mit magnetischen Medien. `cpio` hat (neben vielen anderen Leistungsmerkmalen) Optionen zum Byte-Swapping, zum Schreiben einer Anzahl verschiedener Archivformate und zum Weiterleiten von Daten an andere Programme über eine Pipeline. Dieses letzte Leistungsmerkmal macht `cpio` zu einer ausgezeichneten Wahl für Installationsmedien. Leider kann `cpio` keine Dateibäume durchlaufen, so dass eine Liste der zu bearbeitenden Dateien über `stdin` angegeben werden muss.

`cpio` unterstützt keine Backups über das Netzwerk. Man kann aber eine Pipeline und `rsh` verwenden, um Daten an ein entferntes Bandlaufwerk zu senden.

```
# for f in directory_list; do
```

```
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host "cat > backup_device"
```

Dabei steht *directory_list* für eine Aufzählung der Verzeichnisse, die Sie sichern wollen. *user@host* gibt den Benutzer auf dem Zielrechner an, der die Sicherung laufen lässt. Der Ort der Sicherung wird durch *backup_device* angegeben (z.B. `/dev/nsa0`).

19.12.4. pax

pax(1) ist die Antwort von IEEE/POSIX auf `tar` und `cpio`. Über die Jahre hinweg sind die verschiedenen Versionen von `tar` und `cpio` leicht inkompatibel geworden. Daher hat POSIX, statt eine Standardisierung zwischen diesen auszufechten, ein neues Archivprogramm geschaffen. `pax` versucht viele der unterschiedlichen `cpio`- und `tar`-Formate zu lesen und zu schreiben, außerdem einige neue, eigene Formate. Die Kommandostruktur ähnelt eher `cpio` als `tar`.

19.12.5. Amanda

Amanda (Advanced Maryland Network Disk Archiver) ist ein Client/Server-Backupsystem, nicht nur ein einzelnes Programm. Ein **Amanda**-Server kann auf einem einzigen Bandlaufwerk Datensicherungen von jeder beliebigen Anzahl von Computern speichern, sofern auf diesen jeweils ein **Amanda**-Client läuft und sie über Netzwerk mit dem **Amanda**-Server verbunden sind.

Ein häufiges Problem bei Standorten mit einer Anzahl großer Festplatten ist, dass das Kopieren der Daten auf Band langsamer vor sich geht als solche Daten anfallen. **Amanda** löst dieses Problem durch Verwendung einer “Holding Disk”, einer Festplatte zum gleichzeitigen Zwischenspeichern mehrerer Dateisysteme.

Für Datensicherungen über einen längeren Zeitraum erzeugt **Amanda** “Archivsets” von allen Dateisystemen, die in **Amandas** Konfigurationsdatei genannt werden. Ein Archivset ist eine Gruppe von Bändern mit vollen Backups und Reihen von inkrementellen (oder differentiellen) Backups, die jeweils nur die Unterschiede zum vorigen Backup enthalten. Zur Wiederherstellung von beschädigten Dateisystemen benötigt man Das Letzte volle Backup und alle darauf folgenden inkrementellen Backups.

Die Konfigurationsdatei ermöglicht die Feineinstellung der Backups und des Netzwerkverkehrs von **Amanda**. **Amanda** kann zum Schreiben der Daten auf das Band jedes der oben beschriebenen Backuprogramme verwenden. **Amanda** ist nicht Teil des Basissystems, Sie müssen **Amanda** über die Ports-Sammlung oder als Paket installieren.

19.12.6. Tue nichts

“Tue nichts” ist kein Computerprogramm, sondern die am häufigsten angewendete Backupstrategie. Diese kostet nichts, man muss keinen Backup Plan befolgen, einfach nur nein sagen. Wenn etwas passiert, einfach grinsen und ertragen!

Wenn Ihre Zeit und Ihre Daten nicht so wichtig sind, dann ist die Strategie “Tue nichts” das geeignetste Backup-Programm für Ihren Computer. Aber UNIX ist ein nützliches Werkzeug, Sie müssen damit rechnen, dass Sie innerhalb von sechs Monaten eine Sammlung von Dateien haben, die für Sie wertvoll geworden sind.

“Tue nichts” ist die richtige Backupmethode für `/usr/obj` und andere Verzeichnisbäume, die vom Computer exakt wiedererzeugt werden können. Ein Beispiel sind die Dateien, die diese Handbuchseiten darstellen – sie wurden aus

Quelldateien im Format SGML erzeugt. Es ist nicht nötig, Sicherheitskopien der Dateien in den sekundären Formaten wie etwa HTML zu erstellen. Die Quelldateien in SGML sollten jedoch in die regelmäßigen Backups mit einbezogen werden.

19.12.7. Welches Backup-Programm ist am Besten?

`dump`, *Punkt und Schluss.* Elizabeth D. Zwicky hat alle hier genannten Backup-Programme bis zur Erschöpfung ausgetestet. Ihre eindeutige Wahl zur Sicherung aller Daten mit Berücksichtigung aller Besonderheiten von UNIX Dateisystemen ist `dump`.

Elizabeth erzeugte Dateisysteme mit einer großen Vielfalt ungewöhnlicher Bedingungen (und einiger gar nicht so ungewöhnlicher) und testete jedes Programm durch ein Backup und eine Wiederherstellung dieser Dateisysteme. Unter den Besonderheiten waren Dateien mit Löchern, Dateien mit Löchern und einem Block mit Null-Zeichen, Dateien mit ausgefallenen Buchstaben im Dateinamen, unlesbare und nichtschreibbare Dateien, Gerätedateien, Dateien, deren Länge sich während des Backups ändert, Dateien, die während des Backups erzeugt und gelöscht werden, u.v.m. Sie berichtete über ihre Ergebnisse in LISA V im Oktober 1991, s. Torture-testing Backup and Archive Programs (http://www.coredumps.de/doc/dump/zwicky/testdump.doc.html).

19.12.8. Die Wiederherstellung in einem Notfall

19.12.8.1. Vor dem Unglück

Es sind nur vier Vorkehrungen zu treffen, um auf jedes erdenkliche Unglück vorbereitet zu sein.

Als erstes drucken Sie das bsdlabel jeder Ihrer Festplatten (z.B. mittels `bsdlabel da0 | lpr`), die Partitions- und Dateisystemtabelle jeder Festplatte (mit `/etc/fstab`) sowie alle Bootmeldungen, jeweils in zweifacher Ausfertigung.

Zweitens brennen Sie eine “livefs”-CD. Diese CD-ROM enthält alle nötigen Programme, um in einen Reperaturmodus zu starten, aus dem heraus Sie unter anderem dump(8), restore(8), fdisk(8), bsdlabel(8), newfs(8) sowie mount(8) starten können. ISO-Abbilder für das “livefs”-System finden Sie unter ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/9.1/FreeBSD-9.1-RELEASE-i386-livefs.iso.

Drittens, machen Sie oft Backups auf Band. Jede Änderung seit Ihrem letzten Backup kann unwiederbringlich verloren gehen. Versehen Sie die Backup-Bänder mit Schreibschutz.

Viertens, testen Sie das in Schritt 2 erstellte “livefs”-System sowie die für das Backup notwendigen Bänder. Dokumentieren Sie diesen Test und bewahren Sie diese Notizen zusammen mit der “livefs”-CD und den Bändern auf. Wenn der Ernstfall eintritt, werden Sie vielleicht so genervt sein, dass Sie ohne Ihre Notizen vielleicht das Backup auf Ihren Bändern zerstören. (Wie das geht? Man braucht nur unglücklicherweise den Befehl `tar cvf /dev/sa0` einzugeben um ein Band zu überschreiben).

Als zusätzliche Sicherheitsvorkehrung, kann man jeweils die “livefs”-CD und Bänder doppelt erstellen. Eine der Kopien sollte an einem entfernten Standort aufbewahrt werden. Ein entfernter Standort ist NICHT der Keller im gleichen Bürogebäude. Eine Anzahl von Firmen im World Trade Center musste diese Lektion auf die harte Tour lernen. Ein entfernter Standort sollte von Ihrem Computer und Ihren Festplatten physikalisch durch eine erhebliche Entfernung getrennt sein.

19.12.8.2. Nach dem Unglück

Die Schlüsselfrage ist, ob Ihre Hardware überlebt hat. Denn da Sie ja regelmäßig Backups angefertigt haben, brauchen Sie sich um die Software keine Sorgen zu machen.

Falls die Hardware beschädigt wurde, ersetzen Sie zuerst die defekten Teile bevor Sie den Computer benutzen.

Falls die Hardware funktioniert, legen Sie die “livefs”-CD in das Laufwerk ein und starten den Rechner, wodurch das originale Installationsprogramm von FreeBSD gestartet wird. Legen Sie zuerst Ihr Land fest. Danach öffnen Sie das Menü `Fixit -- Repair mode with CDROM/DVD/floppy or start a shell.` und wählen den Eintrag CDROM/DVD -- Use the live filesystem CDROM/DVD aus. `restore` und die anderen Programme, die Sie benötigen, befinden sich dann im Verzeichnis `/mnt2/rescue`.

Stellen Sie die Dateisysteme nacheinander wieder her.

Versuchen Sie die Root-Partition Ihrer ersten Festplatte einzuhängen (z.B. mit `mount /dev/sd0a /mnt`). Wenn das Bsdlabel beschädigt wurde, benutzen Sie `bsdlabel` um die Platte neu zu partitionieren und zu benennen und zwar so, dass die Festplatte mit dem Label übereinstimmt, das Sie ausgedruckt und aufbewahrt haben.

Verwenden Sie `newfs` um neue Dateisysteme auf den Partitionen anzulegen. Hängen Sie nun die Root-Partition der Festplatte mit Schreibzugriff ein (mit `mount -u -o rw /mnt`). Benutzen Sie Ihr Backup-Programm um die Daten für das jeweilige Dateisystem aus den Backup-Bändern wieder her zu stellen (z.B. durch `restore vrf /dev/sta`). Hängen Sie das Dateisystem wieder aus (z.B. durch `umount /mnt`). Wiederholen Sie diesen Ablauf für jedes betroffene Dateisystem.

Sobald Ihr System wieder läuft, machen Sie gleich wieder ein vollständiges Backup auf neue Bänder. Denn die Ursache für den Absturz oder den Datenverlust kann wieder zuschlagen. Eine weitere Stunde, die Sie jetzt noch dranhängen, kann Ihnen später ein weiteres Missgeschick ersparen.

19.13. Netzwerk-, speicher- und dateibasierte Dateisysteme

Verbessert und neu strukturiert von Marc Fonvieille.

Neben Laufwerken, die sich physikalisch im Rechner befinden wie Diskettenlaufwerke, CDs, Festplatten usw., kann FreeBSD auch mit anderen Laufwerken, den *virtuellen Laufwerken*, umgehen.

Dazu zählen Netzwerkdateisysteme wie Network Filesystem und Coda, speicher- und dateibasierte Dateisysteme.

Abhängig von der verwendeten FreeBSD Version werden speicher- und dateibasierte Dateisysteme mit unterschiedlichen Werkzeugen angelegt.

Anmerkung: Gerätedateien werden unter FreeBSD automatisch von devfs(5) angelegt.

19.13.1. Dateibasierte Laufwerke unter FreeBSD

Unter FreeBSD werden virtuelle Laufwerke (md(4)) mit mdconfig(8) erzeugt. Dazu muss das Modul md(4) geladen sein oder das entsprechende Gerät in der Kernelkonfiguration aktiviert sein:

```
device md
```

Mit mdconfig(8) können drei verschiedene virtuelle Laufwerke angelegt werden: speicherbasierte Laufwerke, deren Speicher von malloc(9) zur Verfügung gestellt wird, oder dateibasierte Laufwerke, deren Speicher von einer Datei oder dem Swap-Bereich zur Verfügung gestellt wird. Eine mögliche Anwendung ist das Einhängen von Dateien, die Abbilder von CD-ROMs oder Disketten enthalten.

Das Abbild eines Dateisystems wird wie folgt eingehangen:

Beispiel 19-3. Einhängen eines existierenden Abbildes unter FreeBSD

```
# mdconfig -a -t vnode -f diskimage -u 0
# mount /dev/md0 /mnt
```

Ein neues Dateisystem-Abbild erstellen Sie mit mdconfig(8) wie folgt:

Beispiel 19-4. Erstellen eines dateibasierten Laufwerks mit `mdconfig`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# bsdlabel -w md0 auto
# newfs md0a
/dev/md0a: 5.0MB (10224 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
 160, 2720, 5280, 7840
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

Wenn Sie keine Gerätenummer mit dem Schalter `-u` angeben, wird von md(4) automatisch eine ungenutzte Gerätenummer zugewiesen. Das zugewiesene Gerät wird auf der Standardausgabe ausgegeben (zum Beispiel `md4`). Weitere Informationen entnehmen Sie bitte der Hilfeseite mdconfig(8).

Das Werkzeug mdconfig(8) ist sehr nützlich, doch muss man viele Kommandos absetzen, um ein dateibasiertes Dateisystem zu erstellen. FreeBSD enthält das Werkzeug mdmfs(8), das die notwendigen Schritte in einem Befehl zusammenfasst. Es konfiguriert mit mdconfig(8) ein md(4)-Laufwerk, erstellt darauf mit newfs(8) ein Dateisystem und hängt es anschließend mit mount(8) ein. Das virtuelle Laufwerk aus dem obigen Beispiel kann somit einfach mit den nachstehenden Befehlen erstellt werden:

Beispiel 19-5. Mit `mdmfs` ein dateibasiertes Dateisystem erstellen

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0        4718    4  4338     0%    /mnt
```

Wenn sie die Option `md` ohne Gerätenummer verwenden, wählt md(4) automatisch ein ungenutztes Gerät aus. Weitere Einzelheiten entnehmen Sie bitte der Hilfeseite mdmfs(8).

19.13.2. Speicherbasierte Laufwerke unter FreeBSD

Verwenden Sie ein speicherbasiertes Dateisystem, sollten Sie die Option “swap backing” aktivieren. Setzen Sie diese Option, heißt dies allerdings nicht, dass das speicherbasierte Laufwerk automatisch auf ihre Festplatte ausgelagert wird, vielmehr wird der Speicherplatz danach aus einem Speicherpool angefordert, der bei Bedarf auf die Platte ausgelagert werden kann. Zusätzlich ist es möglich, malloc(9)-gestützte speicherbasierte Laufwerke zu erstellen. Das Anlegen solcher Laufwerke kann allerdings zu einer System-Panic führen, wenn der Kernel danach über zu wenig Speicher verfügt.

Beispiel 19-6. Erstellen eines speicherbasierten Laufwerks mit `mdconfig`

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
        with soft updates
super-block backups (for fsck -b #) at:
 160, 2752, 5344, 7936
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

Beispiel 19-7. Erstellen eines speicherbasierten Laufwerks mit `mdmfs`

```
# mdmfs -s 5m md2 /mnt
# df /mnt

# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md2        4846    2  4458     0%    /mnt
```

19.13.3. Virtuelle Laufwerke freigeben

Wenn ein virtuelles Laufwerk nicht mehr gebraucht wird, sollten Sie dem System die belegten Ressourcen zurückgeben. Hängen Sie dazu zuerst das Dateisystem ab und geben Sie dann die benutzten Ressourcen mit mdconfig(8) frei.

Alle von `/dev/md4` belegten Ressourcen werden mit dem nachstehenden Kommando freigegeben:

```
# mdconfig -d -u 4
```

Eingerichtete md(4)-Geräte werden mit dem Befehl `mdconfig -l` angezeigt.

19.14. Schnappschüsse von Dateisystemen

Beigetragen von Tom Rhodes.

Zusammen mit Soft Updates bietet FreeBSD eine neue Funktion: Schnappschüsse von Dateisystemen.

Schnappschüsse sind Dateien, die ein Abbild eines Dateisystems enthalten und müssen auf dem jeweiligen Dateisystem erstellt werden. Pro Dateisystem darf es maximal 20 Schnappschüsse, die im Superblock vermerkt werden, geben. Schnappschüsse bleiben erhalten, wenn das Dateisystem abgehangen, neu eingehangen oder das System neu gestartet wird. Wenn Sie einen Schnappschuss nicht mehr benötigen, können Sie ihn mit rm(1) löschen. Es ist egal, in welcher Reihenfolge Schnappschüsse gelöscht werden. Es kann allerdings vorkommen, dass nicht der gesamte Speicherplatz wieder freigegeben wird, da ein anderer Schnappschuss einen Teil der entfernten Blöcke für sich beanspruchen kann.

Das unveränderliche `Snapshot`-Dateiflag wird nach der Erstellung des Snaphshots von mksnap_ffs(8) gesetzt. Durch die Verwendung von unlink(1) ist es allerdings möglich, einen Schnappschuss zu löschen.

Schnappschüsse werden mit mount(8) erstellt. Das folgende Kommando legt einen Schnappschuss von `/var` in `/var/snapshot/snap` ab:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Den Schnappschuss können Sie auch mit mksnap_ffs(8) erstellen:

```
# mksnap_ffs /var /var/snapshot/snap
```

Um einen Schnappschuss auf Ihrem System zu finden, verwenden Sie find(1):

```
# find /var -flags snapshot
```

Nachdem ein Schnappschuss erstellt wurde, können Sie ihn für verschiedene Zwecke benutzen:

- Sie können den Schnappschuss für die Datensicherung benutzen und ihn auf eine CD oder ein Band schreiben.
- Sie können den Schnappschuss mit fsck(8) manuell prüfen. Wenn das Dateisystem zum Zeitpunkt der Erstellung des Schnappschusses in Ordnung war, sollte fsck(8) immer erfolgreich durchlaufen. Der Hintergrund-Prozess fsck(8) hat im Übrigen genau diese Aufgabe.
- Sie können den Schnappschuss mit dump(8) sichern. Sie erhalten dann eine konsistente Sicherung des Dateisystems zu dem Zeitpunkt, der durch den Zeitstempel des Schnappschusses gegeben ist. Der Schalter `-L` von dump(8) erstellt für die Sicherung einen Schnappschuss und entfernt diesen am Ende der Sicherung wieder.
- Sie können einen Schnappschuss in den Verzeichnisbaum einhängen und sich dann den Zustand des Dateisystems zu dem Zeitpunkt ansehen, an dem der Schnappschuss erstellt wurde. Der folgende Befehl hängt den Schnappschuss `/var/snapshot/snap` ein:

  ```
  # mdconfig -a -t vnode -f /var/snapshot/snap -u 4
  # mount -r /dev/md4 /mnt
  ```

Sie können sich nun den eingefrorenen Stand des `/var` Dateisystems unterhalb von `/mnt` ansehen. Mit Ausnahme der früheren Schnappschüsse, die als leere Dateien auftauchen, wird zu Beginn alles so aussehen, wie zum Zeitpunkt der Erstellung des Schnappschusses. Wenn Sie den Schnappschuss nicht mehr benötigen, können Sie ihn, wie nachfolgend gezeigt, abhängen:

```
# umount /mnt
# mdconfig -d -u 4
```

Weitere Informationen über Soft Updates und Schnappschüsse von Dateisystemen sowie technische Artikel finden Sie auf der Webseite von Marshall Kirk McKusick (http://www.mckusick.com/).

19.15. Dateisystem-Quotas

Quotas sind eine optionale Funktion des Betriebssystems, die es Ihnen erlauben, den Plattenplatz und/oder die Anzahl der Dateien eines Benutzers oder der Mitglieder einer Gruppe, auf Dateisystemebene zu beschränken. Oft wird dies auf Timesharing-Systemen (Mehrbenutzersystemen) genutzt, da es dort erwünscht ist, die Ressourcen, die ein Benutzer oder eine Gruppe von Benutzern belegen können, zu limitieren. Das verhindert, dass ein Benutzer oder eine Gruppe von Benutzern den ganzen verfügbaren Plattenplatz belegt.

19.15.1. Konfiguration des Systems, um Quotas zu aktivieren

Bevor Quotas benutzt werden können, müssen sie im Kernel konfiguriert werden, wozu die folgende Zeile der Kernelkonfiguration hinzugefügt wird:

```
options QUOTA
```

Im gewöhnlichen `GENERIC` Kernel sind Quotas nicht aktiviert, so dass Sie einen angepassten Kernel konfigurieren und bauen müssen, um Quotas zu benutzen. Weitere Informationen finden Sie in Kapitel 9.

Durch Hinzufügen der folgenden Zeile in `/etc/rc.conf` wird das Quota-System in FreeBSD 7.X und ältere aktiviert:

```
enable_quotas="YES"
```

Seit FreeBSD 8.0-RELEASE und dessen Nachfolger fügen Sie stattdessen die folgende Zeile hinzu:

```
quota_enable="YES"
```

Um den Start des Quota-Systems zu beeinflussen, steht eine weitere Variable zur Verfügung. Normalerweise wird beim Booten die Integrität der Quotas auf allen Dateisystemen mit quotacheck(8) überprüft. quotacheck(8) stellt sicher, dass die Quota-Datenbank mit den Daten auf einem Dateisystem übereinstimmt. Dies ist allerdings ein sehr zeitraubender Prozess, der die Zeit, die das System zum Booten braucht, signifikant beeinflusst. Eine Variable in `/etc/rc.config` erlaubt es Ihnen, diesen Schritt zu überspringen:

```
check_quotas="NO"
```

Schließlich müssen Sie noch in `/etc/fstab` die Plattenquotas auf Dateisystemebene aktivieren. Dort können Sie für alle Dateisysteme Quotas für Benutzer, Gruppen oder für beide aktivieren.

Um Quotas pro Benutzer für ein Dateisystem zu aktivieren, geben Sie für dieses Dateisystem die Option `userquota` im Feld Optionen von `/etc/fstab` an. Beispiel:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Um Quotas für Gruppen einzurichten, verwenden Sie `groupquota` anstelle von `userquota`. Um Quotas für Benutzer und Gruppen einzurichten, ändern Sie den Eintrag wie folgt ab:

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

Die Quotas werden jeweils im Rootverzeichnis des Dateisystems unter dem Namen `quota.user` für Benutzer-Quotas und `quota.group` für Gruppen-Quotas abgelegt. Obwohl fstab(5) beschreibt, dass diese Dateien an anderer Stelle gespeichert werden können, wird das nicht empfohlen, da es den Anschein hat, dass die verschiedenen Quota-Utilities das nicht richtig unterstützen.

Jetzt sollten Sie Ihr System mit dem neuen Kernel booten. `/etc/rc` wird dann automatisch die richtigen Kommandos aufrufen, die die Quota-Dateien für alle Quotas, die Sie in `/etc/fstab` definiert haben, anlegen. Deshalb müssen vorher auch keine leeren Quota-Dateien angelegt werden.

Normalerweise brauchen Sie die Kommandos quotacheck(8), quotaon(8) oder quotaoff(8) nicht händisch aufzurufen, obwohl Sie vielleicht die entsprechenden Seiten im Manual lesen sollten, um sich mit ihnen vertraut zu machen.

19.15.2. Setzen von Quota-Limits

Nachdem Sie Quotas in Ihrem System aktiviert haben, sollten Sie überprüfen, dass Sie auch tatsächlich aktiviert sind. Führen Sie dazu einfach den folgenden Befehl aus:

```
# quota -v
```

Für jedes Dateisystem, auf dem Quotas aktiviert sind, sollten Sie eine Zeile mit der Plattenauslastung und den aktuellen Quota-Limits sehen.

Mit edquota(8) können Sie nun Quota-Limits setzen.

Sie haben mehrere Möglichkeiten, die Limits für den Plattenplatz, den ein Benutzer oder eine Gruppe verbrauchen kann, oder die Anzahl der Dateien, die angelegt werden dürfen, festzulegen. Die Limits können auf dem Plattenplatz (Block-Quotas) oder der Anzahl der Dateien (Inode-Quotas) oder einer Kombination von beiden basieren. Jedes dieser Limits wird weiterhin in zwei Kategorien geteilt: Hardlimits und Softlimits.

Ein Hardlimit kann nicht überschritten werden. Hat der Benutzer einmal ein Hardlimit erreicht, so kann er auf dem betreffenden Dateisystem keinen weiteren Platz mehr beanspruchen. Hat ein Benutzer beispielsweise ein Hardlimit von 500 Kilobytes auf einem Dateisystem und benutzt davon 490 Kilobyte, so kann er nur noch 10 weitere Kilobytes beanspruchen. Der Versuch, weitere 11 Kilobytes zu beanspruchen, wird fehlschlagen.

Im Gegensatz dazu können Softlimits für eine befristete Zeit überschritten werden. Diese Frist beträgt in der Grundeinstellung eine Woche. Hat der Benutzer das Softlimit über die Frist hinaus überschritten, so wird das Softlimit in ein Hardlimit umgewandelt und der Benutzer kann keinen weiteren Platz mehr beanspruchen. Wenn er einmal das Softlimit unterschreitet, wird die Frist wieder zurückgesetzt.

Das folgende Beispiel zeigt die Benutzung von edquota(8). Wenn edquota(8) aufgerufen wird, wird der Editor gestartet, der durch `EDITOR` gegeben ist oder **vi** falls `EDITOR` nicht gesetzt ist. In dem Editor können Sie die Limits eingeben.

```
# edquota -u test

Quotas for user test:
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
        inodes in use: 0, limits (soft = 50, hard = 60)
```

Für jedes Dateisystem, auf dem Quotas aktiv sind, sehen Sie zwei Zeilen, eine für die Block-Quotas und die andere für die Inode-Quotas. Um ein Limit zu modifizieren, ändern Sie einfach den angezeigten Wert. Um beispielsweise

das Blocklimit dieses Benutzers von einem Softlimit von 50 und einem Hardlimit von 75 auf ein Softlimit von 500 und ein Hardlimit von 600 zu erhöhen, ändern Sie die Zeile

```
/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
```

zu:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

Die neuen Limits sind wirksam, wenn Sie den Editor verlassen.

Manchmal ist es erwünscht, die Limits für einen Bereich von UIDs zu setzen. Dies kann mit der `-p` Option von edquota(8) bewerkstelligt werden. Weisen Sie dazu die Limits einem Benutzer zu und rufen danach `edquota -p protouser startuid-enduid` auf. Besitzt beispielsweise der Benutzer `test` die gewünschten Limits, können diese mit dem folgenden Kommando für die UIDs 10.000 bis 19.999 dupliziert werden:

```
# edquota -p test 10000-19999
```

Weitere Informationen erhalten Sie in edquota(8).

19.15.3. Überprüfen von Quota-Limits und Plattennutzung

Sie können quota(1) oder repquota(8) benutzen, um Quota-Limits und Plattennutzung zu überprüfen. Um die Limits oder die Plattennutzung individueller Benutzer und Gruppen zu überprüfen, kann quota(1) benutzt werden. Ein Benutzer kann nur die eigenen Quotas und die Quotas der Gruppe, der er angehört untersuchen. Nur der Superuser darf sich alle Limits ansehen. Mit repquota(8) erhalten Sie eine Zusammenfassung von allen Limits und der Plattenausnutzung für alle Dateisysteme, auf denen Quotas aktiv sind.

Das folgende Beispiel zeigt die Ausgabe von `quota -v` für einen Benutzer, der Quota-Limits auf zwei Dateisystemen besitzt:

```
Disk quotas for user test (uid 1002):
     Filesystem  usage   quota   limit   grace   files   quota   limit   grace
           /usr      65*     50      75   5days       7      50      60
       /usr/var       0      50      75               0      50      60
```

Im Dateisystem `/usr` liegt der Benutzer momentan 15 Kilobytes über dem Softlimit von 50 Kilobytes und hat noch 5 Tage seiner Frist übrig. Der Stern `*` zeigt an, dass der Benutzer sein Limit überschritten hat.

In der Ausgabe von quota(1) werden Dateisysteme, auf denen ein Benutzer keinen Platz verbraucht, nicht angezeigt, auch wenn diesem Quotas zugewiesen wurden. Mit `-v` werden diese Dateisysteme, wie `/usr/var` im obigen Beispiel, angezeigt.

19.15.4. Quotas über NFS

Quotas werden von dem Quota-Subsystem auf dem NFS Server erzwungen. Der rpc.rquotad(8) Dæmon stellt quota(1) die Quota Informationen auf dem NFS Client zur Verfügung, so dass Benutzer auf diesen Systemen ihre Quotas abfragen können.

Aktivieren Sie `rpc.rquotad` in `/etc/inetd.conf` wie folgt:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Anschließend starten Sie `inetd` neu:

```
# /etc/rc.d/inetd restart
```

19.16. Partitionen verschlüsseln

Beigetragen von Lucky Green.

FreeBSD bietet ausgezeichnete Möglichkeiten, Daten vor unberechtigten Zugriffen zu schützen. Wenn das Betriebssystem läuft, schützen Zugriffsrechte und vorgeschriebene Zugriffskontrollen (MAC) (siehe Kapitel 17) die Daten. Die Zugriffskontrollen des Betriebssystems schützen allerdings nicht vor einem Angreifer, der Zugriff auf den Rechner hat. Der Angreifer kann eine Festplatte einfach in ein anderes System einbauen und dort die Daten analysieren.

Die für FreeBSD verfügbaren kryptografischen Subsysteme **GEOM Based Disk Encryption (gbde)** und `geli` sind in der Lage, Daten auf Dateisystemen auch vor hoch motivierten Angreifern zu schützen, die über erhebliche Mittel verfügen. Dieser Schutz ist unabhängig von der Art und Weise, durch die ein Angreifer Zugang zu einer Festplatte oder zu einem Rechner erlangt hat. Im Gegensatz zu schwerfälligen Systemen, die einzelne Dateien verschlüsseln, verschlüsseln **gbde** und `geli` transparent ganze Dateisysteme. Auf der Festplatte werden dabei keine Daten im Klartext gespeichert.

19.16.1. Plattenverschlüsselung mit gbde

1. Wechseln sie zu `root`

   Sie benötigen Superuser-Rechte, um **gbde** einzurichten.

   ```
   % su -
   Password:
   ```

2. Aktivieren Sie gbde(4) in der Kernelkonfigurationsdatei

   Fügen Sie folgende Zeile in Ihre Kernelkonfigurationsdatei ein:

   ```
   options GEOM_BDE
   ```

   Übersetzen und installieren Sie den FreeBSD-Kernel wie in Kapitel 9 beschrieben.

   Starten sie das System neu, um den neuen Kernel zu benutzen.

3. Alternativ zur Neukompilierung des Kernels können Sie auch `kldload` verwenden, um das Kernelmodul gbde(4) zu laden:

   ```
   # kldload geom_bde
   ```

19.16.1.1. Einrichten eines verschlüsselten Dateisystems

Das folgende Beispiel beschreibt, wie ein Dateisystem auf einer neuen Festplatte verschlüsselt wird. Das Dateisystem wird in `/private` eingehangen. Mit **gbde** könnten auch `/home` und `/var/mail` verschlüsselt werden. Die dazu nötigen Schritte können allerdings in dieser Einführung nicht behandelt werden.

1. Installieren der Festplatte

Installieren Sie die Festplatte wie in Abschnitt 19.3 beschrieben. Im Beispiel verwenden wir die Partition `/dev/ad4s1c`. Die Gerätedateien `/dev/ad0s1*` sind Standard-Partitionen des FreeBSD-Systems.

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4
```

2. Verzeichnis für gbde-Lock-Dateien anlegen

   ```
   # mkdir /etc/gbde
   ```

   Die Lock-Dateien sind für den Zugriff von **gbde** auf verschlüsselte Partitionen notwendig. Ohne die Lock-Dateien können die Daten nur mit erheblichem manuellen Aufwand wieder entschlüsselt werden (dies wird auch von der Software nicht unterstützt). Jede verschlüsselte Partition benötigt eine gesonderte Lock-Datei.

3. Vorbereiten der gbde-Partition

   Eine von **gbde** benutzte Partition muss einmalig vorbereitet werden:

   ```
   # gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
   ```

   gbde(8) öffnet eine Vorlage in Ihrem Editor, in der Sie verschiedene Optionen einstellen können. Setzen Sie `sector_size` auf `2048`, wenn Sie UFS1 oder UFS2 benutzen.

   ```
   # $FreeBSD: src/sbin/gbde/template.txt,v 1.1.36.1 2009/08/03 08:13:06 kensmith Exp $
   #
   # Sector size is the smallest unit of data which can be read or written.
   # Making it too small decreases performance and decreases available space.
   # Making it too large may prevent filesystems from working.  512 is the
   # minimum and always safe.  For UFS, use the fragment size
   #
   sector_size     =       2048
   [...]
   ```

   gbde(8) fragt dann zweimal eine Passphrase zum Schutz der Daten ab. Die Passphrase muss beides Mal gleich eingegeben werden. Die Sicherheit der Daten hängt alleine von der Qualität der gewählten Passphrase ab. [1]

   Mit `gbde init` wurde im Beispiel auch die Lock-Datei `/etc/gbde/ad4s1c.lock` angelegt. **gbde**-Lockdateien müssen die Dateiendung ".lock" aufweisen, damit sie von `/etc/rc.d/gbde`, dem Startskript von **gbde**, erkannt werden.

   > **Achtung:** Sichern Sie die Lock-Dateien von **gbde** immer zusammen mit den verschlüsselten Dateisystemen. Ein entschlossener Angreifer kann die Daten vielleicht auch ohne die Lock-Datei entschlüsseln. Ohne die Lock-Datei können Sie allerdings nicht auf die verschlüsselten Daten zugreifen. Dies ist nur noch mit erheblichem manuellen Aufwand möglich, der weder von gbde(8) noch seinem Entwickler unterstützt wird.

4. Einbinden der verschlüsselten Partition in den Kernel

   ```
   # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
   ```

   Das Kommando fragt die Passphrase ab, die Sie beim Vorbereiten der Partition eingegeben haben. Das neue Gerät erscheint danach als `/dev/device_name.bde` im Verzeichnis `/dev`:

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   ```

```
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Dateisystem auf dem verschlüsselten Gerät anlegen

   Wenn der Kernel die verschlüsselte Partition kennt, können Sie ein Dateisystem auf ihr anlegen. Benutzen Sie dazu den Befehl newfs(8). Da ein Dateisystem vom Typ UFS2 sehr viel schneller als eins vom Typ UFS1 angelegt wird, empfehlen wir Ihnen, die Option `-O2` zu benutzen.

   ```
   # newfs -U -O2 /dev/ad4s1c.bde
   ```

   > **Anmerkung:** newfs(8) muss auf einer dem Kernel bekannten **gbde**-Partition (einem Gerät mit dem Namen `*.bde` laufen.

6. Einhängen der verschlüsselten Partition

   Legen Sie einen Mountpunkt für das verschlüsselte Dateisystem an:

   ```
   # mkdir /private
   ```

   Hängen Sie das verschlüsselte Dateisystem ein:

   ```
   # mount /dev/ad4s1c.bde /private
   ```

7. Überprüfen des verschlüsselten Dateisystem

   Das verschlüsselte Dateisystem sollte jetzt von df(1) erkannt werden und benutzt werden können.

   ```
   % df -H
   Filesystem        Size   Used  Avail Capacity  Mounted on
   /dev/ad0s1a      1037M    72M   883M     8%    /
   /devfs            1.0K   1.0K     0B   100%    /dev
   /dev/ad0s1f       8.1G    55K   7.5G     0%    /home
   /dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
   /dev/ad0s1d       6.1G   1.9G   3.7G    35%    /usr
   /dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
   ```

19.16.1.2. Einhängen eines existierenden verschlüsselten Dateisystems

Nach jedem Neustart müssen verschlüsselte Dateisysteme dem Kernel wieder bekannt gemacht werden, auf Fehler überprüft werden und eingehangen werden. Die dazu nötigen Befehle müssen als `root` durchgeführt werden.

1. gbde-Partition im Kernel bekannt geben

   ```
   # gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
   ```

   Das Kommando fragt nach der Passphrase, die Sie beim Vorbereiten der verschlüsselten **gbde**-Partition festgelegt haben.

2. Prüfen des Dateisystems

   Das verschlüsselte Dateisystem kann noch nicht automatisch über `/etc/fstab` eingehangen werden. Daher muss es vor dem Einhängen mit fsck(8) geprüft werden:

   ```
   # fsck -p -t ffs /dev/ad4s1c.bde
   ```

3. Einhängen des verschlüsselten Dateisystems

```
# mount /dev/ad4s1c.bde /private
```

Das verschlüsselte Dateisystem steht danach zur Verfügung.

19.16.1.2.1. Verschlüsselte Dateisysteme automatisch einhängen

Mit einem Skript können verschlüsselte Dateisysteme automatisch bekannt gegeben, geprüft und eingehangen werden. Wir raten Ihnen allerdings aus Sicherheitsgründen davon ab. Starten Sie das Skript manuell an der Konsole oder in einer ssh(1)-Sitzung.

Zu diesem Zweck existiert ein `rc.d`-Skript, an das über Einträge in der Datei rc.conf(5) Argumente übergeben werden können. Dazu ein Beispiel:

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

Durch diese Argumente muss beim Systemstart die **gbde**-Passphrase eingegeben werden. Erst nach Eingabe der korrekten Passphrase wird die **gbde**-verschlüsselte Partition automatisch in den Verzeichnisbaum eingehängt. Dieses Vorgehen ist insbesondere dann nützlich, wenn Sie **gbde** auf einem Notebook einsetzen wollen.

19.16.1.3. Kryptografische Methoden von gbde

gbde(8) benutzt den 128-Bit AES im CBC-Modus, um die Daten eines Sektors zu verschlüsseln. Jeder Sektor einer Festplatte wird mit einem unterschiedlichen AES-Schlüssel verschlüsselt. Mehr Informationen, unter anderem wie die Schlüssel für einen Sektor aus der gegebenen Passphrase ermittelt werden, erhalten Sie in gbde(4).

19.16.1.4. Kompatibilität

sysinstall(8) kann nicht mit verschlüsselten **gbde**-Geräten umgehen. Vor dem Start von sysinstall(8) sind alle `*.bde`-Geräte zu deaktivieren, da sysinstall(8) sonst bei der Gerätesuche abstürzt. Das im Beispiel verwendete Gerät wird mit dem folgenden Befehl deaktiviert:

```
# gbde detach /dev/ad4s1c
```

> **Anmerkung:** Sie können **gbde** nicht zusammen mit **vinum** benutzen, da vinum(4) das geom(4)-Subsystem nicht benutzt.

19.16.2. Plattenverschlüsselung mit `geli`

Beigetragen von Daniel Gerzo.

`geli` ist als alternative kryptografische GEOM-Klasse verfügbar und wird derzeit von Pawel Jakub Dawidek weiterentwickelt. `geli` unterscheidet sich von `gbde` durch unterschiedliche Fähigkeiten und einen unterschiedlichen Ansatz für die Verschlüsselung von Festplatten.

Die wichtigsten Merkmale von geli(8) sind:

- Der Einsatz des crypto(9)-Frameworks – verfügt das System über kryptografische Hardware, wird diese von `geli` automatisch verwendet.
- Die Unterstützung verschiedener kryptografischer Algorithmen (derzeit AES, Blowfish, sowie 3DES).
- Die Möglichkeit, die root-Partition zu verschlüsseln. Um auf die verschlüsselte root-Partition zugreifen zu können, muss beim Systemstart die Passphrase eingegeben werden.
- `geli` erlaubt den Einsatz von zwei voneinander unabhängigen Schlüsseln (etwa einem privaten “Schlüssel” und einem “Unternehmens-Schlüssel”).
- `geli` ist durch einfache Sektor-zu-Sektor-Verschlüsselung sehr schnell.
- Die Möglichkeit, Master-Keys zu sichern und wiederherzustellen. Wenn ein Benutzer seinen Schlüssel zerstört, kann er über seinen zuvor gesicherten Schlüssel wieder auf seine Daten zugreifen.
- `geli` erlaubt es, Platten mit einem zufälligen Einmal-Schlüssel einzusetzen, was insbesondere für Swap-Partitionen und temporäre Dateisysteme interessant ist.

Weitere Informationen zu den Fähigkeiten von `geli` finden Sie in geli(8).

Die folgenden Schritte beschreiben, wie Sie `geli` im FreeBSD-Kernel aktivieren und einen `geli`-Verschlüsselungs-Provider anlegen können.

Da Sie Ihren Kernel anpassen müssen, benötigen Sie außerdem `root`-Privilegien.

1. Aufnahme der `geli`-Unterstützung in Ihre Kernelkonfigurationsdatei

   Fügen Sie die folgenden Zeilen in Ihre Kernelkonfigurationsdatei ein:

   ```
   options GEOM_ELI
   device crypto
   ```

   Bauen und installieren Sie Ihren neuen Kernel wie in Kapitel 9 beschrieben.

   Alternativ können Sie aber auch das `geli`-Kernelmodul beim Systemstart laden. Dazu fügen Sie die folgende Zeile in `/boot/loader.conf` ein:

   ```
   geom_eli_load="YES"
   ```

   Ab sofort wird geli(8) vom Kernel unterstützt.

2. Erzeugen des Master-Keys

   Das folgende Beispiel beschreibt, wie Sie eine Schlüsseldatei erzeugen, die als Teil des Master-Keys für den Verschlüsselungs-Provider verwendet wird, der unter `/private` in den Verzeichnisbaum eingehängt (“gemountet”) wird. Diese Schlüsseldatei liefert zufällige Daten, die für die Verschlüsselung des Master-Keys benötigt werden. Zusätzlich wird der Master-Key durch eine Passphrase geschützt. Die Sektorgröße des Providers beträgt 4 KB. Außerdem wird beschrieben, wie Sie einen `geli`-Provider aktivieren, ein vom ihm verwaltetes Dateisystem erzeugen, es mounten, mit ihm arbeiten und wie Sie es schließlich wieder unmounten und den Provider deaktivieren.

   Um eine bessere Leistung zu erzielen, sollten Sie eine größere Sektorgröße (beispielsweise 4 KB) verwenden.

   Der Master-Key wird durch eine Passphrase sowie die Daten der Schlüsseldatei (die von `/dev/random` stammen) geschützt. Die Sektorgröße von `/dev/da2.eli` (das als Provider bezeichnet wird) beträgt 4 KB.

   ```
   # dd if=/dev/random of=/root/da2.key bs=64 count=1
   ```

```
# geli init -s 4096 -K /root/da2.key /dev/da2
Enter new passphrase:
Reenter new passphrase:
```

Es ist nicht zwingend nötig, sowohl eine Passphrase als auch eine Schlüsseldatei zu verwenden. Die einzelnen Methoden können auch unabhängig voneinander eingesetzt werden.

Wird für die Schlüsseldatei der Wert "-" angegeben, wird dafür die Standardeingabe verwendet. Das folgende Beispiel zeigt, dass Sie auch mehr als eine Schlüsseldatei verwenden können.

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

3. Aktivieren des Providers mit dem erzeugten Schlüssel

```
# geli attach -k /root/da2.key /dev/da2
Enter passphrase:
```

Dadurch wird die (Normaltext-)Gerätedatei `/dev/da2.eli` angelegt.

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Das neue Dateisystem erzeugen

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

Das verschlüsselte Dateisystem wird nun von df(1) angezeigt und kann ab sofort eingesetzt werden.

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
/dev/da2.eli   150G   4.1K   138G     0%    /private
```

5. Das Dateisystem unmounten und den Provider deaktivieren

Wenn Sie nicht mehr mit dem verschlüsselten Dateisystem arbeiten und die unter `/private` eingehängte Partition daher nicht mehr benötigen, sollten Sie diese unmounten und den `geli`-Verschlüsselungs-Provider wieder deaktivieren.

```
# umount /private
# geli detach da2.eli
```

Weitere Informationen zum Einsatz von `geli` finden Sie in geli(8).

19.16.2.1. Der Einsatz des `geli`- `rc.d`-Skripts

`geli` verfügt über ein `rc.d`-Skript, das den Einsatz von `geli` deutlich vereinfacht. Es folgt nun ein Beispiel, in dem `geli` über die Datei rc.conf(5) konfiguriert wird:

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
```

Durch diese Einträge wird `/dev/da2` als `geli`-Provider festgelegt. Der Master-Key befindet sich in `/root/da2.key`. Beim Aktivieren des `geli`-Providers wird keine Passphrase abgefragt (beachten Sie, dass dies nur

dann möglich ist, wenn Sie `geli` mit dem Parameter `-P` initialisieren). Wird das System heruntergefahren, wird der `geli`-Provider zuvor deaktiviert.

Weitere Informationen zur Konfiguration der `rc.d`-Skripten finden Sie im Abschnitt rc.d des Handbuchs.

19.17. Den Auslagerungsspeicher verschlüsseln

Geschrieben von Christian Brüffer.

Die Verschlüsselung des Auslagerungsspeichers ist unter FreeBSD einfach einzurichten. Je nach dem, welche FreeBSD-Version Sie einsetzen, können Konfiguration und mögliche Optionen allerdings unterschiedlich sein. Sie können entweder das gbde(8)- oder das geli(8)-Verschlüsselungs-Subsystem einsetzen. Beide Subsysteme werden über das rc.d-Skript `encswap` gestartet.

Der letzte Abschnitt, Partitionen verschlüsseln, enthält eine kurze Beschreibung der verschiedenen Verschlüsselungs-Subsysteme.

19.17.1. Warum sollte der Auslagerungsspeicher verschlüsselt werden?

Wie die Verschlüsselung von Plattenpartitionen dient auch die Verschlüsselung des Auslagerungsspeichers dem Schutz sensitiver Informationen. Stellen Sie sich etwa eine Anwendung vor, die ein Passwort erfordert. Solange dieses Passwort im Hauptspeicher verbleibt, ist alles in Ordnung. Beginnt Ihr Betriebssystem allerdings, Daten auf die Festplatte auszulagern, um im Hauptspeicher Platz für andere Anwendungen zu schaffen, kann es passieren, dass Ihr Passwort im Klartext in den Auslagerungsspeicher geschrieben wird, was es einem potentiellen Angreifer leicht macht, Ihr Passwort herauszufinden. Die Verschlüsselung Ihres Auslagerungsspeichers kann dieses Problem lösen.

19.17.2. Vorbereitungen

Anmerkung: Für die weiteren Ausführungen dieses Abschnitts stellt `ad0s1b` die Swap-Partition dar.

Noch ist Ihr Auslagerungsspeicher nicht verschlüsselt. Es könnte allerdings sein, dass bereits Passwörter oder andere sensitive Daten als Klartext im Auslagerungsspeicher vorhanden sind. Daher sollten Sie den Auslagerungsspeicher komplett mit zufällig generierten Zeichen überschreiben, bevor Sie ihn verschlüsseln:

```
# dd if=/dev/random of=/dev/ad0s1b bs=1m
```

19.17.3. Den Auslagerungsspeicher mit gbde(8) verschlüsseln

In der Datei `/etc/fstab` sollte das Suffix `.bde` an den Gerätenamen der Swap-Partition anhängt werden:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.bde         none            swap    sw              0       0
```

19.17.4. Den Auslagerungsspeicher mit geli(8) verschlüsseln

Alternativ können Sie Ihren Auslagerungsspeicher auch mit geli(8) verschlüsseln. Die Vorgehensweise ist dabei ähnlich. Allerdings hängen Sie bei der Verwendung von geli(8) in `/etc/fstab` das Suffix `.eli` an den Gerätenamen der Swap-Partition an:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ad0s1b.eli         none            swap    sw              0       0
```

In der Voreinstellung verschlüsselt geli(8) den Auslagerungsspeicher mit dem AES-Algorithmus und einer Schlüssellänge von 128 Bit.

Es ist möglich, diese Optionen durch das Setzen der `geli_swap_flags`-Option in `/etc/rc.conf` anzupassen. Die folgende Zeile weist das rc.d-Skript `encswap` an, geli(8)-Swap-Partitionen mit dem Blowfish-Algorithmus und einer Schlüssellänge von 128 Bit zu verschlüsseln. Zusätzlich wird die Sektorgröße auf 4 Kilobyte gesetzt und die Option “detach on last close” aktiviert:

```
geli_swap_flags="-e blowfish -l 128 -s 4096 -d"
```

Eine Auflistung möglicher Optionen für den Befehl `onetime` finden Sie in der Manualpage zu geli(8).

19.17.5. Die korrekte Funktion testen

Nachdem Sie Ihr System neu gestartet haben, können Sie die korrekte Funktion Ihres verschlüsselten Auslagerungsspeichers prüfen, indem Sie sich die Ausgabe von `swapinfo` ansehen.

Wenn Sie gbde(8) einsetzen, erhalten Sie eine Meldung ähnlich der folgenden:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.bde    542720        0   542720     0%
```

Wenn Sie geli(8) einsetzen, erhalten Sie hingegen eine Ausgabe ähnlich der folgenden:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ad0s1b.eli    542720        0   542720     0%
```

19.18. Highly Available Storage (HAST)

Beigetragen von Daniel Gerzo. Mit Beiträgen von Freddie Cash, Pawel Jakub Dawidek, Michael W. Lucas und Viktor Petersson. Übersetzt von Benedict Reuschling.

19.18.1. Überblick

Hochverfügbarkeit ist eine der Hauptanforderungen von ernsthaften Geschäftsanwendungen und hochverfügbarer Speicher ist eine Schlüsselkomponente in solchen Umgebungen. Highly Available STorage, oder HAST, wurde von Pawel Jakub Dawidek als ein Framework entwickelt, welches die transparente Speicherung der gleichen Daten über mehrere physikalisch getrennte Maschinen ermöglicht, die über ein TCP/IP-Netzwerk verbunden sind. HAST kann

als ein netzbasiertes RAID1 (Spiegel) verstanden werden und ist dem DRBD®-Speichersystem der GNU/Linux-Plattform ähnlich. In Kombination mit anderen Hochverfügbarkeitseigenschaften von FreeBSD wie CARP, ermöglicht es HAST, hochverfügbare Speichercluster zu bauen, die in der Lage sind, Hardwareausfällen zu widerstehen.

Nachdem Sie diesen Abschnitt gelesen haben, werden Sie folgendes wissen:

- Was HAST ist, wie es funktioniert und welche Eigenschaften es besitzt.
- Wie man HAST auf FreeBSD aufsetzt und verwendet.
- Wie man CARP und devd(8) kombiniert, um ein robustes Speichersystem zu bauen.

Bevor Sie diesen Abschnitt lesen, sollten Sie:

- die Grundlagen von UNIX und FreeBSD verstanden haben (Kapitel 4).
- wissen, wie man Netzwerkschnittstellen und andere Kernsysteme von FreeBSD konfiguriert (Kapitel 12).
- ein gutes Verständnis der FreeBSD-Netzwerkfunktionalität besitzen (Teil IV in dem *Das FreeBSD-Handbuch*).
- FreeBSD 8.1-RELEASE oder höher einsetzen.

Das HAST-Projekt wurde von der FreeBSD Foundation mit Unterstützung der OMCnet Internet Service GmbH (http://www.omc.net/) und TransIP BV (http://www.transip.nl/) gesponsert.

19.18.2. HAST-Merkmale

Die Hauptmerkmale des HAST-Systems sind:

- Es kann zur Maskierung von I/O-Fehlern auf lokalen Festplatten eingesetzt werden.
- Dateisystem-unabhängig, was es erlaubt, jedes von FreeBSD unterstützte Dateisystem zu verwenden.
- Effiziente und schnelle Resynchronisation: es werden nur die Blöcke synchronisiert, die während der Ausfallzeit eines Knotens geändert wurden.
- Es kann in einer bereits bestehenden Umgebung eingesetzt werden, um zusätzliche Redundanz zu erreichen.
- Zusammen mit CARP, **Heartbeat**, oder anderen Werkzeugen, ist es möglich, ein robustes und dauerhaftes Speichersystem zu bauen.

19.18.3. HAST im Einsatz

HAST stellt auf Block-Ebene eine synchrone Replikation eines beliebigen Speichermediums auf mehreren Maschinen zur Verfügung. Daher werden mindestens zwei Knoten (physikalische Maschinen) benötigt: der `primary` (auch bekannt als `master`) Knoten, sowie der `secondary` (`slave`) Knoten. Diese beiden Maschinen zusammen werden als Cluster bezeichnet.

> **Anmerkung:** HAST ist momentan auf insgesamt zwei Knoten im Cluster beschränkt.

Da HAST in einer primär-sekundär-Konfiguration funktioniert, ist immer nur ein Knoten des Clusters zu jeder Zeit aktiv. Der `primäre` Knoten, auch `active` genannt, ist derjenige, der alle I/O-Anfragen verarbeitet, die an die HAST-Schnittstelle gesendet werden. Der `secondary`-Knoten wird automatisch vom `primary`-Knoten aus synchronisiert.

Die physischen Komponenten des HAST-Systems sind:

- lokale Platte (am Primärknoten)
- Platte am entfernten Rechner (Sekundärknoten)

HAST arbeitet synchron auf Blockebene, was es für Dateisysteme und Anwendungen transparent macht. HAST stellt gewöhnliche GEOM-Provider im Verzeichnis `/dev/hast/` für die Verwendung durch andere Werkzeuge oder Anwendungen zur Verfügung, somit gibt es keinen Unterschied zwischen dem Einsatz von durch HAST bereitgestellten Geräten und herkömmlichen Platten, Partitionen, etc.

Jede Schreib-, Lösch- oder Entleerungsoperation wird an die lokale und über TCP/IP zu der entfernt liegenden Platte gesendet. Jede Leseoperation wird von der lokalen Platte durchgeführt, es sei denn, die lokale Platte ist nicht aktuell oder es tritt ein I/O-Fehler auf. In solchen Fällen wird die Leseoperation an den Sekundärknoten geschickt.

19.18.3.1. Synchronisation und Replikationsmodi

HAST versucht, eine schnelle Fehlerbereinigung zu gewährleisten. Aus diesem Grund ist es sehr wichtig, die Synchronisationszeit nach dem Ausfall eines Knotens zu reduzieren. Um eine schnelle Synchronisation zu ermöglichen, verwaltet HAST eine Bitmap von unsauberen Bereichen auf der Platte und synchronisiert nur diese während einer regulären Synchronisation (mit Ausnahme der initialen Synchronisation).

Es gibt viele Wege, diese Synchronisation zu behandeln. HAST implementiert mehrere Replikationsarten, um unterschiedliche Methoden der Synchronisation zu realisieren:

- *memsync*: meldet Schreiboperationen als vollständig, wenn die lokale Schreiboperation beendet ist und der entfernt liegende Knoten die Ankunft der Daten bestätigt hat, jedoch bevor die Daten wirklich gespeichert wurden. Die Daten werden auf dem entfernt liegenden Knoten direkt nach dem Senden der Bestätigung gespeichert. Dieser Modus ist dafür gedacht, Latenzen zu verringern und zusätzlich eine gute Verlässlichkeit zu bieten. Der *memsync*-Replikationsmodus ist momentan noch nicht implementiert.
- *fullsync*: meldet Schreiboperationen als vollständig, wenn die lokale Schreiboperation beendet ist und die entfernte Schreiboperation ebenfalls abgeschlossen wurde. Dies ist der sicherste und zugleich der langsamste Replikationsmodus. Er stellt den momentanen Standardmodus dar.
- *async*: meldet Schreiboperationen als vollständig, wenn lokale Schreibvorgänge abgeschlossen wurden. Dies ist der schnellste und gefährlichste Replikationsmodus. Er sollte verwendet werden, wenn die Latenz zu einem entfernten Knoten bei einer Replikation zu hoch ist für andere Modi. Der *async*-Replikationsmodus ist zum gegenwärtigen Zeitpunkt nicht implementiert.

> **Warnung:** Momentan wird nur der *fullsync*-Replikationsmodus unterstützt.

19.18.4. HAST-Konfiguration

HAST benötigt `GEOM_GATE`-Unterstützung, um korrekt zu funktionieren. Der `GENERIC`-Kernel enthält jedoch `GEOM_GATE` *nicht* von vornherein, jedoch ist in der Standardinstallation von FreeBSD `geom_gate.ko` als ladbares Modul vorhanden. Stellen Sie bei Systemen, bei denen nur das Allernötigste vorhanden sein soll, sicher, dass dieses Modul zur Verfügung steht. Als Alternative lässt sich die `GEOM_GATE`-Unterstützung direkt in den Kernel statisch einbauen, indem Sie die folgende Zeile zu Ihrer Kernelkonfigurationsdatei hinzufügen:

```
options GEOM_GATE
```

Das HAST-Framework besteht aus Sicht des Betriebssystems aus mehreren Bestandteilen:

- Dem hastd(8)-Dienst, welcher für die Datensynchronisation verantwortlich ist,
- Dem hastctl(8) Management-Werkzeug,
- Der Konfigurationsdatei hast.conf(5).

Das folgende Beispiel beschreibt, wie man zwei Knoten als `master-slave` / `primary-secondary` mittels HAST konfiguriert, um Daten zwischen diesen beiden auszutauschen. Die Knoten werden als *`hasta`* mit der IP-Adresse *`172.16.0.1`* und *`hastb`* mit der IP-Adresse *`172.16.0.2`* bezeichnet. Beide Knoten besitzen eine dedizierte Festplatte `/dev/`*`ad6`* mit der gleichen Grösse für den HAST-Betrieb. Der HAST-Pool (manchmal auch Ressource genannt, z.B. der GEOM-Provider in `/dev/hast/`) wird als *`test`* bezeichnet.

Die Konfiguration von HAST wird in der Datei `/etc/hast.conf` vorgenommen. Diese Datei sollte auf beiden Knoten gleich sein. Die denkbar einfachste Konfiguration ist folgende:

```
resource test {
       on hasta {
               local /dev/ad6
               remote 172.16.0.2
       }
       on hastb {
               local /dev/ad6
               remote 172.16.0.1
       }
}
```

Schlagen Sie in der hast.conf(5)-Manualpage nach, wenn Sie an erweiterten Konfigurationsmöglichkeiten interessiert sind.

> **Tipp:** Es ist ebenfalls möglich, den Hostnamen in den `remote`-Anweisungen zu verwenden. Stellen Sie in solchen Fällen sicher, dass diese Rechner auch aufgelöst werden können, also in der Datei `/etc/hosts` aufgeführt sind, oder alternativ im lokalen DNS.

Da nun die Konfiguration auf beiden Rechnern vorhanden ist, sind Sie in der Lage, den HAST-Pool zu erstellen. Lassen Sie die folgenden Kommandos auf beiden Knoten ablaufen, um die initialen Metadaten auf die lokale Platte zu schreiben und starten Sie anschliessend den hastd(8)-Dienst:

```
# hastctl create test
# /etc/rc.d/hastd onestart
```

> **Anmerkung:** Es ist *nicht* möglich, GEOM-Provider mit einem bereits bestehenden Dateisystem zu verwenden (z.B. um einen bestehenden Speicher in einen von HAST verwalteten Pool zu konvertieren), weil diese Prozedur bestimmte Metadaten auf den Provider schreiben muss und dafür nicht genug freier Platz zur Verfügung stehen wird.

HAST ist nicht dafür verantwortlich, die Rolle (`primary` oder `secondary`) für den jeweiligen Knoten festzulegen. Die Rolle des Knotens muss vom Administrator oder einer anderen Software wie **Heartbeat** mittels des hastctl(8)-Werkzeugs festgelegt werden. Auf dem primären Knoten (*`hasta`*) geben Sie nun den folgenden Befehl ein:

```
# hastctl role primary test
```

Geben Sie nun, ähnlich wie zuvor, das folgende Kommando auf dem sekundären Knoten (*`hastb`*) ein:

```
# hastctl role secondary test
```

> **Achtung:** Es kann passieren, dass beide Knoten nicht in der Lage sind, miteinander zu kommunizieren und dadurch beide als primäre Knoten konfiguriert sind; die Konsequenz daraus wird als `split-brain` bezeichnet. Um diese Situation zu bereinigen, folgen Sie den Schritten, die in Abschnitt 19.18.5.2 beschrieben sind.

Es ist möglich das Ergebnis des hastctl(8)-Werkzeugs auf jedem Knoten zu überprüfen:

```
# hastctl status test
```

Der wichtigste Teil ist die `status`-Textzeile der Ausgabe, die auf jedem Knoten `complete` lauten sollte. Falls der Status als `degraded` zurückgemeldet wird, ist etwas schief gegangen. Zu diesem Zeitpunkt hat die Synchronisation zwischen den beiden Knoten bereits begonnen. Die Synchronisation ist beendet, wenn das Kommando `hastctl status` meldet, dass die `dirty`-Bereiche 0 Bytes betragen.

Der letzte Schritt ist, ein Dateisystem auf dem `/dev/hast/`*`test`* GEOM-Provider anzulegen und dieses ins System einzuhängen. Dies muss auf dem `primary`-Knoten durchgeführt werden (da `/dev/hast/`*`test`* nur auf dem `primary`-Knoten erscheint). Dies kann ein paar Minuten dauern, abhängig von der Grösse der Festplatte:

```
# newfs -U /dev/hast/test
# mkdir /hast/test
# mount /dev/hast/test /hast/test
```

Sobald das HAST-Framework richtig konfiguriert wurde, besteht der letzte Schritt nun darin, sicherzustellen, dass HAST während des Systemstarts automatisch gestartet wird. Die folgende Zeile sollte zur Datei `/etc/rc.conf` hinzugefügt werden:

```
hastd_enable="YES"
```

19.18.4.1. Failover-Konfiguration

Das Ziel dieses Beispiels ist, ein robustes Speichersystem zu bauen, welches Fehlern auf einem beliebigen Knoten widerstehen kann. Die Schlüsselaufgabe in diesem Szenario besteht darin, zu verhindern, dass der `primary`-Knoten des Clusters ausfällt. Sollte es dennoch passieren, ist der `secondary`-Knoten da, um nahtlos einzuspringen, das Dateisystem zu prüfen, einzuhängen und mit der Arbeit fortzufahren, ohne dass auch nur ein einzelnes Bit an Daten verloren ging.

Um diese Aufgabe zu bewerkstelligen, ist es nötig, eine weitere Eigenschaft zu nutzen, die unter FreeBSD verfügbar ist, welche ein automatisches Failover auf der IP-Schicht ermöglicht: CARP. CARP steht für Common Address Redundancy Protocol und erlaubt es mehreren Rechnern im gleichen Netzsegment, die gleiche IP-Adresse zu verwenden. Setzen Sie CARP auf beiden Knoten des Clusters anhand der Dokumentation in Abschnitt 32.13 auf. Nachdem dieser Schritt abgeschlossen ist, sollte jeder Knoten seine eigene `carp0`-Schnittstelle mit der geteilten IP-Adresse *`172.16.0.254`* besitzen. Selbstverständlich muss der primäre HAST-Knoten des Clusters der CARP-Masterknoten sein.

Der HAST-Pool, welcher im vorherigen Abschnitt erstellt wurde, ist nun bereit für den Export über das Netzwerk auf den anderen Rechner. Dies kann durch den Export über NFS, **Samba** etc. erreicht werden, indem die geteilte IP-Addresse *`172.16.0.254`* verwendet wird. Das einzige ungelöste Problem ist der automatische Failover, sollte der primäre Knoten einmal ausfallen.

Falls die CARP-Schnittstelle aktiviert oder deaktiviert wird, generiert das FreeBSD-Betriebssystem ein devd(8)-Ereignis, was es ermöglicht, Zustandsänderungen auf den CARP-Schnittstellen zu überwachen. Eine Zustandsänderung auf der CARP-Schnittstelle ist ein Indiz dafür, dass einer der Knoten gerade ausgefallen oder wieder verfügbar ist. In diesem Fall ist es möglich, ein Skript zu starten, welches den Failover automatisch durchführt.

Um diese Zustandsänderungen auf der CARP-Schnittstelle abzufangen, müssen die folgenden Zeilen in der Datei `/etc/devd.conf` auf jedem Knoten eingefügt werden:

```
notify 30 {
       match "system" "IFNET";
       match "subsystem" "carp0";
       match "type" "LINK_UP";
       action "/usr/local/sbin/carp-hast-switch master";
};

notify 30 {
       match "system" "IFNET";
       match "subsystem" "carp0";
       match "type" "LINK_DOWN";
       action "/usr/local/sbin/carp-hast-switch slave";
};
```

Um diese neue Konfiguration zu aktivieren, starten Sie devd(8) auf beiden Knoten neu, um die neue Konfiguration wirksam werden zu lassen:

```
# /etc/rc.d/devd restart
```

Für den Fall, dass die `carp0`-Schnittstelle aktiviert oder deaktiviert wird (sich also der Status der Schnittstelle ändert), erzeugt das System eine Meldung, was es dem devd(8)-Subsystem ermöglicht, ein beliebiges Skript zu starten, in diesem Fall also `/usr/local/sbin/carp-hast-switch`. Dies ist das Skript, dass den automatischen Failover durchführt. Für genauere Informationen zu der obigen devd(8)-Konfiguration, lesen Sie die devd.conf(5)-Manualpage.

Ein Beispiel für ein solches Skript könnte wie folgt aussehen:

```
#!/bin/sh

# Original script by Freddie Cash <fjwcash@gmail.com>
# Modified by Michael W. Lucas <mwlucas@BlackHelicopters.org>
```

```
# and Viktor Petersson <vpetersson@wireload.net>

# The names of the HAST resources, as listed in /etc/hast.conf
resources="test"

# delay in mounting HAST resource after becoming master
# make your best guess
delay=3

# logging
log="local0.debug"
name="carp-hast"

# end of user configurable stuff

case "$1" in
       master)
                logger -p $log -t $name "Switching to primary provider for ${resources}."
                sleep ${delay}

                # Wait for any "hastd secondary" processes to stop
                for disk in ${resources}; do
                        while $( pgrep -lf "hastd: ${disk} \(secondary\)" > /dev/null 2>&1 ); do
                                sleep 1
                        done

                        # Switch role for each disk
                        hastctl role primary ${disk}
                        if [ $? -ne 0 ]; then
                                logger -p $log -t $name "Unable to change role to primary for reso
                                exit 1
                        fi
                done

                # Wait for the /dev/hast/* devices to appear
                for disk in ${resources}; do
                        for I in $( jot 60 ); do
                                [ -c "/dev/hast/${disk}" ] && break
                                sleep 0.5
                        done

                        if [ ! -c "/dev/hast/${disk}" ]; then
                                logger -p $log -t $name "GEOM provider /dev/hast/${disk} did not a
                                exit 1
                        fi
                done

                logger -p $log -t $name "Role for HAST resources ${resources} switched to primary.


                logger -p $log -t $name "Mounting disks."
                for disk in ${resources}; do
                        mkdir -p /hast/${disk}
```

```
                        fsck -p -y -t ufs /dev/hast/${disk}
                        mount /dev/hast/${disk} /hast/${disk}
                done

        ;;

        slave)
                logger -p $log -t $name "Switching to secondary provider for ${resources}."

                # Switch roles for the HAST resources
                for disk in ${resources}; do
                        if ! mount | grep -q "^/dev/hast/${disk} on "
                        then
                        else
                                umount -f /hast/${disk}
                        fi
                        sleep $delay
                        hastctl role secondary ${disk} 2>&1
                        if [ $? -ne 0 ]; then
                                logger -p $log -t $name "Unable to switch role to secondary for res
                                exit 1
                        fi
                        logger -p $log -t $name "Role switched to secondary for resource ${disk}."
                done
        ;;
esac
```

Im Kern führt das Skript die folgenden Aktionen durch, sobald ein Knoten zum `master` / `primary` wird:

- Es ernennt den HAST-Pool als den primären für einen gegebenen Knoten.
- Es prüft das Dateisystem, dass auf dem HAST-Pool erstellt wurde.
- Es hängt die Pools an die richtige Stelle im System ein.

Wenn ein Knoten zum `backup` / `secondary` ernannt wird:

- Hängt es den HAST-Pool aus dem Dateisystem aus.
- Degradiert es den HAST-Pool zum sekundären.

> **Achtung:** Bitte beachten Sie, dass dieses Skript nur ein Beispiel für eine mögliche Lösung darstellt. Es behandelt nicht alle möglichen Szenarien, die auftreten können und sollte erweitert bzw. abgeändert werden, so dass z.B. benötigte Dienste gestartet oder gestoppt werden usw.

> **Tipp:** Für dieses Beispiel wurde ein Standard-UFS Dateisystem verwendet. Um die Zeit für die Wiederherstellung zu verringern, kann ein UFS mit Journal oder ein ZFS-Dateisystem benutzt werden.

Weitere detaillierte Informationen mit zusätzlichen Beispielen können auf der HAST Wiki (http://wiki.FreeBSD.org/HAST)-Seite abgerufen werden.

19.18.5. Fehlerbehebung

19.18.5.1. Allgemeine Tipps zur Fehlerbehebung

HAST sollte generell ohne Probleme funktionieren. Jedoch kann es, wie bei jeder anderen Software auch, zu gewissen Zeiten sein, dass sie sich nicht so verhält wie angegeben. Die Quelle dieser Probleme kann unterschiedlich sein, jedoch sollte als Faustregel gewährleistet werden, dass die Zeit für beide Knoten im Cluster synchron läuft.

Die Anzahl an Debugging-Meldungen von hastd(8) sollte erhöht werden, wenn Fehler von HAST bereinigt werden. Dies kann durch das Starten des hastd(8)-Dienstes mit der Option `-d` erreicht werden. Wichtig zu wissen ist, dass diese Option mehrfach angegeben werden kann, um die Anzahl an Meldungen weiter zu erhöhen. Sie können viele nützliche Informationen auf diese Art bekommen. Sie sollten ebenfalls die Verwendung der Option `-F` in Erwägung ziehen, die den hastd(8)-Dienst in den Vordergrund bringt.

19.18.5.2. Auflösung des Split-brain-Zustands

Die Konsequenz aus der Situation, wenn beide Knoten des Clusters nicht in der Lage sind, miteinander zu kommunizieren und dadurch beide als primäre Knoten fungieren, wird als `split-brain` bezeichnet. Dies ist ein gefährlicher Zustand, weil es beiden Knoten erlaubt ist, Änderungen an den Daten vorzunehmen, die miteinander nicht in Einklang gebracht werden können. Diese Situation sollte vom Systemadministrator händisch bereinigt werden.

Um diese Situation zu beheben, muss der Administrator entscheiden, welcher Knoten die wichtigsten Änderungen von beiden besitzt (oder diese manuell miteinander vermischen) und anschliessend den HAST-Knoten die volle Synchronisation mit jenem Knoten durchführen zu lassen, welcher die beschädigten Daten besitzt. Um dies zu tun, geben Sie die folgenden Befehle auf dem Knoten ein, der neu synchronisiert werden soll:

```
# hastctl role init <resource>
# hastctl create <resource>
# hastctl role secondary <resource>
```

Fußnoten

1. Die Auswahl einer sicheren und leicht zu merkenden Passphrase wird auf der Webseite Diceware Passphrase (http://world.std.com/~reinhold/diceware.html) beschrieben.

Kapitel 20. GEOM: Modulares Framework zur Plattentransformation

Geschrieben von Tom Rhodes. Übersetzt von Daniel Seuffert und Johann Kois.

20.1. Übersicht

Dieses Kapitel behandelt den Einsatz von Laufwerken mit dem GEOM-Framework in FreeBSD. Dies beinhaltet auch die wichtigen RAID-Überwachungswerkzeuge, welche das Framework zur Konfiguration nutzen. Dieses Kapitel enthält keine tiefschürfenden Betrachtungen, wie GEOM I/O nutzt oder steuert, sein zugrundeliegendes Subsystem oder den Quelltext von GEOM. Diese Information wird durch die geom(4)-Manualpage und seine zahlreichen “SEE ALSO”-Verweise bereitgestellt. Dieses Kapitel ist auch kein ausführlicher Führer für RAID-Konfigurationen. Nur durch GEOM unterstützte RAID-Klassen werden erörtert.

Nach Lesen dieses Kapitels werden Sie folgendes wissen:

- Welche Art von RAID-Unterstützung durch GEOM verfügbar ist.
- Wie man die Basis-Dienstprogramme nutzt, um verschiedene RAID-Stufen zu konfigurieren, zu manipulieren und zu warten.
- Wie man mittels GEOM spiegelt, striped, verschlüsselt und entfernte Laufwerke verbindet.
- Wie man an Laufwerken, welche an das GEOM-Framework angeschlossen sind, Fehler behebt.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Verstehen, wie FreeBSD Laufwerke behandelt (Kapitel 19).
- Wissen wie man einen neuen FreeBSD-Kernel installiert und konfiguriert (Kapitel 9).

20.2. Einführung in GEOM

GEOM erlaubt den Zugriff und die Kontrolle von Klassen — Master Boot Records, BSD-Label usw. — durch die Nutzung von Datenträgern (Providern) oder den besonderen Dateien in `/dev`. Verschiedene Software RAID-Konfigurationen unterstützend, wird GEOM Ihnen transparenten Zugriff auf das Betriebssystem und System-Dienstprogramme gewähren.

20.3. RAID0 - Striping

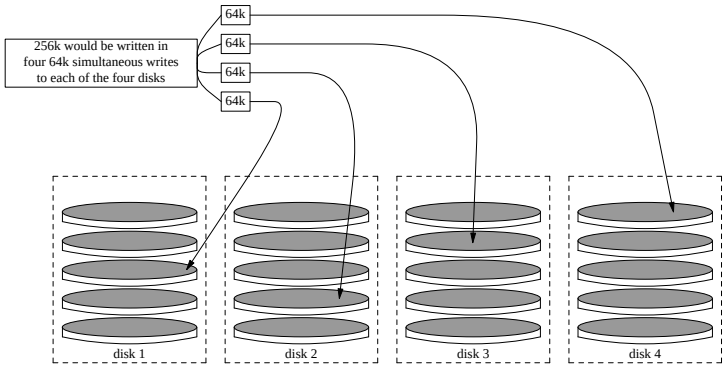
Geschrieben von Tom Rhodes und Murray Stokely.

Striping (stripe = Streifen) ist eine Methode, um verschiedene Laufwerke in einem einzigen Datenträger zusammenzufassen. In vielen Fällen wird dies durch die Nutzung von Hardware-Controllern bewerkstelligt. Das GEOM-Subsystem unterstützt Software-RAID0 (welches auch als Striping bekannt ist).

In einem RAID0-System werden die Daten in einzelne Blöcke aufgeteilt, welche über alle angeschlossenen Laufwerke in einem Datenfeld (Array) geschrieben werden. Anstatt darauf warten zu müssen, dass 256K auf ein

einzelnes Laufwerk geschrieben werden, kann ein RAID0-System gleichzeitig 64K auf jedes von 4 Laufwerken schreiben mit entsprechend besserer I/O-Leistung. Dieser Durchsatz kann durch die Verwendung mehrerer Controller noch zusätzlich gesteigert werden.

Jedes Laufwerk in einem RAID0-Stripe muss die gleiche Größe haben, da I/O-Anforderungen für das Lesen und Schreiben abwechselnd auf mehrere Laufwerke parallel erfolgen.



Erzeugen eines Stripe von unformatierten ATA-Platten

1. Laden Sie das `geom_stripe.ko`-Modul:

   ```
   # kldload geom_stripe
   ```

2. Stellen Sie sicher, dass ein geeigneter Mount-Punkt existiert. Falls dieser Datenträger eine Root-Partition werden soll, dann nutzen Sie zeitweise einen anderen Mount-Punkt, beispielsweise `/mnt`:

   ```
   # mkdir /mnt
   ```

3. Bestimmen Sie die Gerätenamen derjenigen Platten, welche gestriped werden sollen, und erzeugen Sie ein neues Stripe-Gerät. Das folgende Beispiel verwendet zwei unbenutzte und unpartitionierte ATA-Platten, die gestriped werden sollen. Lauten die Gerätenamen `/dev/ad2` und `/dev/ad3`, so verwenden Sie folgenden Befehl:

   ```
   # gstripe label -v st0 /dev/ad2 /dev/ad3
   Metadata value stored on /dev/ad2.
   Metadata value stored on /dev/ad3.
   Done.
   ```

4. Schreiben Sie einen Standard-Label (auch als Partitions-Tabelle bekannt) auf den neuen Datenträger und installieren Sie den normalen Bootstrap-Code:

   ```
   # bsdlabel -wB /dev/stripe/st0
   ```

5. Dieser Prozess sollte zwei weitere Geräte im Verzeichnis `/dev/stripe` (zusätzlich zum Gerät `st0`) erzeugt haben. Diese schliessen `st0a` und `st0c` ein. Nun kann ein Dateisystem auf dem Gerät `st0a` mit dem `newfs`-Dienstprogramm erzeugt werden:

   ```
   # newfs -U /dev/stripe/st0a
   ```

   Viele Zahlen rauschen nun über Ihren Bildschirm und nach ein paar Sekunden wird der Prozess abgeschlossen sein. Der Datenträger wurde erzeugt und kann in den Verzeichnisbaum eingehängt werden.

Um das erzeugte Stripe manuell zu mounten:

```
# mount /dev/stripe/st0a /mnt
```

Um das erzeugte Dateisystem automatisch während des Startvorgangs zu mounten, müssen Sie die Datenträgerinformation in die Datei `/etc/fstab` schreiben. Dazu legen Sie einen permanenten Mountpunkt namens `stripe` an:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /stripe ufs rw 2 2" \
    >> /etc/fstab
```

Das `geom_stripe.ko`-Modul muss ebenfalls automatisch beim Systemstart geladen werden (durch die Aufnahme der folgenden Zeile in die Datei `/boot/loader.conf`):

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

20.4. RAID1 - Spiegelung

Spiegelung (Mirroring) ist eine Technik, welche von vielen Firmen und Heimnutzern eingesetzt wird, um Daten ohne Unterbrechung zu sichern. Wenn ein Spiegel existiert, dann bedeutet dies einfach nur, dass PlatteB die PlatteA dupliziert. Oder PlatteC+D duplizieren PlatteA+A. Der wichtigste Aspekt ist, dass Daten einer Platte oder Partition dupliziert werden, unabhängig von der Konfiguration der Platte. Dadurch kann später diese Information leichter wiederhergestellt, ohne Zugriffsunterbrechung gesichert oder sogar physisch in einem Datentresor gelagert werden.

Stellen Sie zu Beginn sicher, dass ihr System zwei Platten mit identischer Größe aufweist. In dieser Übung gehen wir davon aus, dass es direkt zugängliche (da(4)) SCSI-Platten sind.

20.4.1. Die primäre Platte spiegeln

Angenommen, FreeBSD wurde auf der ersten Platte `da0` installiert, dann sollte gmirror(8) angewiesen werden, seine primären Daten auf dieser Platte zu speichern.

Bevor Sie den Spiegel aufbauen, sollten Sie die maximale Protokollierung aktivieren und den Zugang zum Gerät gestatten. Dazu setzen Sie die sysctl(8)-Option `kern.geom.debugflags` auf den folgenden Wert:

```
# sysctl kern.geom.debugflags=17
```

Nun können Sie den Spiegel aufbauen. Beginnen Sie den Prozess, indem Sie die Metadaten-Informationen auf das Gerät der primären Platte speichern. Konkret erzeugen Sie dabei das Gerät `/dev/mirror/gm`, indem Sie den folgenden Befehl ausführen:

> **Warnung:** Die Spiegelung der Bootplatte kann zu Datenverlust führen, wenn Sie Daten im letzten Sektor der Platte gespeichert haben. Dieses Risiko lässt sich minimieren, wenn Sie den Spiegel unmittelbar nach der Installation von FreeBSD aufsetzen. Die im folgenden beschriebene Vorgehensweise ist ebenfalls nicht kompatibel mit den Standard-Installationseinstellungen von FreeBSD 9.*x*, die das neue GPT-Partitionsschema verwenden. GEOM wird GPT-Metadaten überschreiben, was zu Datenverlust und einem möglicherweise nicht bootbarem System führt.

```
# gmirror label -vb round-robin gm0 /dev/da0
```

Ihr System sollte wie folgt antworten:

```
Metadata value stored on /dev/da0.
Done.
```

Initialisieren Sie GEOM. Dadurch wird das Kernelmodul `/boot/kernel/geom_mirror.ko` geladen:

```
# gmirror load
```

> **Anmerkung:** Wenn dieser Befehl erfolgreich ausgeführt wurde, wird die Gerätedatei `gm0` im Verzeichnis `/dev/mirror` erzeugt.

Stellen Sie sicher, dass das Kernelmodul `geom_mirror.ko` beim Systemstart automatisch geladen wird:

```
# echo 'geom_mirror_load="YES"' >> /boot/loader.conf
```

Bearbeiten Sie die Datei `/etc/fstab` und ersetzen Sie alle Verweise auf die alte Gerätedatei `da0` durch die neue Gerätedatei `gm0` des Plattenspiegels. Um die Datei `/etc/fstab` bearbeiten zu können, müssen Sie als Benutzer `root` am System angemeldet sein.

> **Anmerkung:** Sollte vi(1) ihr bevorzugter Texteditor sein, kouml;nnen Sie diese Änderungen ganz einfach wie folgt durchführen:
>
> ```
> # vi /etc/fstab
> ```
>
> Bevor Sie die Datei bearbeiten, sollten Sie ein Backup anlegen. Haben Sie die Datei mit vi(1) geöffnet, können Sie durch die Eingabe von **`:w /etc/fstab.bak`** eine Sicherungskopie der Datei anlegen. Danach ersetzen Sie alle alten Referenzen auf `da0` durch `gm0`, indem Sie **`:%s/da/mirror\/gm/g`** eingeben.

Die geänderte `fstab` sollte nun ähnlich wie im folgenden Beispiel aussehen. Es spielt dabei keine Rolle, ob Sie SCSI- oder ATA-Platten verwenden. Das RAID-Gerät heißt in jedem Fall `gm`.

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/mirror/gm0s1b                                    none            swap    sw              0
/dev/mirror/gm0s1a                            /               ufs     rw              1       1
/dev/mirror/gm0s1d                            /usr            ufs     rw              0       0
/dev/mirror/gm0s1f                            /home           ufs     rw              2       2
#/dev/mirror/gm0s2d                           /store          ufs     rw              2       2
/dev/mirror/gm0s1e                            /var            ufs     rw              2       2
/dev/acd0               /cdrom          cd9660  ro,noauto       0       0
```

Führen Sie einen Systemneustart durch:

```
# shutdown -r now
```

Wenn das System gestartet wird, sollten Sie nun nur noch `gm0`-Geräte anstatt der bisherigen `da0`-Geräte sehen. Nachdem das System vollständig initialisiert wurde, können Sie die neue Konfiguration testen, indem Sie den Befehl `mount` ausführen:

```
# mount
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/mirror/gm0s1a   1012974  224604   707334    24%    /
```

```
devfs                       1       1        0   100%    /dev
/dev/mirror/gm0s1f  45970182   28596 42263972     0%    /home
/dev/mirror/gm0s1d   6090094 1348356  4254532    24%    /usr
/dev/mirror/gm0s1e   3045006 2241420   559986    80%    /var
devfs                       1       1        0   100%    /var/named/dev
```

Hier ist alles in Ordnung. Alle Werte sehen aus wie erwartet. Um die Synchronisierung zu beginnen, integrieren Sie nun die Platte `da1` in den Spiegel, indem Sie den folgenden Befehl eingeben:

```
# gmirror insert gm0 /dev/da1
```

Während die Platten gespiegelt werden, können Sie den Fortschritt durch die Eingabe des folgenden Befehls überprüfen:

```
# gmirror status
```

Nachdem die Plattenspiegelung erfolgreich abgeschlossen wurde (und alle Daten synchronisiert wurden), sollte Sie eine Ausgabe ähnlich der folgenden erhalten, wenn Sie den Befehl erneut ausführen:

```
      Name    Status  Components
mirror/gm0  COMPLETE  da0
                      da1
```

Sollten Probleme aufgetreten oder sollte die Synchronisierung noch nicht abgeschlossen sein, wäre der Status `DEGRADED` anstatt `COMPLETE`.

20.4.2. Fehlerbehebung

20.4.2.1. Das System weigert sich zu starten

Falls das System startet und eine Eingabeaufforderung ähnlich der folgenden erscheint:

```
ffs_mountroot: can't find rootvp
Root mount failed: 6
mountroot>
```

Starten Sie den Rechner neu mit der Power- oder Resettaste. Wählen Sie im Startmenü Option sechs (6). Dadurch erscheint eine Eingabeaufforderung für loader(8). Laden Sie nun das Kernelmodul händisch:

```
OK? load geom_mirror
OK? boot
```

Falls dies funktioniert, wurde das Modul (aus welchen Gründen auch immer) nicht richtig geladen. Prüfen Sie, ob Ihr Eintrag in der Datei `/boot/loader.conf` korrekt ist. Sollte das Problem weiterhin bestehen, nehmen Sie die Zeile

```
options GEOM_MIRROR
```

in die Konfigurationsdatei des Kernels auf und führen Sie einen Rebuild und eine erneute Installation durch. Dies sollte das Problem beseitigen.

20.4.3. Wiederherstellung des Systems nach einem Plattenausfall

Das Schöne an der Plattenspiegelung ist, dass eine kaputte Platte ersetzt werden kann, ohne dass Sie dabei Daten verlieren.

Basierend auf der vorhin besprochenen RAID1-Konfiguration, nehmen wir nun an, dass die Platte `da1` ausgefallen ist und daher ersetzt werden muss. Um dies zu tun, müssen Sie feststellen, welche Platte ausgefallen ist und das System herunterfahren. Nun können Sie die kaputte Platte gegen eine neue Platte austauschen und das System wieder starten. Nachdem der Systemstart abgeschlossen ist, verwenden Sie die folgenden Befehle, um die Plattenspiegelung wieder zu reaktivieren:

```
# gmirror forget gm0

# gmirror insert gm0 /dev/da1
```

Der Befehl `gmirror status` erlaubt es Ihnen, den Fortschritt bei der Wiederherstellung der Plattenspiegelung zu beobachten. Das ist alles, was Sie tun müssen.

20.5. GEOM Gate Netzwerkgeräte

GEOM unterstützt die Verwendung entfernter Geräte wie Festplatten, CD-ROMs, Dateien usw. mittels Nutzung der Gate-Dienstprogramme. Dies ist vergleichbar mit NFS.

Zu Beginn muss eine Exportdatei erzeugt werden. Diese Datei legt fest, wer Zugriff auf die exportierten Ressourcen hat und welche Zugriffstechniken angeboten werden. Um zum Beispiel den vierten Slice auf der ersten SCSI-Platte zu exportieren, ist die folgende Datei `/etc/gg.exports` mehr als ausreichend:

```
192.168.1.0/24 RW /dev/da0s4d
```

Sie wird allen Hosts innerhalb des privaten Netzwerkes den Zugriff auf das Dateisystem auf der Partition `da0s4d` erlauben.

Um dieses Gerät zu exportieren, stellen Sie bitte sicher, dass es momentan nicht gemounted ist und starten Sie den ggated(8) Server-Daemon:

```
# ggated
```

Um nun `mount` auf der Client-Maschine auszuführen, geben Sie bitte die folgenden Befehle ein:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
ggate0
# mount /dev/ggate0 /mnt
```

Von nun an kann auf das Gerät über den Mount-Punkt `/mnt` zugegriffen werden.

> **Anmerkung:** Es sollte darauf hingewiesen werden, dass dies scheitern wird, falls das Gerät momentan entweder auf dem Server oder irgendeiner anderen Maschine gemountet ist.

Wenn das Gerät nicht länger gebraucht wird, dann kann es mit dem Befehl umount(8) ausgehängt werden (genauso wie jedes andere Laufwerk auch).

20.6. Das Labeln von Laufwerken

Während der Initialisierung des Systems legt der FreeBSD-Kernel für jedes gefundene Gerät Knotenpunkte an. Diese Methode für die Überprüfung auf vorhandene Geräte wirft einige Fragen auf. Was passiert beispielsweise, wenn ein neues USB-Laufwerk hinzugefügt wird? Es ist sehr wahrscheinlich, dass ein Flash-Speicher-Gerät den Gerätenamen `da0` erhält, während gleichzeitig das bisherige `da0` zu `da1` wird. Dies verursacht Probleme beim Einhängen von Dateisystemen, wenn diese in der `/etc/fstab` aufgeführt sind und schlussendlich mag das auch dazu führen, dass das System nicht mehr startet.

Eine Lösung für dieses Problem ist das Aneinanderketten der SCSI-Geräte, damit ein neues Gerät, welches der SCSI-Karte hinzugefügt wird, unbenutzte Gerätenummern erhält. Aber was geschieht, wenn ein USB-Gerät möglicherweise die primäre SCSI-Platte ersetzt? Dies kann passieren, weil USB-Geräte normalerweise vor der SCSI-Karte geprüft werden. Eine Lösung ist das Hinzufügen dieser Geräte, nachdem das System gestartet ist. Eine andere Lösung könnte sein, nur ein einzelnes ATA-Laufwerk zu nutzen und die SCSI-Geräte niemals in der `/etc/fstab` aufzuführen.

Es gibt allerdings eine bessere Lösung. Durch Verwendung des `glabel`-Dienstprogramms kann ein Administrator oder Benutzer seine Laufwerke mit Labeln versehen und diese in der `/etc/fstab` nutzen. Da `glabel` seine Label im letzten Sektor jedes vorhandenen Datenträgers speichert, wird das Label persistent bleiben (auch über Neustarts hinweg). Durch Nutzung dieses Labels als Gerät kann das Dateisystem immer gemountet sein, unabhängig davon, durch welchen Geräte-Knotenpunkt auf ihn zugegriffen wird.

> **Anmerkung:** Der Label muss permanent (dauerhaft) sein. Man kann das Dienstprogramm `glabel` nutzen, um sowohl transiniente als auch permanente Label zu erzeugen. Aber nur permanente (persistente) Label bleiben konsistent über Neustarts hinweg. Lesen Sie die glabel(8)-Manualpage für weitere Unterschiede zwischen den Label-Typen.

20.6.1. Label-Typen und Beispiele

Es gibt zwei Arten von Labeln: generische Label und Dateisystem-Label. Label können dauerhaft (permanent) oder temporär sein. Permanente Label können mit tunefs(8) oder newfs(8) in einem speziellen spezielles Verzeichnis in `/dev` erzeugt werden, welches entsprechend der Dateisystem-Art benannt wird. UFS2-Dateisystem-Label werden zum Beispiel im Verzeichnis `/dev/ufs` angelegt. Permanente Label können außerdem durch den Befehl `glabel label` erzeugt werden. Diese Label sind dann allerdings nicht dateisystemspezisch und werden im Unterverzeichnis `/dev/label` erzeugt.

Ein temporäres Label verschwindet mit dem nächsten Systemstart. Diese Label werden im Verzeichnis `/dev/label` erzeugt und sind ideal für Testzwecke. Ein temporäres Label kann durch den Befehl `glabel create` erzeugt werden. Weitere Informationen finden sich in der Manualpage glabel(8).

Um ein permanentes Label auf einem UFS2-Dateisystem ohne Löschung von Daten zu erzeugen, kann man folgenden Befehl verwenden:

```
# tunefs -L home /dev/da3
```

> **Warnung:** Wenn das Dateisystem voll ist, kann dies zu Datenkorruption führen; aber egal wie, falls das Dateisystem voll ist, sollte das Hauptziel die Entfernung ungenützter Dateien und nicht das Hinzufügen von Labeln sein.

Ein Label sollte nun in `/dev/ufs` vorhanden sein, der zu `/etc/fstab` hinzugefügt wird:

```
/dev/ufs/home           /home           ufs     rw              2       2
```

> **Anmerkung:** Das Dateisystem darf nicht gemountet sein beim Versuch, `tunefs` auszuführen.

Nun kann das Dateisystem wie üblich gemountet werden:

```
# mount /home
```

Von nun an kann der Geräte-Knotenpunkt sich ohne negative Effekte auf das System ändern, solange das Kernelmodul `geom_label.ko` beim Systemstart mittels `/boot/loader.conf` geladen wird oder die `GEOM_LABEL`-Kernel-Option aktiv ist.

Dateisysteme können auch mit einem Standard-Label erzeugt werden (mittels des Flags `-L` in `newfs`). Lesen Sie bitte die Manualpage von newfs(8) für weitere Informationen.

Der folgende Befehl kann genutzt werden, um das Label zu beseitigen:

```
# glabel destroy home
```

Das folgende Beispiel zeigt Ihnen, wie Sie Label für die Partitionen einer Bootplatte erzeugen.

Beispiel 20-1. Die Partitionen einer Bootplatte labeln

Durch das Erstellen von permanenten Labeln für die Partitionen einer Bootplatte sollte das System selbst dann noch normal starten können, wenn Sie die Platte an einen anderen Controller anschließen oder in ein anderes System installieren. In diesem Beispiel nehmen wir an, dass nur eine einzige ATA-Platte verwendet wird, die Ihr System derzeit als `ad0` erkennt. Weiters nehmen wir an, dass Sie das Standard-Partionierungsschema von FreeBSD vewendet haben und Ihre Platte daher die Dateisysteme `/`, `/var`, `/usr` sowie `/tmp` aufweist. Zusätzlich wurde eine Swap-Partition angelegt.

Starten Sie das System neu. Am loader(8)-Prompt drücken Sie die Taste **4**, um in den Single-User-Modus zu gelangen. Dort führen Sie die folgenden Befehle aus:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs
# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var
# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
# glabel label tmp /dev/ad0s1e
GEOM_LABEL: Label for provider /dev/ad0s1e is label/tmp
# glabel label swap /dev/ad0s1b
GEOM_LABEL: Label for provider /dev/ad0s1b is label/swap
# exit
```

Das System startet daraufhin in den Multi-User-Modus. Nachdem der Startvorgang abgeschlossen ist, editieren Sie die Datei `/etc/fstab` und ersetzen die konventionellen Gerätedateien durch die entsprechenden Label. Ihre modifizierte `/etc/fstab` sollte nun ähnlich der folgenden Ausgabe aussehen:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/label/swap         none            swap    sw              0       0
```

```
/dev/label/rootfs       /               ufs     rw              1       1
/dev/label/tmp          /tmp            ufs     rw              2       2
/dev/label/usr          /usr            ufs     rw              2       2
/dev/label/var          /var            ufs     rw              2       2
```

Starten Sie Ihr System neu. Traten keine Probleme auf, wird das System normal hochfahren und Sie erhalten die folgende Ausgabe, wenn Sie den Befehl `mount` ausführen:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

Beginnend mit FreeBSD 7.2, unterstützt glabel(8) einen neuen Labeltyp für UFS-Dateisysteme. Dieser basiert auf der eindeutigen Dateisystem-ID `ufsid`. Derartige Label finden sich im Verzeichnis `/dev/ufsid` und werden während des Systemstarts automatisch erzeugt. Es ist möglich, diese `ufsid`-Label zum automatischen Einhängen von Partitionen in der Datei `/etc/fstab` einzusetzen. Verwenden Sie den Befehl `glabel status`, um eine Liste aller Dateisysteme und ihrer `ufsid`-Label zu erhalten:

```
% glabel status
                  Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

In diesem Beispiel repräsentiert `ad4s1d` das `/var`-Dateisystem, während `ad4s1f` dem `/usr`-Dateisystem entspricht. Wenn Sie die angegebenen `ufsid`-Werte verwenden, können diese Dateisysteme durch die folgenden Einträge in der Datei `/etc/fstab` gemountet werden:

```
/dev/ufsid/486b6fc38d330916        /var        ufs        rw        2      2
/dev/ufsid/486b6fc16926168e        /usr        ufs        rw        2      2
```

Jede Partition, die ein `ufsid`-Label aufweist, kann auf diese Art gemountet werden. Dies hat den Vorteil, dass Sie keine permanenten Label mehr anlegen müssen, wobei sich die Platten nach wie vor über geräteunabhängige Namen ansprechen und mounten lassen.

20.7. UFS Journaling in GEOM

Mit FreeBSD 7.0 wurde eine lang erwartete Funktion, das Journaling, implementiert. Diese Funktion wird über das GEOM-Subsystem realisiert und kann über das Werkzeug gjournal(8) eingerichtet werden.

Was ist Journaling? Bei Journaling wird ein Protokoll über alle Dateisystemtransaktionen angelegt, inklusive aller Veränderungen, aus denen ein kompletter Schreibvorgang besteht, bevor diese Änderungen (Metadaten sowie tatsächliche Schreibvorgänge) physisch auf der Festplatte ausgeführt werden. Dieses Protokoll kann später erneut aufgerufen werden, um diese Vorgänge zu wiederholen (beispielsweise um Systeminkonsistenzen zu vermeiden).

Diese Technik bietet eine weitere Möglichkeit, sich vor Datenverlust und Dateisystem-Inkonsistenzen zu schützen. Im Gegensatz zu Soft Updates (die Metadaten-Aktualisierungen verfolgen und erzwingen) und Snapshots (die ein Image eines Dateisystems darstellen) wird bei Journaling ein tatsächliches Protokoll in einem speziell dafür bereitgestellten Bereich der Festplatte (oder manchmal sogar auf einer separaten Platte) gespeichert.

Im Gegensatz zu anderen Journaling-Dateisystemen arbeitet die `gjournal`-Methode blockbasiert und wurde nicht als Teil des Dateisystems implementiert, sondern als GEOM-Erweiterung.

Um die `gjournal`-Unterstützung zu aktivieren, muss der FreeBSD-Kernel die folgende Option enthalten (was seit FreeBSD 7.0 bereits in der Voreinstellung der Fall ist):

```
options UFS_GJOURNAL
```

Um ein Volume mit Journalunterstützung beim Systemstart automatisch zu mounten, muss das Kernelmodul `geom_journal.ko` ebenfalls automatisch geladen werden (durch einen entsprechenden Eintrag in der Datei `/boot/loader.conf`):

```
geom_journal_load="YES"
```

Alternativ können Sie auch einen angepassten Kernel bauen, der diese Funktionalität enthält, indem Sie die folgende Zeile in Ihrer Kernelkonfigurationsdatei aufnehmen:

```
options     GEOM_JOURNAL
```

Das Anlegen eines neuen Journals auf einem freien Dateisystem erfolgt durch die folgenden Schritte (im Folgenden wird angenommen, dass es sich bei `da4` um eine neue SCSI-Platte handelt):

```
# gjournal load
# gjournal label /dev/da4
```

Danach sollten die Gerätedateien `/dev/da4` sowie `/dev/da4.journal` vorhanden sein. Nun können Sie auf diesem Gerät ein Dateisystem anlegen:

```
# newfs -O 2 -J /dev/da4.journal
```

Dieser Befehl erzeugt ein UFS2-Dateisystem auf dem Gerät, für das im letzten Schritt das Journaling aktiviert wurde.

Danach hängen Sie das neue Dateisystem mit `mount` in Ihren Verzeichnisbaum ein:

```
# mount /dev/da4.journal /mnt
```

> **Anmerkung:** Falls auf Ihrem System mehrere Slices angelegt sind (beispielsweise `ad4s1` sowie `ad4s2`), wird `gjournal` für jedes Slice ein Journal anlegen (also `ad4s1.journal` sowie `ad4s2.journal`).

Um die Leistung zu optimieren, kann das Journal auf eine externe Platte ausgelagert werden. In einem solchen Fall geben Sie die Gerätedatei der Platte nach dem Gerät an, für das Sie Journaling aktivieren wollen. Theoretisch ist es auch möglich, Journaling auf bereits existierenden Dateisystemen durch das Werkzeug `tunefs` zu aktivieren. Machen Sie aber in jedem Fall ein Backup Ihrer Daten, bevor Sie versuchen, ein existierendes Dateisystem zu ändern. `gjournal` wird zwar den Vorgang abbrechen, wenn es das Journal nicht erzeugen kann, allerdings schützt Sie dies nicht vor Datenverlust durch einen fehlerhaften Einsatz von `tunefs`.

Es ist möglich, Journale auch für die Bootplatte eines FreeBSD-System zu verwenden. Lesen Sie bitte den Artikel Implementing UFS Journaling on a Desktop PC (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/gjournal-desktop/article.html), wenn Sie an einer derartigen Konfiguration interessiert sind.

Kapitel 21. Dateisystemunterstützung

Geschrieben von Tom Rhodes. Übersetzt von Benedict Reuschling und Daniel Seuffert.

21.1. Übersicht

Dateisysteme sind ein wesentlicher Bestandteil von Betriebssystemen. Sie erlauben es den Benutzern Dateien zu laden und zu speichern, ermöglichen den Zugriff auf die Daten und machen Festplatten überhaupt erst nützlich. Unterschiedliche Betriebssysteme besitzen normalerweise eine Gemeinsamkeit, nämlich deren mitgeliefertes Dateisystem. Bei FreeBSD ist dieses Dateisystem bekannt unter dem Namen Fast File System FFS, das direkt auf dem Original-Unix™ Dateisystem, UFS genannt, basiert. Dieses ist das von FreeBSD mitgelieferte Dateisystem, das auf Festplatten für den Dateizugriff verwendet wird.

FreeBSD unterstützt auch eine Vielzahl von anderen Dateisystemen, um auf Daten von anderen Betriebssystemen lokal zuzugreifen, wie z.B. Daten auf USB-Speichermedien, Flash-Speichern und Festplatten. Es gibt auch Unterstützung für fremde Dateisysteme. Dabei handelt es sich um Dateisysteme, die auf anderen Betriebssystemen entwickelt wurden, wie beispielsweise das Linux Extended File System (EXT) und das Z-Dateisystem (ZFS) von Sun.

Es gibt verschiedene Stufen der Unterstützung in FreeBSD für diese unterschiedlichen Dateisysteme. Manche benötigen ein geladenes Kernelmodul, andere die Installation bestimmter Werkzeuge. Dieses Kapitel dient dazu, den Benutzern von FreeBSD dazu helfen, auf andere Dateisysteme zuzugreifen, beginnend mit Suns Z-Dateisystem (ZFS).

Nachdem Sie dieses Kapitel gelesen haben, werden Sie die folgenden Dinge wissen:

- Den Unterschied zwischen eingebauten und unterstützten Dateisystemen.
- Welche Dateisysteme von FreeBSD unterstützt werden.
- Wie man fremde Dateisysteme aktiviert, konfiguriert, darauf zugreift und diese verwendet.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Grundlagen von UNIX und FreeBSD verstehen (Kapitel 4).
- Mit den Grundlagen der Konfiguration und dem Bauen des Kernels vertraut sein (Kapitel 9).
- Problemlos Software von Drittherstellern in FreeBSD installieren können (Kapitel 5).
- sich ein wenig mit Festplatten, Speicher und Gerätenamen in FreeBSD auskennen (Kapitel 19).

21.2. Das Z-Dateisystem (ZFS)

Das Z-Dateisystem ist eine neue von Sun entwickelte Technologie, mit dem Konzept einer gepoolten Speichermethodik. Das bedeutet, dass Speicher nur verwendet wird, wenn dieser als Datenspeicher benutzt wird. ZFS wurde auch für maximale Datenintegrität entwickelt und unterstützt dabei mehrfache Kopien, Schnappschüsse und Prüfsummen für Daten. Ein neues Datenreplikationsmodell, bekannt als RAID-Z, wurde ebenfalls hinzugefügt. Das RAID-Z-Modell ist ähnlich zu RAID5, wurde aber mit dem Ziel entworfen, Datenverfälschung beim Schreiben zu verhindern.

21.2.1. ZFS Einstellungen

Das ZFS-Teilsystem benötigt viele Systemressourcen, weshalb gewisse Einstellungen notwendig sind, um maximale Effizienz während des täglichen Gebrauchs zu gewährleisten. Da es sich um eine experimentelle Funktion in FreeBSD handelt, wird sich das in naher Zukunft ändern. Wie dem auch sei, zum gegenwärtigen Zeitpunkt wird die Anwendung der folgenden Schritte empfohlen.

21.2.1.1. Hauptspeicher

Der verfügbare Hauptspeicher im System sollte mindestens 1 Gigabyte betragen, jedoch werden 2 Gigabyte oder mehr empfohlen. In allen gezeigten Beispielen in diesem Abschnitt verwendet das System 1 Gigabyte Hauptspeicher mit mehreren anderen Einstellungen.

Manche Nutzer hatten Erfolg bei der Verwendung von weniger als 1 GB Hauptspeicher, aber mit dieser begrenzten Menge an RAM ist es sehr wahrscheinlich, dass FreeBSD eine Panic wegen erschöpftem Hauptspeicher erleiden wird, wenn es hohen Belastungen ausgesetzt ist.

21.2.1.2. Kernelkonfiguration

Es wird vorgeschlagen, nicht benötigte Treiber und Optionen aus der Kernelkonfigurationsdatei zu entfernen. Da die meisten Geräte als Module verfügbar sind, können diese einfach mittels der Datei `/boot/loader.conf` geladen werden.

Nutzer der i386-Architektur sollten die folgende Option in ihrer Kernelkonfigurationsdatei hinzufügen, den Kernel neu erstellen und anschliessend das System neustarten:

```
options         KVA_PAGES=512
```

Diese Option wird den Adressraum des Kernels vergrössern, was es ermöglicht, die Einstellung `vm.kvm_size` über die momentan verhängte Grenze von 1 GB (2 GB für PAE) zu erhöhen. Um den passenden Wert dieser Option zu ermitteln, teilen Sie den gewünschten Adressraum in Megabyte durch vier. In diesem Fall beträgt er `512` für 2 GB.

21.2.1.3. Einstellungen des Loaders

Der `kmem`-Addressraum sollte auf allen FreeBSD-Architekturen erhöht werden. Die folgende Option, die dem Testsystem mit einem Gigabyte Hauptspeicher der Datei `/boot/loader.conf` hinzugefügt und welches anschliessend neu gestartet wurde, war erfolgreich:

```
vm.kmem_size="330M"
vm.kmem_size_max="330M"
vfs.zfs.arc_max="40M"
vfs.zfs.vdev.cache.size="5M"
```

Eine detailliertere Liste von Vorschlägen zu ZFS-verwandten Einstellungen finden Sie unter http://wiki.freebsd.org/ZFSTuningGuide.

21.2.2. Verwenden von ZFS

Es existiert ein Startmechanismus, der es FreeBSD erlaubt, ZFS als Pool während des Systemstarts zu initialisieren. Um das zu tun, geben Sie die folgenden Befehle ein:

```
# echo 'zfs_enable="YES"' >> /etc/rc.conf
# /etc/rc.d/zfs start
```

Für den Rest dieses Dokuments wird angenommen, dass drei SCSI-Platten im System verfügbar sind und dass deren Gerätenamen *da0*, *da1* und *da2* lauten. Benutzer von IDE-Hardware können *ad*-Geräte an Stelle von SCSI-Hardware einsetzen.

21.2.2.1. Pool mit nur einer Platte

Um ein einfaches, nicht-redundantes ZFS auf einer einzelnen Festplatte zu erstellen, benutzen Sie das `zpool`-Kommando:

```
# zpool create example /dev/da0
```

Um den neuen Pool anzusehen, überprüfen Sie die Ausgabe von `df`:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

Diese Ausgabe zeigt deutlich, dass der `example`-Pool nicht nur erstellt, sondern auch *gemountet* wurde. Er ist genau wie andere Dateisysteme verfügbar, Dateien können darin erstellt und von den Benutzern aufgelistet werden, wie im folgenden Beispiel gezeigt wird:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r--   1 root  wheel    0 Aug 29 23:15 testfile
```

Leider verwendet dieser Pool keine der Vorteile der ZFS-Eigenschaften. Erstellen Sie ein Dateisystem auf diesem Pool und aktivieren Sie die Komprimierung darauf:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

Jetzt ist `example/compressed` ein von ZFS komprimiertes Dateisystem. Versuchen Sie, ein paar grosse Dateien in das Verzeichnis `/example/compressed` zu kopieren.

Die Komprimierung kann jetzt deaktiviert werden mittels:

```
# zfs set compression=off example/compressed
```

Um das Dateisystem aus dem Verzeichnisbaum abzuhängen, geben Sie den folgenden Befehl ein und vergewissern Sie sich über `df` vom Erfolg dieser Aktion:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
example      17547008       0 17547008     0%    /example
```

Mounten Sie das Dateisystem erneut, um es wieder verfügbar zu machen und bestätigen Sie mit `df`:

```
# zfs mount example/compressed
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
```

Der Pool und das Dateisystem können genausogut über die Ausgabe von `mount` überwacht werden:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/data on /example/data (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

Wie zu beobachten ist, können ZFS-Dateisysteme nach deren Erstellung genauso wie normale Dateisysteme verwendet werden, jedoch sind auch noch viele andere Eigenschaften verfügbar. Im folgenden Beispiel wird ein neues Dateisystem, `data`, erstellt. Wichtige Dateien sollen hier gespeichert werden, weshalb das Dateisystem angewiesen wird, jeweils zwei Kopien jedes Datenblocks zu unterhalten:

```
# zfs create example/data
# zfs set copies=2 example/data
```

Es ist nun möglich, den Speicherplatzverbrauch der Daten mittels `df` erneut zu betrachten:

```
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
example/data        17547008       0 17547008     0%    /example/data
```

Beachten Sie, dass jedem Dateisystem des Pools die gleiche Menge an Speicher zur Verfügung steht. Das ist der Grund für die Verwendung von `df` in all diesen Beispielen, da es zeigt, dass das Dateisystem nur den Speicher

belegt, den es auch benötigt und alles wird von dem gleichen Pool abgezogen. ZFS macht Konzepte wie Volumen und Partitionen überflüssig und erlaubt mehrere Dateisysteme auf demselben Pool. Zerstören Sie die Datensysteme und anschliessend den Pool, da sie nicht länger gebraucht werden:

```
# zfs destroy example/compressed
# zfs destroy example/data
# zpool destroy example
```

Festplatten werden mit der Zeit schlechter und fallen aus, eine unvermeidliche Tatsache. Wenn diese Platte ausfällt, sind die Daten verloren. Eine Möglichkeit, diesen Datenverlust beim Plattenausfall zu vermeiden, ist die Verwendung von RAID. ZFS unterstützt diese Eigenschaft im Entwurf seiner Pools und wird im nächsten Abschnitt behandelt.

21.2.2.2. ZFS RAID-Z

Wie zuvor bereits erwähnt, wird in diesem Abschnitt angenommen, dass drei SCSI-Geräte vorhanden sind (`da0`, `da1` und `da1` bzw. `ad0` und so weiter, falls IDE-Platten verwendet werden). Um einen RAID-Z Pool zu erstellen, geben Sie das folgende Kommando ein:

```
# zpool create storage raidz da0 da1 da2
```

> **Anmerkung:** Sun empfiehlt, dass die Anzahl von Geräten in einer RAID-Z Konfiguration drei bis neun beträgt. Falls Ihre Anforderungen unbedingt einen einzelnen Pool, bestehend aus zehn oder mehr Platten, erfordern, sollten Sie überlegen, diesen in kleinere RAID-Z Gruppen aufzuteilen. Sollten Sie nur zwei Platten zur Verfügung haben und trotzdem Redundanz benötigen, ziehen Sie den Einsatz der ZFS-Mirror (Spiegel) Fähigkeiten in Betracht. Lesen Sie die zpool(8) Manualpage, um mehr darüber zu erfahren.

Der `storage`-zPool sollte jetzt erstellt worden sein. Sie können das überprüfen, indem Sie die Befehle mount(8) und df(1) wie zuvor verwenden. Weitere Plattenspeicher können an das Ende der oben stehenden Liste hinzugefügt werden. Erstellen Sie ein neues Dateisystem in dem Pool, `home` genannt, in dem später Dateien von Benutzern platziert werden:

```
# zfs create storage/home
```

Nun kann die Komprimierung aktiviert und zusätzliche Kopien der Benutzerverzeichnisse und der darin enthaltenen Dateien angelegt werden. Dies geschieht über die gleichen Befehle wie bereits zuvor:

```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```

Um dieses Verzeichnis als neues Benutzerverzeichnis zu verwenden, kopieren Sie die Nutzerdaten dort hin und erstellen Sie die entsprechenden Symlinks:

```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```

Anwender sollten jetzt ihre Daten in dem neu angelegten `/storage/home` Dateisystem auffinden. Prüfen Sie das, indem Sie einen neuen Benutzer hinzufügen und sich als dieser Benutzer am System anmelden.

Versuchen Sie, einen Schnappschuss anzulegen, der später wieder zurückgerollt werden kann:

```
# zfs snapshot storage/home@08-30-08
```

Beachten Sie, dass die Schnappschuss-Option nur auf echte Dateisysteme, jedoch nicht auf Verzeichnisse oder eine Datei angewendet werden kann. Das @-Zeichen dient als Begrenzer zwischen dem Dateisystem- oder Volumenamen. Wenn ein Benutzerverzeichnis zerstört wird, können Sie es über den folgenden Befehl wieder herstellen:

```
# zfs rollback storage/home@08-30-08
```

Um eine Liste von allen verfügbaren Schnappschüssen zu erhalten, starten Sie das `ls`-Kommando in Verzeichnis `.zfs/snapshot` des entsprechenden Dateisystems. Beispielsweise können Sie den vorhin angelegten Schnappschuss mit dem folgenden Befehl auflisten:

```
# ls /storage/home/.zfs/snapshot
```

Es ist möglich ein Skript zu schreiben, dass monatliche Schnappschüsse der Nutzerdaten anlegt. Allerdings werden die Schnappschüsse mit der Zeit eine grosse Menge an Speicherplatz einnehmen. Den vorherigen Schnappschuss können Sie über das folgende Kommando löschen:

```
# zfs destroy storage/home@08-30-08
```

Nach all diesen Tests gibt es keinen Grund, das Verzeichnis `/storage/home` noch länger in seinem momentanen Zustand zu belassen. Ernennen Sie es zum echten `/home`-Dateisystem:

```
# zfs set mountpoint=/home storage/home
```

Die Eingabe der Befehle `df` und `mount` zeigt, dass das System das Dateisystem nun als das echte `/home` behandelt:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

Damit ist die RAID-Z-Konfiguration abgeschlossen. Um über den Status des Dateisystems mittels des nächtlichen periodic(8)-Skripts auf dem Laufenden gehalten zu werden, geben Sie das folgende Kommando ein:

```
# echo 'daily_status_zfs_enable="YES"' >> /etc/periodic.conf
```

21.2.2.3. Wiederherstellung von RAID-Z

Jedes Software-RAID besitzt Verfahren, um dessen `Zustand` zu überwachen. ZFS ist da keine Ausnahme. Der Status von RAID-Z Geräten kann mittels des folgenden Kommandos betrachtet werden:

```
# zpool status -x
```

Wenn alle Pools gesund sind und alles normal ist, wird die folgende Nachricht zurückgegeben:

```
all pools are healthy
```

Wenn ein Problem existiert (möglicherweise ist eine Platte ausgefallen), wird der Zustand des Pools ähnlich dem Folgenden ausgegeben:

```
  pool: storage
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
       Sufficient replicas exist for the pool to continue functioning in a
       degraded state.
action: Online the device using 'zpool online' or replace the device with
       'zpool replace'.
 scrub: none requested
config:

       NAME        STATE     READ WRITE CKSUM
       storage     DEGRADED     0     0     0
         raidz1    DEGRADED     0     0     0
           da0     ONLINE       0     0     0
           da1     OFFLINE      0     0     0
           da2     ONLINE       0     0     0

errors: No known data errors
```

Das bedeutet, dass das Gerät vom Systemadministrator abgeschaltet wurde. In diesem Fall trifft das zu. Um eine Platte abzuschalten, wurde das folgende Kommando eingegeben:

```
# zpool offline storage da1
```

Es ist jetzt möglich, `da1` zu ersetzen, nachdem das System ausgeschaltet wurde. Wenn das System wieder läuft, kann der folgende Befehl benutzt werden, um die Platte zu ersetzen:

```
# zpool replace storage da1
```

Von da an kann der Status erneut überprüft werden, jedoch dieses Mal ohne die Option `-x`, um die Zustandsinformation zu bekommen:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
config:

       NAME        STATE     READ WRITE CKSUM
       storage     ONLINE       0     0     0
         raidz1    ONLINE       0     0     0
           da0     ONLINE       0     0     0
           da1     ONLINE       0     0     0
           da2     ONLINE       0     0     0
```

```
errors: No known data errors
```

Wie in diesem Beispiel gezeigt, scheint alles wieder normal zu sein.

21.2.2.4. Datenüberprüfung

Wie bereits erwähnt, verwendet ZFS `Prüfsummen`, um die Integrität der gespeicherten Daten zu verifizieren. Die Prüfsummen werden automatisch beim Erstellen des Dateisystem aktiviert und können über den folgenden Befehl deaktiviert werden:

```
# zfs set checksum=off storage/home
```

Das ist jedoch kein schlauer Einfall, da die Prüfsummen nur ganz wenig Speicherplatz einnehmen und viel nützlicher sind, wenn Sie aktiviert bleiben. Es scheint auch kein nennenswerter Ressourcenverbrauch mit deren Aktivierung verbunden zu sein. Wenn die Prüfsummen aktiv sind, kann ZFS die Datenintegrität über den Vergleich der Prüfsummen gewährleisten. Dieser Prozess wird als “reinigen” bezeichnet. Um die Datenintegrität des `storage`-Pools zu überprüfen, geben Sie den folgenden Befehl ein:

```
# zpool scrub storage
```

Dieser Prozess kann einige Zeit in Anspruch nehmen, abhängig davon, wieviele Daten gespeichert sind. Es handelt sich dabei auch um eine I/O-intensive Aktion, weshalb auch jeweils nur eine dieser Operationen durchgeführt werden darf. Nachdem die Reinigung abgeschlossen ist, wird der Status aktualisiert und kann über eine Statusabfrage eingesehen werden:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: scrub completed with 0 errors on Sat Aug 30 19:57:37 2008
config:

        NAME        STATE     READ WRITE CKSUM
        storage     ONLINE       0     0     0
          raidz1    ONLINE       0     0     0
            da0     ONLINE       0     0     0
            da1     ONLINE       0     0     0
            da2     ONLINE       0     0     0

errors: No known data errors
```

Die Zeit des Abschlusses der Aktion kann in diesem Beispiel direkt abgelesen werden. Die Prüfsummen helfen dabei, sicherzustellen, dass die Datenintegrität über einen langen Zeitraum hinaus erhalten bleibt.

Es gibt viele weitere Optionen für das Z-Dateisystem, lesen Sie dazu die Manualpage zfs(8) und zpool(8).

Kapitel 22. Der Vinum Volume Manager

Ursprünglich geschrieben von Greg Lehey. Übersetzt von Johann Kois und Kay Abendroth.

22.1. Übersicht

Egal, über welche und wieviele Festplatten Ihr System auch verfügt, immer wieder werden Sie mit den folgenden Problemen konfrontiert:

- Ihre Platten sind zu klein.
- Sie sind zu langsam.
- Ihre Platten sind unzuverlässig.

Um derartige Probleme zu lösen, wurden verschiedene Methoden entwickelt. Eine Möglichkeit bietet der Einsatz von mehreren, manchmal auch redundant ausgelegten Platten. Zusätzlich zur Unterstützung verschiedener Erweiterungskarten und Controller für Hardware-RAID-Systeme enthält das FreeBSD-Basissystem auch den Vinum Volume Manager, einen Blockgerätetreiber, der die Einrichtung virtueller Platten unterstützt. Bei *Vinum* handelt es sich um einen sogenannten *Volume Manager*, einen virtuellen Plattentreiber, der obige drei Probleme löst. Vinum bietet Ihnen größere Flexibilität, Leistung und Zuverlässigkeit als die klassische Datenspeicherung auf einzelne Festplatten. Dazu unterstützt Vinum RAID-0, RAID-1 und RAID-5 (sowohl einzeln als auch in Kombination).

Dieses Kapitel bietet Ihnen einen Überblick über potentielle Probleme der klassischen Datenspeicherung auf Festplatten sowie eine Einführung in den Vinum Volume Manager.

> **Anmerkung:** Für FreeBSD 5.X wurde Vinum überarbeitet und an die GEOM-Architektur (Kapitel 20) angepasst, wobei die ursprünglichen Ideeen und Begriffe sowie die auf der Platte benötigten Metadaten beibehalten wurden. Die überarbeitete Version wird als *gvinum* (für *GEOM-Vinum*) bezeichnet. Die folgenden Ausführungen verwenden den Begriff *Vinum* als abstrakten Namen, unabhängig davon, welche Variante implementiert wurde. Sämtliche Befehlsaufrufe erfolgen über `gvinum`, welches nun das Kernelmodul `geom_vinum.ko` (statt `vinum.ko`) benötigt. Analog finden sich alle Gerätedateien nun unter `/dev/gvinum` statt unter `/dev/vinum`. Seit FreeBSD 6.x ist die alte Vinum-Implementierung nicht mehr im Basissystem enthalten.

22.2. Ihre Platten sind zu klein.

Festplatten werden zwar immer größer, parallel dazu steigt aber auch die Größe der zu speichernden Daten an. Es kann also nach wie vor vorkommen, dass Sie ein Dateisystem benötigen, welches die Größe Ihrer Platten übersteigt. Zwar ist dieses Problem nicht mehr so akut wie noch vor einigen Jahren, aber es existiert nach wie vor. Einige Systeme lösen dieses Problem durch die Erzeugung eines abstrakten Gerätes, das seine Daten auf mehreren Platten speichert.

22.3. Mögliche Engpässe

Moderne Systeme müssen häufig parallel auf Daten zugreifen. Große FTP- und HTTP-Server können beispielsweise Tausende von parallelen Sitzungen verwalten und haben mehrere 100 MBit/s-Verbindungen zur Außenwelt. Diese Bandbreite überschreitet die durchschnittliche Transferrate der meisten Platten bei weitem.

Aktuelle Plattenlaufwerke können Daten mit bis zu 70 MB/s sequentiell übertragen, wobei dieser Wert in einer Umgebung, in der viele unabhängige Prozesse auf eine gemeinsame Platte zugreifen, die jeweils nur einen Bruchteil dieses Wertes erreichen, von geringer Aussagekraft ist. In solchen Fällen ist es interessanter, das Problem vom Blickwinkel des Platten-Subsystems aus zu betrachten. Der wichtigste Parameter ist dabei die Last, die eine Übertragung auf dem Subsystem verursacht. Unter Last versteht man dabei die Zeit, in der die Platte mit der Übertragung der Daten beschäftigt ist.

Bei jedem Plattenzugriff muss das Laufwerk zuerst die Köpfe positionieren und auf den ersten Sektor warten, bis er den Lesekopf passiert. Dann wird die Übertragung gestartet. Diese Aktionen können als atomar betrachtet werden, da es keinen Sinn macht, diese zu unterbrechen.

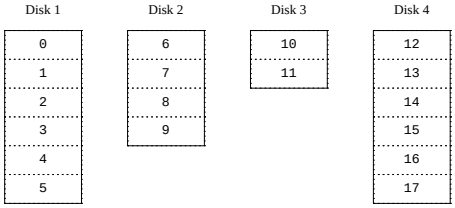
Nehmen wir beispielsweise an, dass wir 10 kB transferieren wollen. Aktuelle hochperformante Platten können die Köpfe im Durchschnitt in 3,5 ms positionieren und drehen sich mit maximal 15.000 U/min. Daher beträgt die durchschnittliche Rotationslatenz (eine halbe Umdrehung) 2 ms. Bei einer Transferrate von 70 MB/s dauert die eigentliche Übertragung von 10 kB etwa 150 μs, fast nichts im Vergleich zur Positionierungszeit. In einem solchen Fall beträgt die effektive Transferrate nur etwas mehr als 1 MB/s. Die Tranferrate ist also stark von der Größe der zu tranferierenden Daten abhängig.

Die traditionelle und offensichtliche Lösung zur Beseitigung dieses Flaschenhalses sind “mehr Spindeln”. Statt einer einzigen großen Platte werden mehrere kleinere Platten mit demselben Gesamtspeicherplatz benutzt. Jede Platte ist in der Lage, unabhängig zu positionieren und zu transferieren, weshalb der effektive Durchsatz um einen Faktor nahe der Zahl der eingesetzten Platten steigt.

Obwohl die Platten Daten parallel transferieren können, ist es nicht möglich, Anfragen gleichmäßig auf die einzelnen Platten zu verteilen. Daher wird die Last auf bestimmten Laufwerken immer höher sein als auf anderen Laufwerken. Daraus ergibt sich auch, dass die exakte Verbesserung des Datendurchsatzes immer kleiner ist als die Anzahl der involvierten Platten.

Die gleichmäßige Verteilung der Last auf die einzelnen Platten ist stark abhängig von der Art, wie die Daten auf die Laufwerke aufgeteilt werden. In den folgenden Ausführungen wird eine Platte als eine große Anzahl von Datensektoren dargestellt, die durch Zahlen adressierbar sind (ähnlich den Seiten eines Buches). Die naheliegendste Methode ist es, die virtuelle Platte (wieder analog den Seiten eines Buches) in Gruppen aufeinanderfolgender Sektoren zu unterteilen, die jeweils der Größe der einzelnen physischen Platten entsprechen. Diese Vorgehensweise wird als *Konkatenation* bezeichnet und hat den Vorteil, dass die Platten keine spezielle Größenbeziehung haben müssen. Sie funktioniert gut, solange der Zugriff gleichmäßig auf den Adressraum der virtuellen Platte verteilt wird. Wenn sich der Zugriff allerdings auf einen kleinen Bereich konzentriert, ist die Verbesserung vernachlässigbar klein. Abbildung 22-1 verdeutlicht die Verteilung der Speichereinheiten in einer konkatenierten Anordnung.

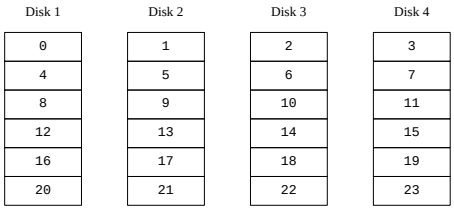
Abbildung 22-1. Konkatenierte Anordnung



Ein alternatives Mapping unterteilt den Adressraum in kleinere, gleich große Komponenten und speichert diese sequentiell auf verschiedenen Geräten. Zum Beispiel werden die ersten 256 Sektoren auf der ersten Platte, die nächsten 256 Sektoren auf der zweiten Platte gespeichert und so weiter. Nachdem die letzte Platte beschrieben wurde, wird dieser Vorgang solange wiederholt, bis die Platten voll sind. Dieses Mapping nennt man *Striping* oder RAID-0. [1]

Striping erfordert einen etwas größeren Aufwand, um die Daten zu lokalisieren, und kann zusätzliche E/A-Last verursachen, wenn eine Übertragung über mehrere Platten verteilt ist. Auf der anderen Seite erlaubt es aber eine gleichmäßigere Verteilung der Last auf die einzelnen Platten. Abbildung 22-2 veranschaulicht die Abfolge, in der Speichereinheiten in einer striped-Anordnung alloziert werden.

Abbildung 22-2. Striped-Anordnung



22.4. Datenintegrität

Das dritte Problem, welches aktuelle Platten haben, ist ihre Unzuverlässigkeit. Obwohl sich die Zuverlässigkeit von Festplatten in den letzten Jahren stark verbessert hat, handelt es sich bei ihnen nach wie vor um die Komponente eines Servers, die am ehesten ausfällt. Fällt eine Festplatte aus, können die Folgen katastrophal sein: Es kann Tage dauern, bis eine Platte ersetzt und alle Daten wiederhergestellt sind.

Die traditionelle Art, dieses Problem anzugehen, war es, Daten zu *spiegeln*, also zwei Kopien der Daten auf getrennten Platten zu verwahren. Diese Technik wird auch als RAID Level 1 oder RAID-1 bezeichnet. Jeder

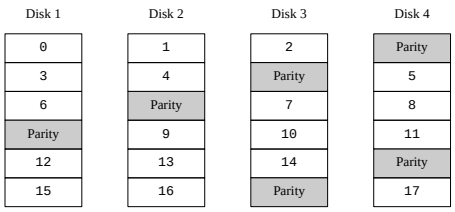
Schreibzugriff findet auf beiden Datenträgern statt. Ein Lesezugriff kann daher von beiden Laufwerken erfolgen, sodass beim Ausfall eines Laufwerks die Daten immer noch auf dem anderen Laufwerk verfügbar sind.

Spiegeln verursacht allerdings zwei Probleme:

- Es verursacht höhere Kosten, da doppelt so viel Plattenspeicher wie bei einer nicht-redundanten Lösung benötigt wird.
- Die Gesamtleistung des Systems sinkt, da Schreibzugriffe auf beiden Laufwerken ausgeführt werden müssen, daher wird im Vergleich zu einem nicht gespiegelten Datenträger die doppelte Bandbreite benötigt. Lesezugriffe hingegen sind davon nicht betroffen, es sieht sogar so aus, als würden diese schneller ausgeführt.

Eine alternative Lösung ist *Parity*, das in den RAID-Leveln 2, 3, 4 und 5 implementiert ist. Von diesen ist RAID-5 der interessanteste. So wie in VINUM implementiert, ist es eine Variante einer gestripten Anordung, welche einen Block jedes Stripes als Paritätsblock für einen der anderen Blöcke verwendet. Wie in RAID-5 vorgeschrieben, ist die Position dieses Paritätsblockes auf jedem Stripe unterschiedlich. Die Nummern in den Datenblöcken geben die relativen Blocknummern an.

Abbildung 22-3. RAID-5 Aufbau



Im Vergleich zur Spiegelung hat RAID-5 den Vorteil, dass es signifikant weniger Speicherplatz benötigt. Lesezugriffe sind vergleichbar schnell mit jenen bei einem Striped-Aufbau, aber Schreibzugriffe sind deutlich langsamer (etwa 25% der Lesegeschwindigkeit). Wenn eine Platte ausfällt, kann das Array in einem "schwachen" Modus weiterarbeiten: Ein Lesezugriff auf eine der übrigen erreichbaren Platten wird normal ausgeführt, ein Lesezugriff auf die ausgefallene Platte muss aber zunächst mit dem zugehörigen Block aller verbleibender Platten rückberechnet werden.

22.5. Vinum-Objekte

Um die in den vorigen Abschnitte besprochenen Probleme zu lösen, verwendet Vinum eine vierstufige Objekthierarchie:

- Das auffälligste Objekt ist die virtuelle Platte, die *Volume* genannt wird. Volumes haben im Wesentlichen die gleichen Eigenschaften wie ein UNIX-Laufwerk, obwohl es ein paar kleine Unterschiede gibt. So existieren für Volumes beispielsweise keine Größenbeschränkungen.

- Volumes bestehen aus einem oder mehreren *Plexus*, von denen jeder den gesamten Adressraum eines Datenträgers repräsentiert. Diese Hierarchieebene ist für die benötigte Redundanz der Daten erforderlich. Stellen Sie sich die Plexus als eigenständige Platten in einem gespiegelten Array vor, von denen jede die gleichen Daten enthält.
- Da Vinum im UNIX-Plattenspeicher-Framework arbeitet, wäre es möglich, als Grundbaustein für Multiplatten-Plexus UNIX-Partitionen zu verwenden. In der Praxis ist dieser Ansatz aber zu unflexibel, da UNIX-Platten nur eine begrenzte Anzahl von Partitionen haben können. Daher unterteilt Vinum stattdessen eine einzige UNIX-Partition (die *Platte*) in zusammenhängende Bereiche, die als *Subdisks* bezeichnet werden und als Grundbausteine für einen Plexus benutzt werden.
- Subdisks befinden sich auf Vinum-*Platten*, eigentlich UNIX-Partitionen. Vinum-Platten können eine beliebige Anzahl von Subdisks haben und den gesamten Speicher der Platte mit Ausnahme eines kleinen Bereiches am Anfang der Platte (welcher zur Speicherung von Konfigurations- und Statusinformationen verwenden wird) verwenden.

Der folgende Abschnitt beschreibt, wie diese Objekte die von Vinum benötigten Funktionen zur Verfügung stellen.

22.5.1. Überlegungungen zur Größe eines Volumes

Plexus können mehrere Subdisks beinhalten, die über alle Platten der Vinum-Konfiguration verteilt sind. Daraus folgt, dass die Größe einer Platte nicht die Größe eines Plexus (und damit eines Volumes) limitiert.

22.5.2. Redundante Datenspeicherung

Vinum implementiert die Datenspiegelung, indem es ein Volume auf mehrere Plexus verteilt. Jeder Plexus ist dabei die Repräsentation der Daten eines Volumes. Ein Volume kann aus bis zu acht Plexus bestehen.

Obwohl ein Plexus die gesamten Daten eines Volumes repräsentiert, ist es möglich, dass Teile der Repräsentation physisch fehlen, entweder aufgrund des Designs (etwa durch nicht definierte Subdisks für Teile des Plexus) oder durch Zufall (als ein Ergebnis eines Plattenfehlers). Solange wenigstens ein Plexus die gesamten Daten für den kompletten Adressbereich des Volumes zur Verfügung stellen kann, ist das Volume voll funktionsfähig.

22.5.3. Überlegungen zur Leistung

Sowohl Konkatenation als auch Striping werden von Vinum auf der Plexus-Ebene realisiert:

- Ein *konkatenierter Plexus* benutzt abwechselnd den Adressraum jeder Subdisk.
- Ein *gestripter Plexus* striped die Daten über jede Subdisk. Die Subdisks müssen alle die gleiche Größe haben, und es muss mindestens zwei Subdisks in Reihenfolge geben, um ihn von einem konkatenierten Plexus unterscheiden zu können.

22.5.4. Wie ist ein Plexus aufgebaut?

Die Version von Vinum, welche von FreeBSD-9.1 bereitgestellt wird, implementiert zwei Arten von Plexus:

- Konkatenierte Plexus sind die flexibelsten: Sie können aus einer beliebigen Anzahl von Subdisks unterschiedlicher Größe bestehen. Der Plexus kann erweitert werden, indem man zusätzliche Subdisks hinzufügt. Sie brauchen weniger CPU-Zeit als gestripte Plexus, obwohl der Unterschied des CPU-Overheads nicht messbar ist. Auf der

anderen Seite sind sie aber sehr anfällig für das Entstehen von "hot spots", wobei eine Platte sehr aktiv ist, andere hingegen nahezu ungenutzt sind.

- Der größte Vorteil eines gestripten Plexus (RAID-0) ist die Verringerung von "hot spots". Dies wird durch die Auswahl eines Stripes optimaler Größe (etwa 256 kB) erreicht, wodurch die Last gleichmäßig auf die Platten verteilt werden kann. Nachteile dieser Vorgehensweise sind ein (geringfügig) komplexerer Code sowie einige Restriktionen für die Subdisks: Diese müssen alle die gleiche Größe haben, und das Erweitern eines Plexus durch das Hinzufügen neuer Subdisks ist so kompliziert, dass es von Vinum derzeit nicht unterstützt wird. Vinum fordert noch eine weitere triviale Beschränkung: Ein gestripter Plexus muss aus mindestens zwei Subdisks bestehen, da er ansonsten nicht von einem konkatenierten Plexus unterscheidbar ist.

Tabelle 22-1 fasst die Vor- und Nachteile jedes Plexus-Aufbaus zusammen.

Tabelle 22-1. Vinum-Plexus - Aufbau

Plexus-Typ	Minimum an Subdisks?	Kann Subdisks hinzufügen?	Müssen die gleiche Größe haben	Applikation
konkateniert	1	ja	nein	Großer Datenspeicher mit maximaler Platzierungsflexibilität und moderater Leistung
gestriped	2	nein	ja	Hohe Leistung in Kombination mit gleichzeitigem Zugriff

22.6. Einige Beispiele

Vinum verwaltet eine *Konfigurationsdatenbank* für alle einem individuellen System bekannten Objekte. Zu Beginn erzeugt ein Benutzer mit gvinum(8) eine Konfigurationsdatenbank aus einer oder mehreren Konfigurationsdateien. Vinum speichert danach eine Kopie der Konfigurationsdatenbank in jedem von ihm kontrollierten Slice (von Vinum als *Device* bezeichnet). Da die Datenbank bei jedem Statuswechsel aktualisiert wird, kann nach einem Neustart der Status jedes Vinum-Objekts exakt wiederhergestellt werden.

22.6.1. Die Konfigurationsdatei

Die Konfigurationsdatei beschreibt individuelle Vinum-Objekte. Die Beschreibung eines einfachen Volumes könnte beispielsweise so aussehen:

```
drive a device /dev/da3h
volume myvol
  plex org concat
    sd length 512m drive a
```

Diese Datei beschreibt vier Vinum-Objekte:

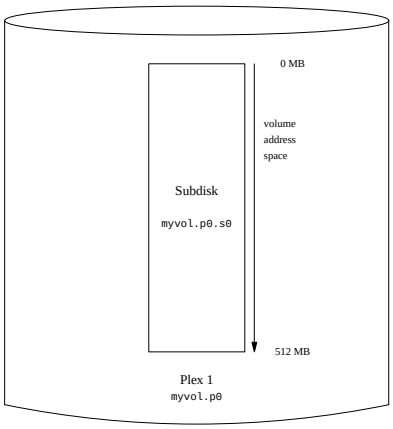
- Die *drive*-Zeile beschreibt eine Plattenpartition (*drive*) sowie ihre Position in Bezug auf die darunter liegende Hardware. Die Partition hat dabei den symbolischen Namen *a*. Diese Trennung von symbolischen Namen und Gerätenamen erlaubt es, die Position von Platten zu ändern, ohne dass es zu Problemen kommt.
- Die *volume*-Zeile beschreibt ein Volume. Dafür wird nur ein einziges Attribut, der Name des Volumes, benötigt. In unserem Fall hat das Volume den Namen *myvol*.
- Die *plex*-Zeile definiert einen Plexus. Auch hier wird nur ein Parameter, und zwar die Art des Aufbau, benötigt (in unserem Fall *concat*). Es wird kein Name benötigt, das System generiert automatisch einen Namen aus dem Volume-Namen und dem Suffix *.px* wobei *x* die Nummer des Plexus innerhalb des Volumes angibt. So wird dieser Plexus den Namen *myvol.p0* erhalten.
- Die *sd*-Zeile beschreibt eine Subdisk. Um eine Subdisk einzurichten, müssen Sie zumindest den Namen der Platte, auf der Sie die Subdisk anlegen wollen, sowie die Größe der Subdisk angeben. Analog zur Definition eines Plexus wird auch hier kein Name benötigt: Das System weist automatisch Namen zu, die aus dem Namen des Plexus und dem Suffix *.sx* gebildet werden, wobei *x* die Nummer der Subdisk innerhalb des Plexus ist. Folglich gibt Vinum dieser Subdisk den Namen *myvol.p0.s0*.

Nach dem Verarbeiten dieser Datei erzeugt gvinum(8) die folgende Ausgabe:

```
# gvinum -> create config1
Configuration summary
Drives:         1 (4 configured)
Volumes:        1 (4 configured)
Plexes:         1 (8 configured)
Subdisks:       1 (16 configured)

 D a                    State: up       Device /dev/da3h        Avail: 2061/2573 MB (80%)

 V myvol                State: up       Plexes:       1 Size:        512 MB

 P myvol.p0           C State: up       Subdisks:     1 Size:        512 MB

 S myvol.p0.s0          State: up       PO:        0  B Size:        512 MB
```

Diese Ausgabe entspricht dem verkürzten Ausgabeformat von gvinum(8) und wird in Abbildung 22-4 grafisch dargestellt.

Abbildung 22-4. Ein einfaches Vinum-Volume



Dieses und die folgenden Beispiele zeigen jeweils ein Volume, welches die Plexus enthält, die wiederum die Subdisk enthalten. In diesem trivialen Beispiel enthält das Volume nur einen Plexus, der wiederum nur aus einer Subdisk besteht.

Eine solche Konfiguration hätte allerdings keinen Vorteil gegenüber einer konventionellen Plattenpartion. Das Volume enthält nur einen einzigen Plexus, daher gibt es keine redundante Datenspeicherung. Da der Plexus außerdem nur eine einzige Subdisk enthält, unterscheidet sich auch die Speicherzuweisung nicht von der einer konventionellen Plattenpartition. Die folgenden Abschnitte beschreiben daher verschiedene interessantere Konfigurationen.

22.6.2. Verbesserte Ausfallsicherheit durch Spiegelung

Die Ausfallsicherheit eines Volumes kann durch Spiegelung der Daten erhöht werden. Beim Anlegen eines gespiegelten Volumes ist es wichtig, die Subdisks jedes Plexus auf verschiedene Platten zu verteilen, damit ein Plattenausfall nicht beide Plexus unbrauchbar macht. Die folgende Konfiguration spiegelt ein Volume:

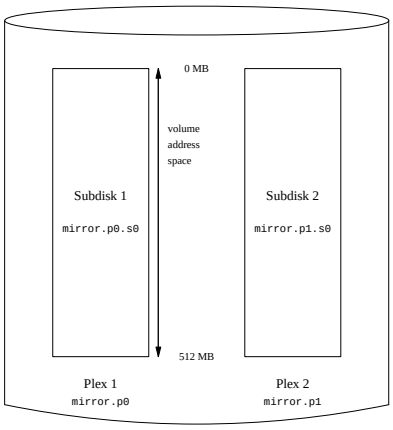
```
drive b device /dev/da4h
volume mirror
plex org concat
  sd length 512m drive a
   plex org concat
     sd length 512m drive b
```

Bei diesem Beispiel war es nicht nötig, noch einmal eine Platte *a* zu spezifizieren, da Vinum die Übersicht über alle Objekte und seine Konfigurationsdatenbank behält. Nach dem Abarbeiten dieser Definition sieht die Konfiguration wie folgt aus:

```
Drives:         2 (4 configured)
Volumes:        2 (4 configured)
Plexes:         3 (8 configured)
Subdisks:       3 (16 configured)

D a                     State: up       Device /dev/da3h        Avail: 1549/2573 MB (60%)
D b                     State: up       Device /dev/da4h        Avail: 2061/2573 MB (80%)

V myvol                 State: up       Plexes:       1 Size:        512 MB
V mirror                State: up       Plexes:       2 Size:        512 MB

P myvol.p0            C State: up       Subdisks:     1 Size:        512 MB
P mirror.p0           C State: up       Subdisks:     1 Size:        512 MB
P mirror.p1           C State: initializing     Subdisks:     1 Size:        512 MB

S myvol.p0.s0           State: up       PO:        0  B Size:        512 MB
S mirror.p0.s0          State: up       PO:        0  B Size:        512 MB
S mirror.p1.s0          State: empty    PO:        0  B Size:        512 MB
```

Abbildung 22-5 stellt diese Struktur grafisch dar.

Abbildung 22-5. Ein gespiegeltes Vinum Volume



In diesem Beispiel enthält jeder Plexus die vollen 512 MB des Adressraumes. Wie im vorangegangenen Beispiel enthält jeder Plexus nur eine Subdisk.

22.6.3. Die Leistung optimieren

Das gespiegelte Volume des letzten Beispieles ist resistenter gegenüber Fehlern als ein ungespiegeltes Volume, seine Leistung ist hingegen schlechter, da jeder Schreibzugriff auf das Volume einen Schreibzugriff auf beide Platten erfordert und damit mehr der insgesamt verfügbaren Datentransferrate benötigt. Steht also die optimale Leistung im Vordergrund, muss anders vorgegangen werden: Statt alle Daten zu spiegeln, werden die Daten über so viele Platten wie möglich gestriped. Die folgende Konfiguration zeigt ein Volume mit einem über vier Platten hinwegreichenden Plexus:

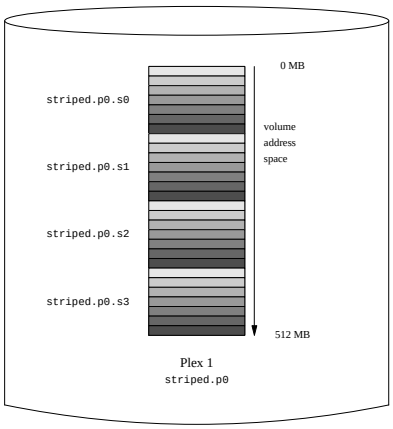
```
drive c device /dev/da5h
drive d device /dev/da6h
volume stripe
plex org striped 512k
  sd length 128m drive a
  sd length 128m drive b
  sd length 128m drive c
  sd length 128m drive d
```

Wie zuvor ist es nicht nötig, die Platten zu definieren, die Vinum schon bekannt sind. Nach dem Abarbeiten dieser Definition sieht die Konfiguration wie folgt aus:

```
    Drives:         4 (4 configured)
    Volumes:        3 (4 configured)
    Plexes:         4 (8 configured)
    Subdisks:       7 (16 configured)

D a                     State: up       Device /dev/da3h        Avail: 1421/2573 MB (55%)
D b                     State: up       Device /dev/da4h        Avail: 1933/2573 MB (75%)
```

```
D c                     State: up       Device /dev/da5h        Avail: 2445/2573 MB (95%)
D d                     State: up       Device /dev/da6h        Avail: 2445/2573 MB (95%)

V myvol                 State: up       Plexes:       1 Size:        512 MB
V mirror                State: up       Plexes:       2 Size:        512 MB
V striped               State: up       Plexes:       1 Size:        512 MB

P myvol.p0            C State: up       Subdisks:     1 Size:        512 MB
P mirror.p0           C State: up       Subdisks:     1 Size:        512 MB
P mirror.p1           C State: initializing     Subdisks:     1 Size:        512 MB
P striped.p1            State: up       Subdisks:     1 Size:        512 MB

S myvol.p0.s0           State: up       PO:        0  B Size:        512 MB
S mirror.p0.s0          State: up       PO:        0  B Size:        512 MB
S mirror.p1.s0          State: empty    PO:        0  B Size:        512 MB
S striped.p0.s0         State: up       PO:        0  B Size:        128 MB
S striped.p0.s1         State: up       PO:      512 kB Size:        128 MB
S striped.p0.s2         State: up       PO:     1024 kB Size:        128 MB
S striped.p0.s3         State: up       PO:     1536 kB Size:        128 MB
```

Abbildung 22-6. Ein Striped Vinum Volume



Dieses Volume wird in Abbildung 22-6 dargestellt. Die Schattierung der Stripes zeigt die Position innerhalb des Plexus-Adressraumes an. Die hellsten Stripes kommen zuerst, die dunkelsten zuletzt.

22.6.4. Ausfallsicherheit und Leistung

Mit entsprechender Hardware ist es möglich, Volumes zu bauen, welche gegenüber Standard-UNIX-Partitionen beides, nämlich erhöhte Ausfallsicherheit und erhöhte Leistung, aufweisen können. Eine typische Konfigurationsdatei könnte etwa so aussehen:

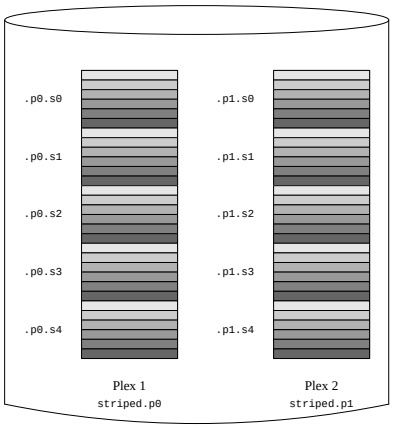
```
 volume raid10
plex org striped 512k
  sd length 102480k drive a
```

```
    sd length 102480k drive b
    sd length 102480k drive c
    sd length 102480k drive d
    sd length 102480k drive e
  plex org striped 512k
    sd length 102480k drive c
    sd length 102480k drive d
    sd length 102480k drive e
    sd length 102480k drive a
    sd length 102480k drive b
```

Die Subdisks des zweiten Plexus sind gegenüber denen des ersten Plexus um zwei Platten verschoben. Dadurch wird sichergestellt, dass Schreibzugriffe nicht auf den gleichen Subdisks auftreten, auch wenn eine Übertragung über zwei Platten geht.

Abbildung 22-7 veranschaulicht die Struktur dieses Volumes.

Abbildung 22-7. Ein gespiegeltes, Striped Vinum Volume



22.7. Objektbenennung

Wie oben beschrieben, weist Vinum den Plexus und Subdisks Standardnamen zu, wenngleich diese überschrieben werden können. Das Überschreiben dieser Standardnamen wird allerdings nicht empfohlen. Erfahrungen mit dem VERITAS Volume Manager (der eine willkürliche Benennung von Objekten erlaubt) haben gezeigt, dass diese Flexibilität keinen entscheidenden Vorteil bringt und zudem Verwirrung stiften kann.

Namen dürfen zwar alle nichtleeren Zeichen enthalten, es ist aber sinnvoll, nur Buchstaben, Ziffern und den Unterstrich zu verwenden. Die Namen von Volumes, Plexus und Subdisks können bis zu 64 Zeichen lang sein, die Namen von Platten dürfen hingegen nur bis zu 32 Zeichen lang sein.

Vinum-Objekten werden Gerätedateien in der `/dev/gvinum`-Hierarchie zugewiesen. Die weiter oben dargestellte Konfiguration würde Vinum dazu veranlassen, die folgenden Gerätedateien zu erstellen:

- Geräte-Einträge für jedes Volume. Dieses sind die Hauptgeräte, die von Vinum benutzt werden. Somit würde die Konfiguration von oben folgende Geräte beinhalten: `/dev/gvinum/myvol`, `/dev/gvinum/mirror`, `/dev/gvinum/striped`, `/dev/gvinum/raid5` sowie `/dev/gvinum/raid10`.
- Alle Volumes bekommen direkte Einträge unter `/dev/gvinum/`.
- Die Verzeichnisse `/dev/gvinum/plex` und `/dev/gvinum/sd`, die Gerätedateien für jeden Plexus sowie jede Subdisk enthalten.

Stellen Sie sich folgende Konfigurationsdatei vor:

```
    drive drive1 device /dev/sd1h
    drive drive2 device /dev/sd2h
    drive drive3 device /dev/sd3h
    drive drive4 device /dev/sd4h
volume s64 setupstate
  plex org striped 64k
    sd length 100m drive drive1
    sd length 100m drive drive2
    sd length 100m drive drive3
    sd length 100m drive drive4
```

Nach Abarbeitung dieser Datei erstellt gvinum(8) die folgende Struktur unter `/dev/gvinum`:

```
    drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 plex
    crwxr-xr--  1 root  wheel   91,   2 Apr 13 16:46 s64
    drwxr-xr-x  2 root  wheel       512 Apr 13 16:46 sd

/dev/vinum/plex:
total 0
crwxr-xr--  1 root  wheel   25, 0x10000002 Apr 13 16:46 s64.p0

/dev/vinum/sd:
total 0
crwxr-xr--  1 root  wheel   91, 0x20000002 Apr 13 16:46 s64.p0.s0
crwxr-xr--  1 root  wheel   91, 0x20100002 Apr 13 16:46 s64.p0.s1
crwxr-xr--  1 root  wheel   91, 0x20200002 Apr 13 16:46 s64.p0.s2
crwxr-xr--  1 root  wheel   91, 0x20300002 Apr 13 16:46 s64.p0.s3
```

Es wird empfohlen, für Plexus und Subdisks keine eigenen Namen zu vergeben. Dies gilt aber nicht für Vinum-Platten. Durch die Benennung von Vinum-Platten wird es erst möglich, eine Platte an einen anderen Ort zu verschieben und sie trotzdem noch automatisch erkennen zu lassen. Plattennamen können bis zu 32 Zeichen lang sein.

22.7.1. Dateisysteme erstellen

Volumes erscheinen (mit einer Ausnahme) dem System nicht anders als Platten. Anders als UNIX-Platten partitioniert Vinum seine Volumes nicht, weshalb diese auch keine Partitionstabellen haben. Dies wiederum hat Modifikationen an einigen Platten-Tools, insbesondere newfs(8), nötig gemacht, welche bis dahin den letzten Buchstaben eines Vinum-Volume-Namen als Partitionsbezeichner identifiziert haben. Zum Beispiel könnte eine

Platte einen Namen wie `/dev/ad0a` oder `/dev/da2h` haben. Diese Namen bedeuten, dass es sich um die erste Partition (`a`) der ersten (0) IDE-Platte (`ad`) und respektive die achte Partition (`h`) der dritten (2) SCSI-Platte (`da`) handelt. Im Vergleich dazu könnte ein Vinum-Volume beispielsweise `/dev/gvinum/concat` heißen, ein Name, der in keiner Beziehung mit einem Partitionsnamen steht.

Um nun ein Dateisystem auf diesem Volume anzulegen, benutzen Sie newfs(8):

```
# newfs /dev/gvinum/concat
```

22.8. Vinum konfigurieren

Der `GENERIC`-Kernel enthät kein Vinum. Es ist zwar möglich, einen speziellen Kernel zu bauen, der Vinum beinhaltet, empfohlen wird aber, Vinum als ein Kernelmodul (über kld) zu laden. Dazu müssen Sie nicht einmal kldload(8) benutzen, da beim Start von gvinum(8) automatisch überprüft wird, ob das Modul bereits geladen wurde. Falls das Modul noch nicht geladen wurde, wird es daraufhin geladen.

22.8.1. Inbetriebnahme

Vinum speichert seine Konfigurationsinformationen auf den Platten-Slices im Wesentlichen genauso ab wie in den Konfigurationsdateien. Beim Lesen der Konfigurationsdatenbank erkennt Vinum eine Anzahl von Schlüsselwörtern, die in den Konfigurationsdateien nicht erlaubt sind. Zum Beispiel könnte eine Platten-Konfiguration den folgenden Text enthalten:

```
volume myvol state up
volume bigraid state down
plex name myvol.p0 state up org concat vol myvol
plex name myvol.p1 state up org concat vol myvol
plex name myvol.p2 state init org striped 512b vol myvol
plex name bigraid.p0 state initializing org raid5 512b vol bigraid
sd name myvol.p0.s0 drive a plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p0.s1 drive b plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p1.s0 drive c plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p1.s1 drive d plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p2.s0 drive a plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 0b
sd name myvol.p2.s1 drive b plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 524288b
sd name myvol.p2.s2 drive c plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1048576b
sd name myvol.p2.s3 drive d plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1572864b
sd name bigraid.p0.s0 drive a plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 0b
sd name bigraid.p0.s1 drive b plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 4194304b
sd name bigraid.p0.s2 drive c plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 8388608b
sd name bigraid.p0.s3 drive d plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 12582912b
sd name bigraid.p0.s4 drive e plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 16777216b
```

Die offensichtlichen Unterschiede sind hier die Anwesenheit von Informationen über explizite Speicherorte und Benennungen (beides ist zwar erlaubt, aber es wird dem Benutzer davon abgeraten, es zu benutzen) und Informationen über die Zustände (welche für den Benutzer nicht zur Verfügung stehen). Vinum speichert keine Informationen über Platten in den Konfigurationsinformationen, es findet die Platten durch Scannen nach Vinum-Markierungen auf den eingerichteten Laufwerken. Dies ermöglicht es, Vinum-Platten auch dann noch korrekt zu identifizieren, wenn sie schon andere UNIX-Platten-IDs zugewiesen bekommen haben.

22.8.1.1. Automatische Inbetriebnahme

> **Anmerkung:** *Gvinum* unterstützt eine automatische Inbetriebnahme immer, wenn das Kernelmodul über loader.conf(5) geladen ist. Um das *Gvinum* Modul beim Hochfahren des Systems zu laden, fügen Sie die Zeile `geom_vinum_load="YES"` in `/boot/loader.conf` ein.

Beim starten von Vinum durch den Befehl `vinum start` liest Vinum die Konfigurationsdatenbank von einer der Vinum-Platten. Unter normalen Umständen enthält jede Platte eine identische Kopie der Konfigurationsdatenbank, so dass es keine Rolle spielt, von welcher der Platten diese eingelesen wird. Nach einem Plattencrash muss Vinum allerdings zunächst feststellen, welche der Platten zuletzt aktualisiert wurde und dann die Konfiguration von dieser Platte lesen. Danach werden (falls nötig) die Konfigurationen der "alten" Platten aktualisiert.

22.9. Vinum für das Root-Dateisystem benutzen

Auf einem System, das mit Hilfe von Vinum vollgespiegelte Dateisysteme hat, ist es wünschenswert, auch das Root-Dateisystem zu spiegeln. Solch eine Konfiguration ist allerdings weniger trivial als das Spiegeln eines gewöhnlichen Dateisystems, weil:

- Das Root-Dateisystem in einer sehr frühen Phase des Bootvorgangs verfügbar sein muss, und damit auch die Vinum-Infrastruktur.
- Das Volume, welches das Root-Dateisystem enthält, auch den Bootstrap und den Kernel enthält, die wiederum nur mit den systemeigenen Tools (zum Beispiel dem BIOS bei handelsüblichen PCs) gelesen werden können und meist nicht dazu gebracht werden können, Vinum zu verstehen.

Im folgenden Abschnitt wird der Begriff “Root-Volume” benutzt, um das Vinum-Volume zu beschreiben, welches das Root-Dateisystem enthält. Es ist eine gute Idee, für dieses Volume den Namen `"root"` zu benutzen, aber es ist in keiner Weise technisch nötig (Das folgende Beispiel geht allerdings davon aus, dass dies der Fall ist.).

22.9.1. Vinum für das Root-Dateisystem rechtzeitig starten

Damit dies gelingt, müssen Sie folgende Aufgaben erledigen:

- Vinum muss zum Zeitpunkt des Bootvorganges im Kernel zur Verfügung stehen. Deswegen ist die Methode zum Start von Vinum, die in Abschnitt 22.8.1.1 beschrieben wird, für diese Aufgabe nicht geeignet. Also muss auch der `start_vinum`-Parameter eigentlich *nicht* gesetzt werden, wenn man das folgende Setup einrichtet. Die erste Möglichkeit wäre es, Vinum statisch in den Kernel zu kompilieren, so dass es ständig verfügbar ist, was aber in der Regel nicht erwünscht ist. Ebenso gibt es die Möglichkeit `/boot/loader` (Abschnitt 13.3.3) das Vinum-Kernelmodul früh genug laden zu lassen (und zwar noch bevor der Kernel gestartet wird). Dies kann bewerkstelligt werden, indem die Zeile

  ```
  geom_vinum_load="YES"
  ```

  in die Datei `/boot/loader.conf` eingetragen wird.
- Für *Gvinum* ist das oben beschriebene Prozedere alles, was Sie tun müssen, da der gesamte Startvorgang automatisch erledigt wird, sobald das Kernelmodul geladen wurde.

22.9.2. Ein Vinum-basiertes Root-Volume dem Bootstrap verfügbar machen

Da der aktuelle FreeBSD-Bootstrap nur 7,5 KB Code enthält und schon ohne Vinum die Aufgabe hat, bestimmte Dateien (wie `/boot/loader`) von einem UFS-Dateisystem zu lesen, ist es schier unmöglich, ihm auch noch die

Interna von Vinum beizubringen, damit er die Vinum-Konfigurationsdaten auslesen und die Elemente eines Boot-Volumes selbst herausfinden könnte. Daher sind ein paar Tricks nötig, um dem Bootstrap-Code die Illusion einer Standard-`"a"`-Partition mit einem Root-Dateisystem vorzugaukeln.

Damit dies überhaupt möglich wird, müssen die folgenden Bedingungen für das Root-Dateisystem erfüllt sein:

- Das Root-Volume darf weder gestriped noch RAID-5 sein.
- Das Root-Volume darf nicht mehr als eine konkatenierte Subdisk pro Plexus enthalten.

Beachten Sie, dass es möglich und wünschenswert ist, mehrere Plexus zu haben, von denen jeder eine Kopie des Root-Dateisystems enthält. Der Bootstrap-Prozess wird hingegen nur einen dieser Plexus benutzen, um den Bootstrap und alle Dateien zu finden, bis der Kernel letztendlich das Root-Dateisystem selbst laden wird. Jede einzelne Subdisk innerhalb dieser Plexus wird dann ihre eigene Illusion der Partition `"a"` brauchen, damit das entsprechende Gerät bootbar wird. Es ist nicht unbedingt notwendig, dass sich jede dieser gefälschten `"a"`-Partitionen auf seinem Gerät an einem Ort befindet, der um den selben Wert verschoben ist wie auf den anderen Geräten, die Plexus des Root-Dateisystems enthalten. Um Unklarheiten zu verhindern, ist es jedoch eine gute Idee, die Vinum-Volumes so zu erstellen, dass die gespiegelten Geräte symmetrisch sind.

Damit diese `"a"`-Partitionen eingerichtet werden können, muss für alle Geräte, die Teil des Root-Dateisystems sind, folgendes getan werden:

1. Der Ort (Verschiebung vom Beginn des Gerätes) und die Größe der Subdisk, die Teil des Root-Volumes ist, muss untersucht werden:

   ```
   # gvinum l -rv root
   ```

   Beachten Sie, dass Vinum-Verschiebungen und -Größen in Bytes gemessen werden. Sie müssen deshalb durch 512 geteilt werden, um die Blockanzahl zu erhalten, wie sie das `bsdlabel`-Kommando verwendet.

2. Führen Sie den Befehl

   ```
   # bsdlabel -e devname
   ```

   für jedes Gerät, dass am Root-Volume beteiligt ist, aus. *`devname`* muss entweder der Name der Platte (wie `da0`), im Falle einer Platte ohne Slice-Tabelle oder der Name des Slices (wie `ad0s1`) sein.

   Wenn es schon eine `"a"`-Partition auf dem Gerät (in der Regel wahrscheinlich ein Prä-Vinum-Root-Dateisystem) gibt, sollte diese umbenannt werden, damit sie weiterhin verfügbar bleibt (nur für den Fall). Sie wird aber nicht länger benutzt, um das System zu starten. Beachten Sie aber, dass aktive Partitionen (wie ein gemountetes Root-Dateisystem) nicht umbenannt werden können, sodass Sie entweder von einem “Fixit”-Medium booten müssen, oder aber mittels eines zweistufigen Prozesses (sofern Sie in einer gespiegelten Umgebung arbeiten) zuerst die Platte ändern, von der gerade nicht gebootet wurde.

   Nun muss die Verschiebung der Vinum-Partition (sofern vorhanden) auf diesem Gerät mit der Veschiebung der entsprechenden Root-Volume-Subdisk addiert werden. Das Resultat wird der `"offset"`-Wert für die neue `"a"`-Partition. Der `"size"`-Wert für diese Partition kann entsprechend der Berechnung ermittelt werden. `"fstype"` sollte `4.2BSD` sein. Die `"fsize"`-, `"bsize"`-, und `"cpg"`- Werte sollten entsprechend dem eigentlichen Dateisystem gewählt werden, obwohl sie in diesem Kontext ziemlich unwichtig sind.

   Auf diese Art und Weise wird eine neue Partition `"a"` etabliert, die die Vinum-Partition auf diesem Gerät überschneidet. Beachte Sie, dass das `bsdlabel`-Kommando diese Überschneidung nur erlaubt, wenn die Partition richtig mit dem `"vinum"`-fstype markiert ist.

3. Das ist alles. Auf jedem Gerät befindet sich nun eine gefälschte `"a"`-Partition, die eine Kopie des Root-Volumes enthält. Es wird dringend empfohlen, das Resultat dieser Konfiguration zu überprüfen:

```
# fsck -n /dev/devnamea
```

Denken Sie stets daran, dass alle Dateien, die Kontrollinformationen enthalten, nun relativ zum Root-Dateisystem innerhalb des Vinum-Volumes sein müssen. Denn ein neu eingerichtetes Vinum-Root-Dateisystem ist möglicherweise inkompatibel zum gerade aktiven Root-Dateisystem. Deshalb müssen insbesondere die Dateien `/etc/fstab` und `/boot/loader.conf` überprüft werden.

Beim nächsten Systemstart sollte der Bootstrap die adäquaten Kontrollinformationen des neuen Vinum-basierten Root-Dateisystems automatisch herausfinden und entsprechend handeln. Am Ende des Kernel-Initialisierungsprozesses (nachdem alle Geräte angezeigt wurden) erhalten Sie bei einer erfolgreichen Konfiguration eine Nachricht ähnlich der folgenden:

```
Mounting root from ufs:/dev/gvinum/root
```

22.9.3. Beispiel eines Vinum-basierten Root-Setups

Nachdem das Vinum-Root-Volume eingerichtet wurde, könnte die Ausgabe von `gvinum l -rv root` bespielsweise so aussehen:

```
...
Subdisk root.p0.s0:
               Size:        125829120 bytes (120 MB)
               State: up
               Plex root.p0 at offset 0 (0  B)
               Drive disk0 (/dev/da0h) at offset 135680 (132 kB)

Subdisk root.p1.s0:
               Size:        125829120 bytes (120 MB)
               State: up
               Plex root.p1 at offset 0 (0  B)
               Drive disk1 (/dev/da1h) at offset 135680 (132 kB)
```

Wichtig ist hier insbesondere ist der Wert `135680` für die Verschiebung (relativ zur Partition `/dev/da0h`). Das entspricht beim Einsatz von `bsdlabel` 265 512-Byte-Plattenblöcken. Dieses Root-Volume ist ebenso 245760 512-Byte-Blöcke groß. `/dev/da1h` enthält die zweite Kopie dieses Root-Volumes und ist symmetrisch aufgebaut.

Das Bsdlabel für diese Geräte könnte so aussehen:

```
...
8 partitions:
#        size   offset    fstype   [fsize bsize bps/cpg]
  a:   245760      281    4.2BSD     2048 16384     0   # (Cyl.    0*- 15*)
  c: 71771688        0    unused        0     0         # (Cyl.    0 - 4467*)
  h: 71771672       16     vinum                       # (Cyl.    0*- 4467*)
```

Wie man leicht feststellen kann, entspricht der Parameter `"size"` der gefälschten `"a"`-Partition dem ausgewiesenen Wert von oben, während der Parameter `"offset"` gleich der Summe der Verschiebung innerhalb der Vinum-Partition `"h"` und der Verschiebung innerhalb des Geräts (oder Slice) ist. Dies ist ein typischer Aufbau, der

nötig ist, um die in Abschnitt 22.9.4.3 beschriebenen Probleme zu vermeiden. Die gesamte Partition `"a"` befindet sich in `"h"`, die alle Vinum-Daten für dieses Gerät enthält.

Beachten Sie, dass in dem oben beschriebenen Beispiel das gesamte Gerät Vinum gewidmet ist und keine Prä-Vinum-Partition zurückgelassen wurde, da es sich im Beispiel um eine neu eingerichtete Platte handelt, die nur für die Vinum-Konfiguration bestimmt war.

22.9.4. Fehlerbehebung

Der folgende Abschnitt beschreibt einige bekannte Probleme und Fallstricke bei der Vinum-Konfiguration sowie deren Behebung.

22.9.4.1. Der System-Bootstrap lädt zwar, das System startet aber nicht.

Wenn aus irgendeinem Grund das System nicht mit dem Booten fortfährt, kann man den Bootstrap während der 10-Sekunden-Warnung durch Drücken der **Leertaste** unterbrechen. Die *loader*-Variablen (wie `vinum.autostart`) können mittels des `show`-Kommandos untersucht, und mit `set` oder `unset` geändert werden.

Wenn das einzige Problem das Fehlen des Vinum-Kernelmoduls in der Liste der automatisch zu ladenden Module ist, hilft ein einfaches `load geom_vinum`.

Danach können Sie den Bootvorgang mit `boot -as` fortsetzen. Die Optionen `-as` fordern den Kernel auf, nach dem zu mountenden Root-Dateisystem zu fragen (`-a`), und den Bootvorgang im Single-User-Modus (`-s`) zu beenden, in dem das Root-Dateisystem schon schreibgeschützt gemountet ist. Auf diese Weise wird keine Dateninkonsistenz zwischen den Plexus riskiert, auch wenn nur ein Plexus eines Multi-Plexus-Volumes gemountet wurde.

Beim Prompt, das nach einem Root-Dateisystem fragt, kann jedes Gerät angegeben werden, dass ein gültiges Root-Dateisystem hat. Wenn `/etc/fstab` richtig konfiguriert wurde, sollte die Vorgabe etwas wie `ufs:/dev/gvinum/root` sein. Eine typische Alternative würde etwas wie `ufs:da0d` sein, welches eine hypothetische Partition sein könnte, die ein Pre-Vinum-Root-Dateisystem enthält. Vorsicht sollte walten, wenn eine der *alias* `"a"`-Partitionen hier eingegeben wird, die eigentlich Referenzen auf die Subdisks des Vinum-Root-Dateisystems sind, da so nur ein Stück eines gespiegelten Root-Gerätes gemountet werden würde. Wenn das Dateisystem später zum Lesen und Schreiben gemountet werden soll, ist es nötig, die anderen Plexus des Vinum-Root-Volumes zu entfernen, weil diese Plexus andernfalls inkonsistente Daten enthalten würden.

22.9.4.2. Nur der primäre Bootstrap lädt

Wenn das Laden von `/boot/loader` fehlschlägt, aber der primäre Bootstrap dennoch lädt (sichtbar an dem einzelnen Strich in der linken Spalte des Bildschirms gleich nachdem der Bootprozess startet), kann man versuchen, den primären Bootstrap an diesem Punkt durch Benutzen der **Leertaste** zu unterbrechen. Dies wird den Bootstrap in der zweiten Phase stoppen (siehe dazu auch Abschnitt 13.3.2). Hier kann nun der Versuch unternommen werden, von einer anderen Partition zu booten, wie beispielsweise dem vorhergehenden Root-Dateisystem, das von `"a"` verschoben wurde.

22.9.4.3. Nichts bootet, der Bootstrap hängt sich auf

Diese Situation wird vorkommen, wenn der Bootstrap durch die Vinum-Installation zerstört worden ist. Unglücklicherweise lässt Vinum am Anfang seiner Partition nur 4 KB frei und schreibt dahinter seine Kopfinformationen. Allerdings benötigen Stufe-Eins- und -Zwei-Bootstraps plus dem dazwischen eingebetteten

bsdlabel momentan 8 KB. Demzufolge wird die Vinum-Installation, wenn die Vinum-Partition mit der Verschiebung 0 (innerhalb eines Slice oder einer Platte, die zum Start bestimmt waren) eingerichtet wurde, den Bootstrap zerstören.

Analog wird eine anschließende Reinstallation eines Bootstrap (zum Beispiel durch Booten eines “Fixit”-Mediums) mit `bsdlabel -B`, wie in Abschnitt 13.3.2 beschrieben, den Vinum-Kopf zerstören und Vinum wird seine Platte(n) nicht mehr finden können. Obwohl keine eigentlichen Vinum-Konfigurationsdaten oder Daten in den Vinum-Volumes zerstört werden und es möglich wäre, alle Daten wiederherzustellen, indem die exakt gleichen Vinum-Konfigurationsdaten noch einmal eingegeben werden, bleibt die Situation schwer zu bereinigen, da es nötig ist, die gesamte Vinum-Partition um mindestens 4 KB nach hinten zu verschieben, damit Bootstrap und Vinum-Kopf nicht mehr kollidieren.

Fußnoten

1. RAID steht für *Redundant Array of Inexpensive Disks* und bietet verschiedene Formen der Fehlertorleranz, obwohl der letzte Begriff etwas irreführend ist, da RAID keine Redundanz bietet.

Kapitel 23. Virtualisierung

Beigetragen von Murray Stokely. Übersetzt von Oliver Peter.

23.1. Übersicht

Virtualisierungssoftware erlaubt es, mehrere Betriebssysteme gleichzeitig auf dem selben Computer laufen zu lassen. Derartige Softwaresysteme für PCs setzen in der Regel ein Host-Betriebssystem voraus, auf dem die Virtualisierungssoftware läuft und unterstützen eine nahezu beliebige Anzahl von Gast-Betriebssystemen.

Nachdem Sie dieses Kapitel gelesen haben,

- Kennen Sie den Unterscheid zwischen einem Host-Betriebssystem und einem Gast-Betriebssystem.
- Können Sie FreeBSD auf einem Intel-basierenden Apple Macintosh installieren.
- Können Sie FreeBSD unter Microsoft Windows und **Virtual PC** installieren.
- Wissen Sie, wie man ein virtualisiertes FreeBSD-System für optimale Leistung konfiguriert.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Grundlagen von UNIX und FreeBSD verstehen (Kapitel 4).
- FreeBSD installieren können (Kapitel 2).
- Wissen, wie man seine Netzwerkverbindung konfiguriert (Kapitel 32).
- Software Dritter installieren können (Kapitel 5).

23.2. FreeBSD als Gast-Betriebssystem

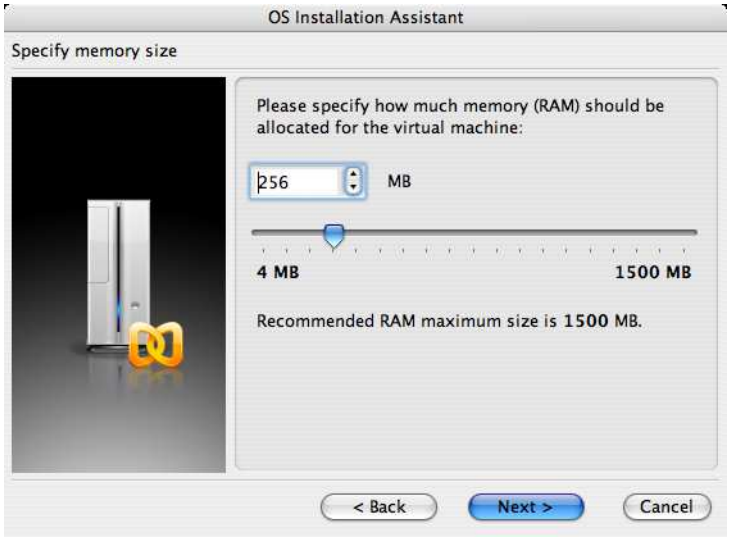
23.2.1. Parallels unter MacOS X

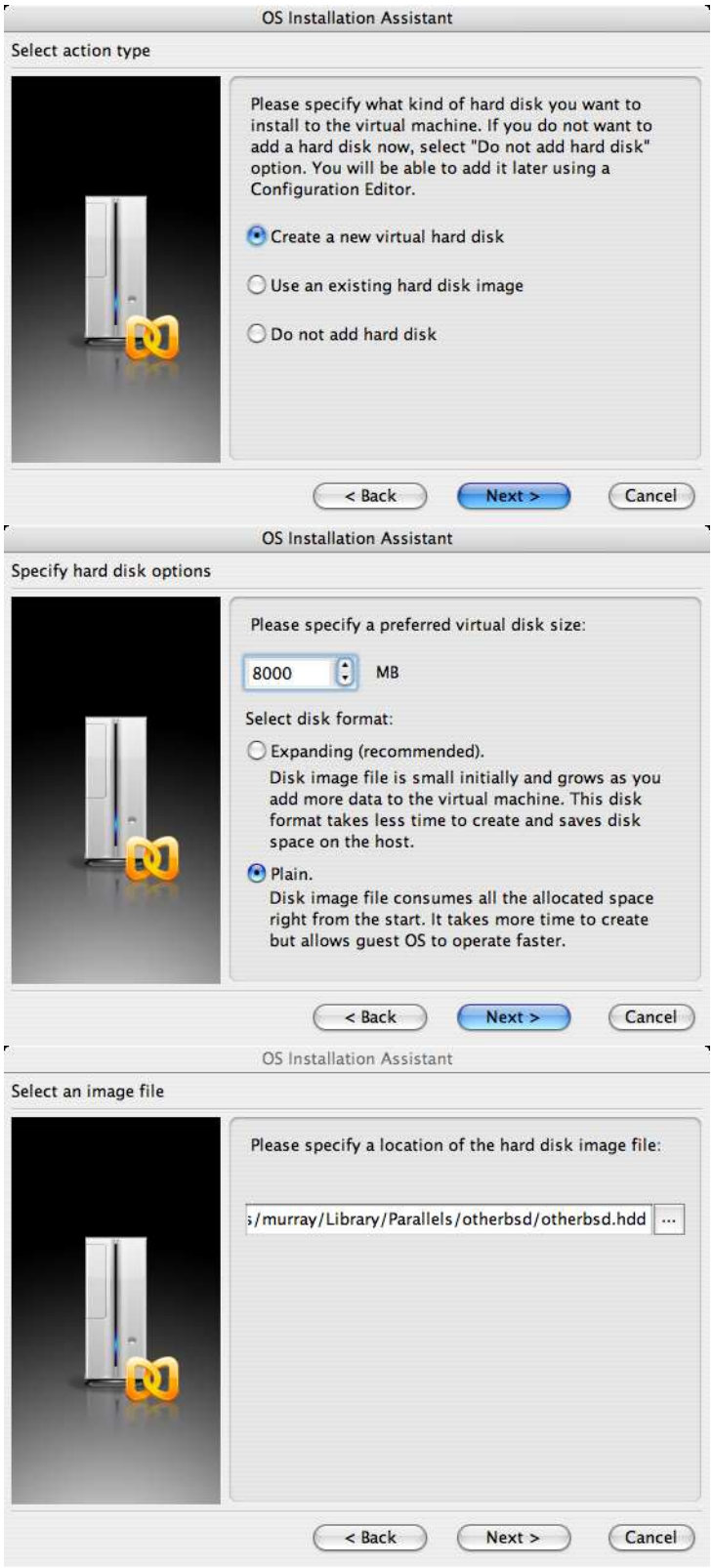
Parallels Desktop für Mac® ist ein kommerzielles Softwareprodukt, welches für Intel-basierende Apple Mac-Computer mit Mac OS X 10.4.6 oder höher verfügbar ist. FreeBSD wird von diesem Softwarepaket als Gast-Betriebssystem vollständig unterstützt. Nach der Installation von **Parallels** auf Mac OS X konfigurieren Sie als erstes eine virtuelle Maschine, in der Sie danach das gewünschte Gast-Betriebssystem (in unserem Fall FreeBSD) installieren.

23.2.1.1. Installation von FreeBSD unter Parallels/Mac OS® X

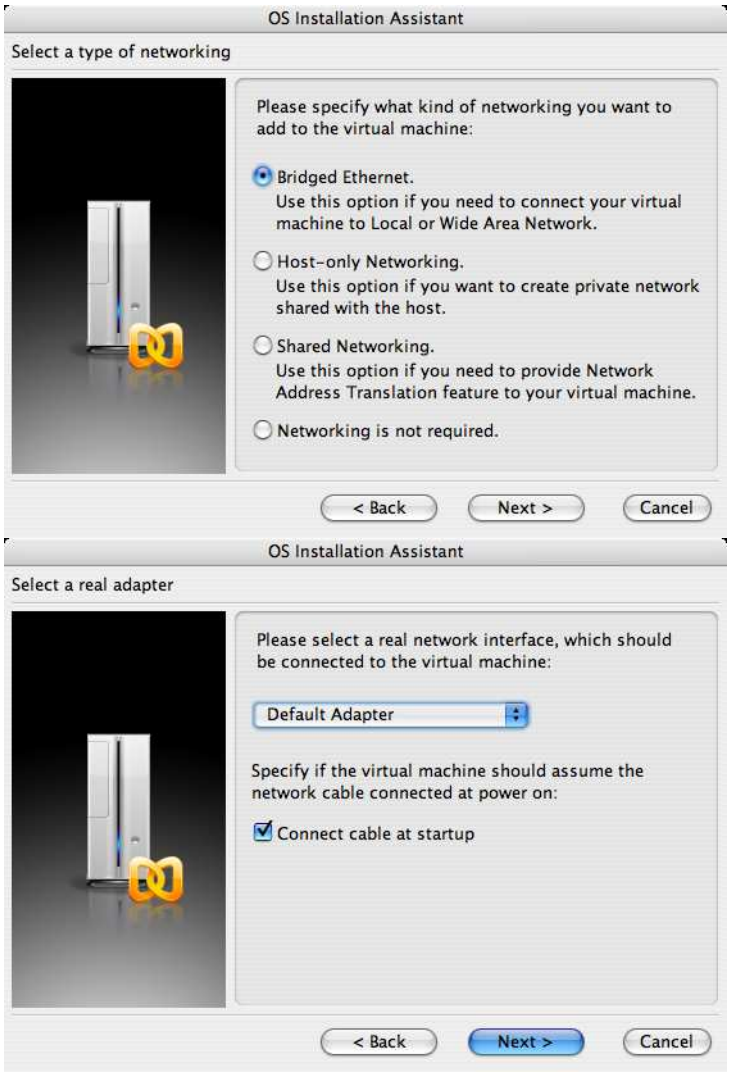
Der erste Schritt bei der Installation von FreeBSD unter **Parallels**/Mac OS X ist es, eine virtuelle Maschine zu konfigurieren, in der Sie FreeBSD installieren können. Dazu wählen Sie bei der Frage nach dem Guest OS Type FreeBSD aus:

Danach legen Sie geeignete Größen für Festplatten- und Arbeitsspeicher für die zu erstellende FreeBSD-Instanz fest. 4 GB Plattenplatz sowie 512 MB RAM sind in der Regel für die Arbeit unter **Parallels** ausreichend:

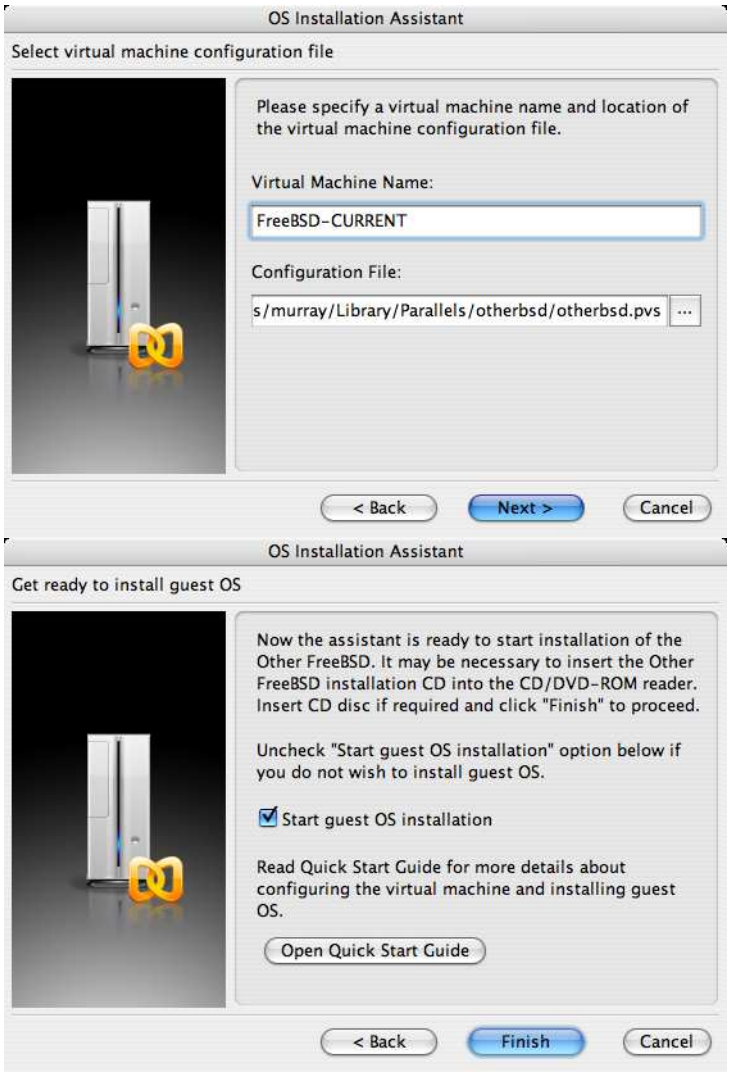




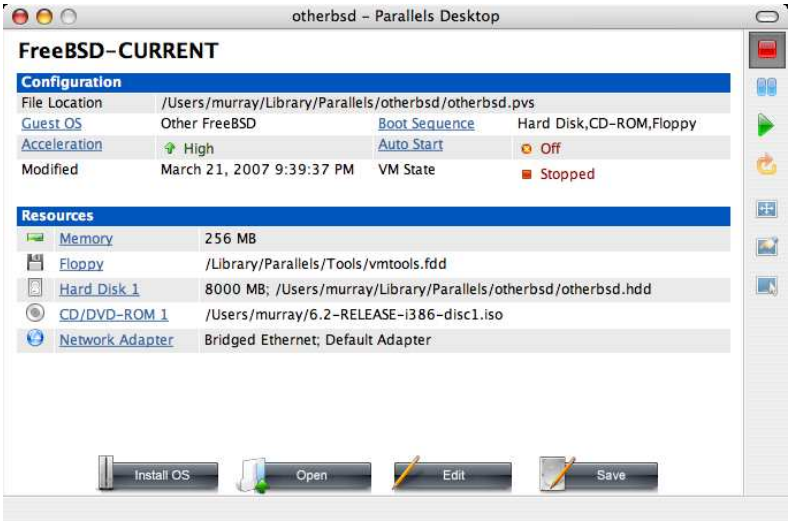
Wählen Sie den gewünschten Netzwerktyp aus und konfigurieren Sie Ihre Netzwerkverbindung:



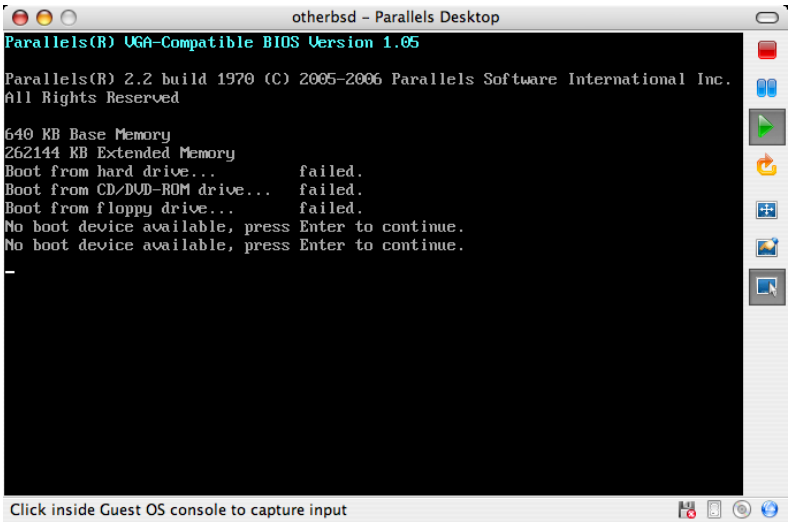
Speichern Sie Ihre Eingaben, um die Konfiguration abzuschließen:



Nachdem Sie die virtuelle Maschine erstellt haben, installieren Sie im nächsten Schritt FreeBSD in dieser virtuellen Maschine. Dazu verwenden Sie am besten eine offizielle FreeBSD-CDROM oder Sie laden von einem offiziellen FTP-Server ein ISO-Abbild auf Ihren Mac herunter. Danach klicken Sie auf das Laufwerksymbol in der rechten unteren Ecke des **Parallels**-Fensters, um ihr virtuelles Laufwerk mit dem ISO-Abbild oder mit dem physikalischen CD-ROM-Laufwerk ihres Computers zu verknüpfen.

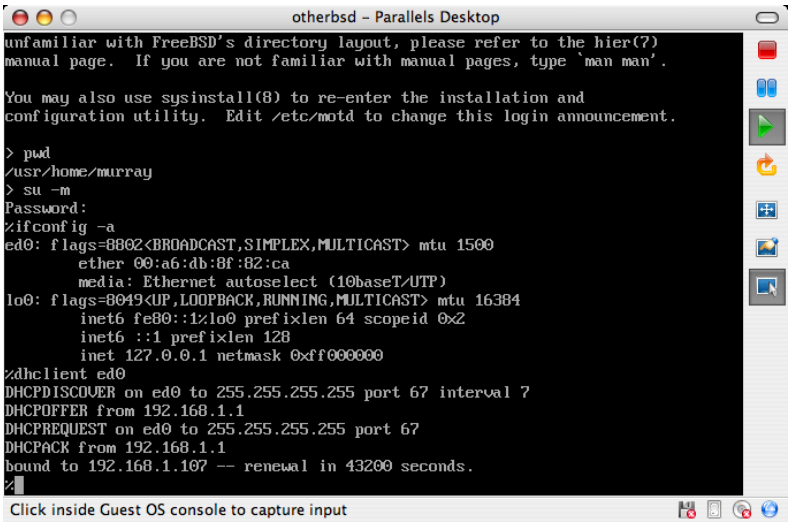


Nachdem Sie diese Verknüpfung hergestellt haben, starten sie die virtuelle FreeBSD-Maschine neu, indem Sie wie gewohnt auf das Symbol "Neustarten" klicken. **Parallels** startet nun ein Spezial-BIOS, das zuerst prüft, ob Sie eine CD-ROM eingelegt haben (genau so, wie es auch ein echtes BIOS machen würde).



In unserem Fall findet das BIOS ein FreeBSD-Installationsmedium und beginnt daher eine normale Installation mit **sysinstall** (wie in Kapitel 2 des Handbuchs beschreiben).

Nachdem die Installation abgeschlossen ist, können Sie die virtuelle Maschine starten.



23.2.1.2. FreeBSD für den Einsatz unter Parallels/Mac OS X optimieren

Nachdem Sie FreeBSD erfolgreich unter Mac OS X mit **Parallels** installiert haben, sollten Sie ihr virtuelles FreeBSD-System für virtualisierte Operationen optimieren:

1. Setzen der Bootloader-Variablen

   Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der **Parallels**-Umgebung zu verringern.

   ```
   kern.hz=100
   ```

   Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter **Parallels** trotzdem rund 15 Prozent der CPU-Leistung eines Single Prozessor iMac®'s verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 5 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

Sie können alle SCSI-, FireWire- und USB-Laufwerks-Treiber entfernen. **Parallels** stellt einen virtuellen Netzwerkadapter bereit, der den ed(4)-Treiber verwendet. Daher können alle Netzwerkgeräte bis auf ed(4) und miibus(4) aus dem Kernel entfernt werden.

3. Netzwerkbetrieb einrichten

   Die einfachste Netzwerkkonfiguration ist der Einsatz von DHCP, um Ihre virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich der Host-Mac befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_ed0="DHCP"` in die Datei `/etc/rc.conf` ein. Weitere Informationen zur Konfiguration des Netzwerks unter FreeBSD finden Sie im Kapitel 32 des Handbuchs.

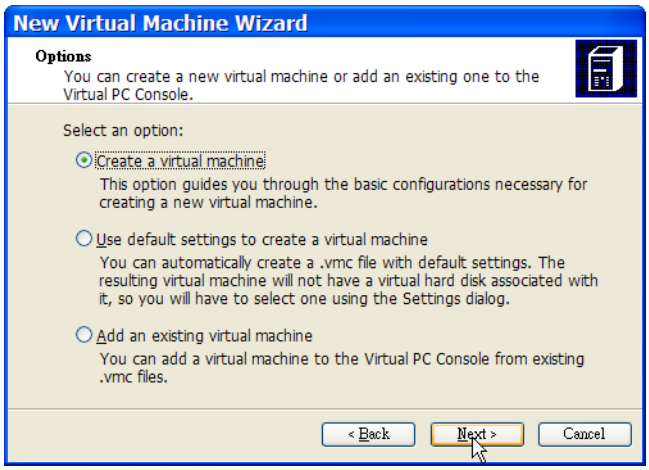
23.2.2. Virtual PC unter Windows

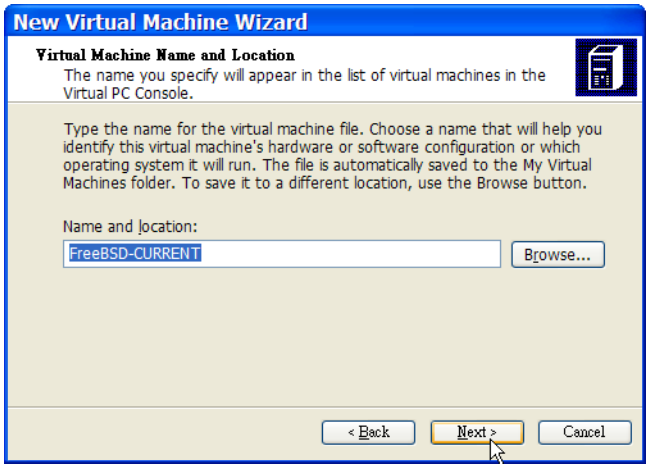
Übersetzt von Johann Kois.

Virtual PC für Windows wird von Microsoft kostenlos zum Download angeboten. Die Systemanforderungen für dieses Programm finden Sie hier (http://www.microsoft.com/windows/downloads/virtualpc/sysreq.mspx). Nachdem Sie **Virtual PC** unter Microsoft Windows installiert haben, müssen Sie eine virtuelle Maschine konfigurieren und das gewünschte Betriebssystem installieren.

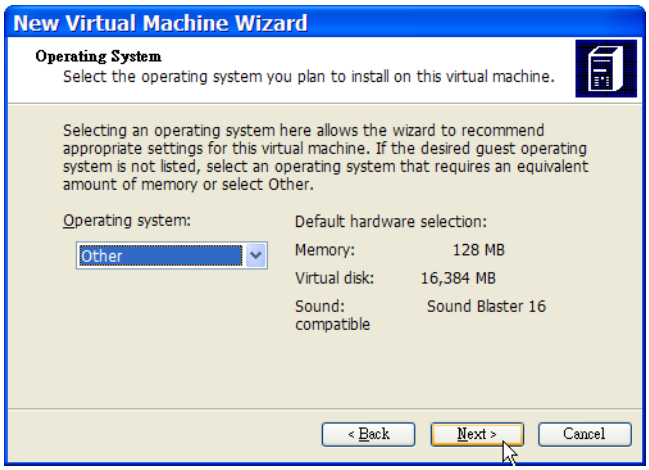
23.2.2.1. FreeBSD in Virtual PC/Microsoft® Windows installieren

Der erste Schritt zur Installation von FreeBSD in Microsoft Windows/**Virtual PC** ist es, eine neue virtuelle Maschine zu erstellen, in die Sie FreeBSD installieren können. Dazu wählen Sie die Option Create a virtual machine, wenn Sie danach gefragt werden:

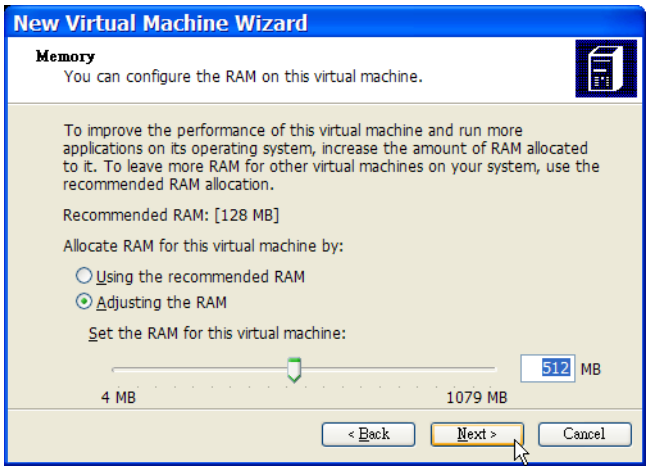


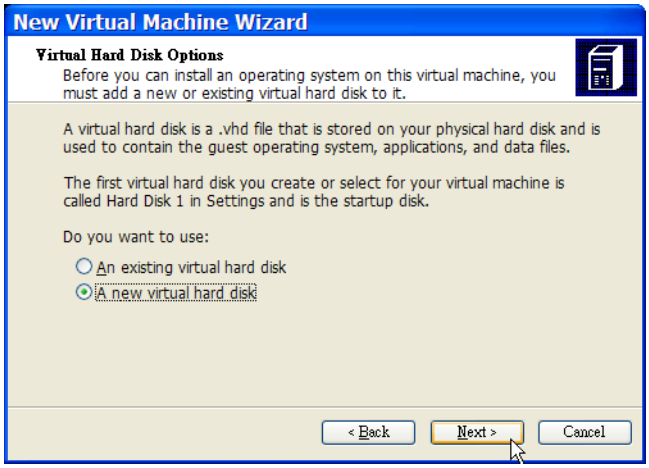


Bei der Frage nach dem Operating system wählen Sie Other:

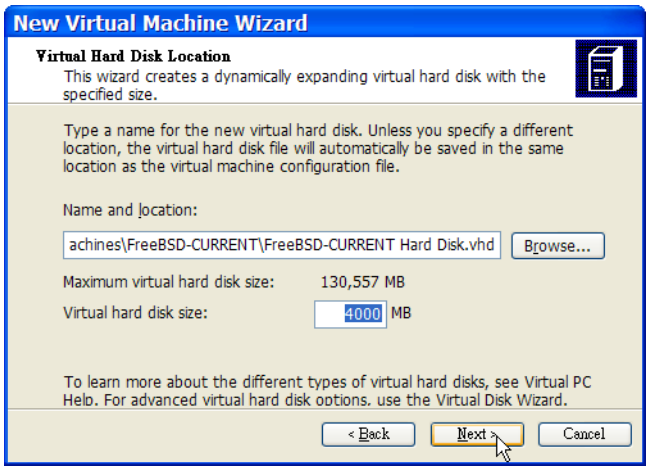


Danach müssen Sie den von Ihnen gewüschten Plattenplatz sowie die Größe des Hauptspeichers angeben. 4 GB Plattenplatz sowie 512 MB RAM sollten für die Installation von FreeBSD in **Virtual PC** ausreichend sein:

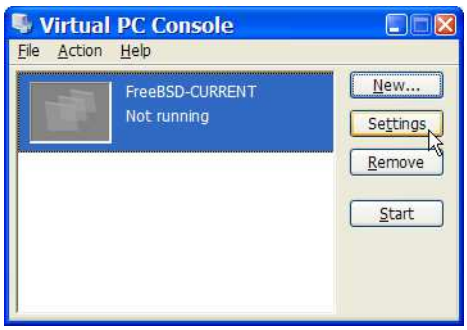


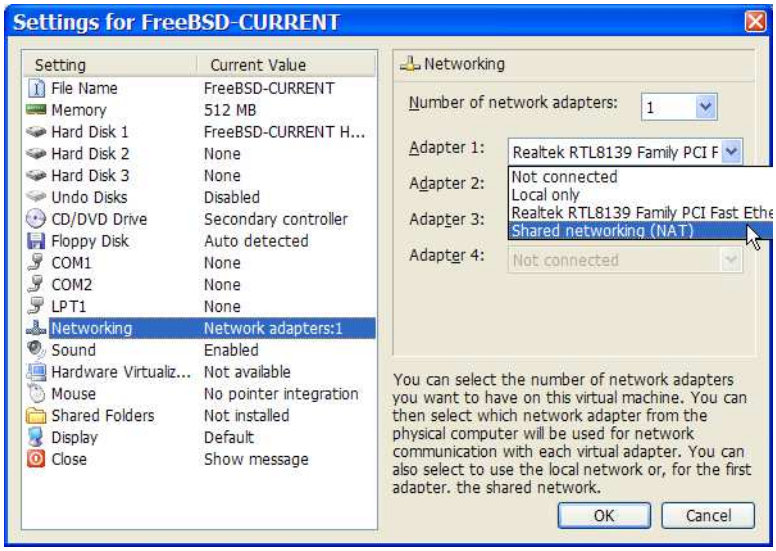


Speichern Sie Ihre Eingaben und beenden Sie die Konfiguration:

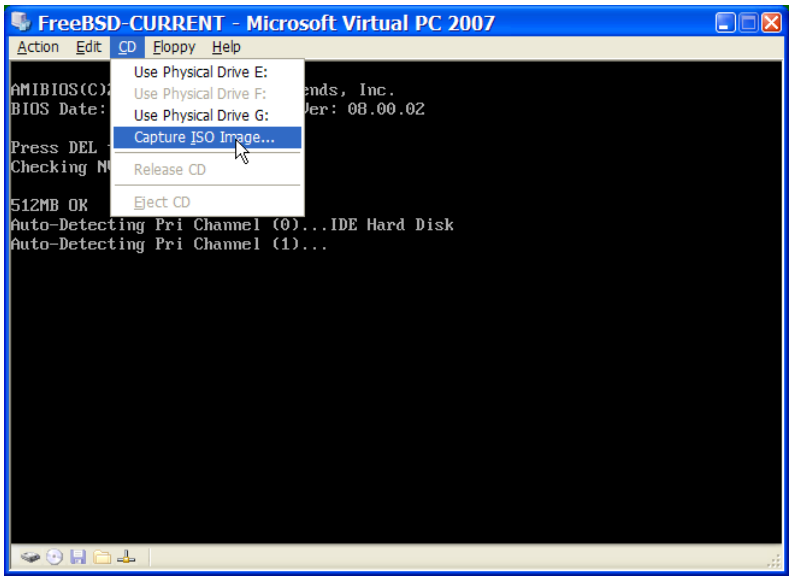


Wählen Sie nun die für FreeBSD erstellte virtuelle Maschine aus und klicken Sie auf Settings, um das Netzwerk zu konfigurieren:

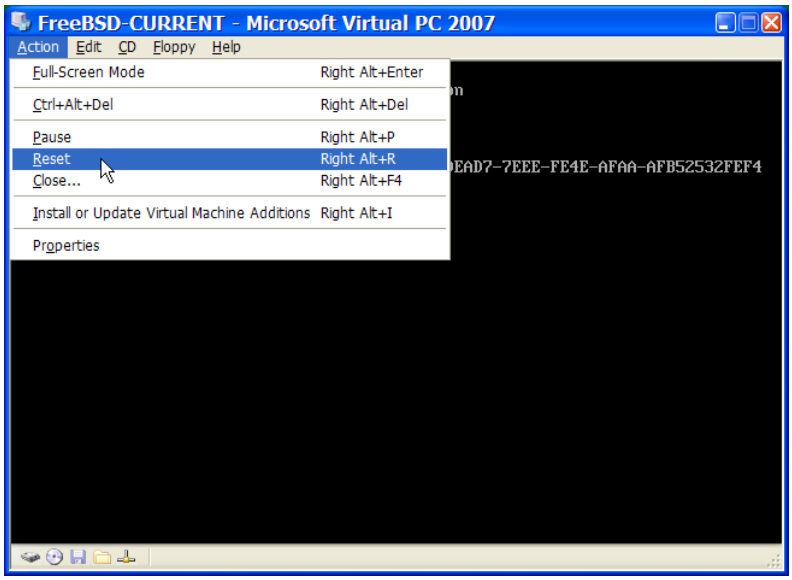




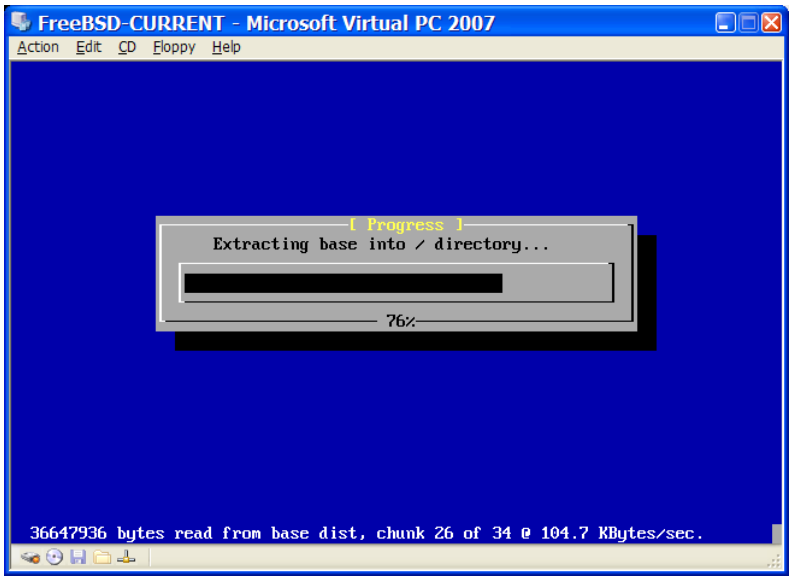
Nun können Sie FreeBSD installieren. Dazu verwenden Sie am besten eine offizielle FreeBSD-CD-ROM oder ein ISO-Image, das Sie von einem offiziellen FreeBSD-FTP-Server heruntergeladen haben. Wenn Sie ein ISO-Image auf Ihrer Festplatte gespeichert haben, oder eine FreeBSD-CD-ROM in Ihr CD-Laufwerk eingelegt haben, doppelklicken Sie auf die virtuelle Maschine, die Sie für FreeBSD angelegt haben. Danach klicken Sie auf CD und wählen die Option Capture ISO Image... im **Virtual PC**-Fenster. Danach können Sie im folgenden Fenster das CD-Laufwerk mit Ihrem physikalischen CD-Laufwerk oder mit dem ISO-Image verknüpfen.



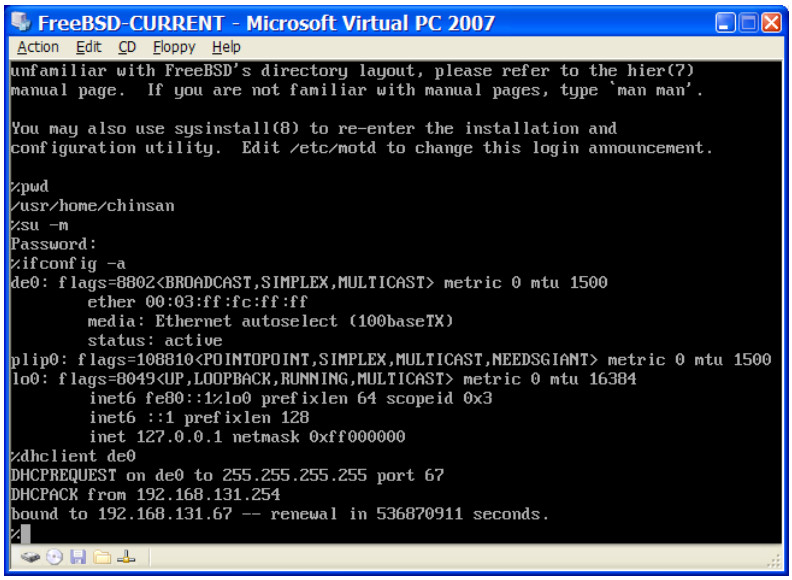
Danach starten Sie die virtuelle Maschine neu, indem Sie zuerst auf Action und danach auf Reset klicken. **Virtual PC** startet Ihre virtuelle Maschine nun neu und prüft zuerst, ob die virtuelle Maschine über ein CD-Laufwerk verfügt.



Da dies hier der Fall ist, beginnt nun eine normale, auf **sysinstall** basierende Installation, die in Kapitel 2 beschrieben wird. Sie können FreeBSD nun installieren. Verzichten Sie an dieser Stelle aber unbedingt auf die X11-Konfiguration.



Nachdem die Installation abgeschlossen ist, entfernen Sie die CD-ROM aus dem Laufwerk (oder lösen die Verknüpfung zum ISO-Image). Danach starten Sie die virtuelle Maschine neu, um FreeBSD zu starten.



23.2.2.2. FreeBSD in Microsoft Windows/Virtual PC konfigurieren

Nachdem Sie FreeBSD auf Ihrem Microsoft Windows-System erfolgreich unter **Virtual PC** installiert haben, sollten Sie ihr virtuelles FreeBSD noch anpassen, um eine optimale Funktion zu gewährleisten.

1. Setzen der Bootloader-Variablen

   Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der **Virtual PC**-Umgebung zu verringern. Dazu fügen Sie die folgende Zeile in die Datei `/boot/loader.conf` ein:

   ```
   kern.hz=100
   ```

Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter **Virutal PC** trotzdem rund 40 Prozent der CPU-Leistung eines Ein-Prozessor-Systems verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 5 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

   Sie können alle SCSI-, FireWire- und USB-Laufwerks-Treiber entfernen. **Virtual PC** stellt einen virtuellen Netzwerkadapter bereit, der den de(4)-Treiber verwendet. Daher können alle Netzwerkgeräte bis auf de(4) und miibus(4) aus dem Kernel entfernt werden.

3. Das Netzwerk einrichten

   Die einfachste Netzwerkkonfiguration ist der Einsatz von DHCP, um Ihre virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich Ihr Host-Microsoft Windows befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_de0="DHCP"` in die Datei `/etc/rc.conf` ein. Weitere Informationen zur Konfiguration des Netzwerks unter FreeBSD finden Sie im Kapitel 32 des Handbuchs.

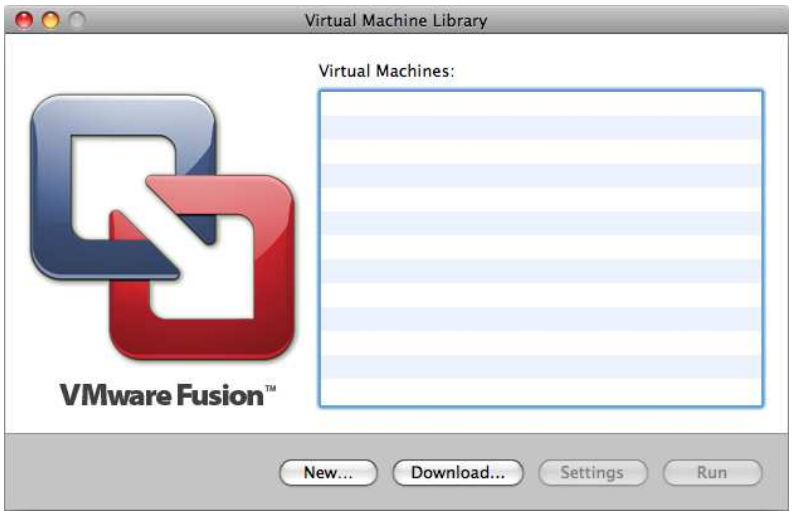
23.2.3. VMware unter MacOS

Übersetzt von Johann Kois.

VMware Fusion für Mac ist ein kommerzielles Programm, das für Intel basierte Apple Mac-Computer mit Mac OS 10.4.9 oder neuer erhältlich ist. FreeBSD wird von diesem Produkt vollständig als Gast-Betriebssystem unterstützt. Nachdem Sie **VMware Fusion** unter Mac OS X installiert haben, können Sie das gewünschte Gastbetriebssystem (in unserem Fall FreeBSD) installieren.

23.2.3.1. FreeBSD in VMware/Mac OS X installieren

Zuerst müssen Sie VMware Fusion starten, um eine virtuelle Maschine zu erstellen. Dazu wählen Sie die Option "New":



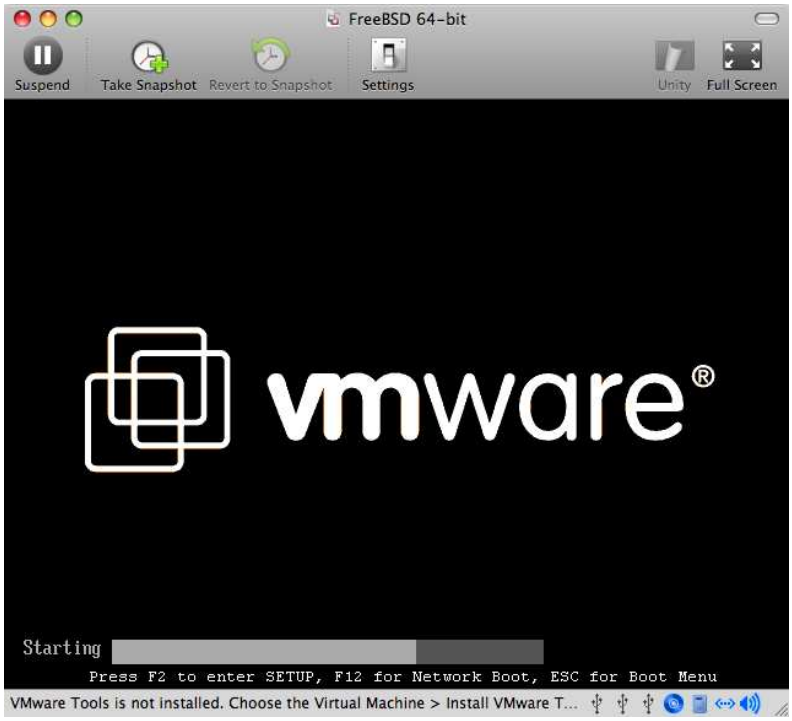
Dadurch wird ein Assistent gestartet, der Ihnen bei der Erzeugung einer neuen virtuellen Maschine behilflich ist. Clicken Sie auf "Continue", um den Prozess zu starten:

Wählen Sie Other als das Operating System, danach FreeBSD oder FreeBSD 64-bit, je nach dem, welche Version Sie installieren wollen, wenn Sie nach der zu installierenden Version gefragt werden:

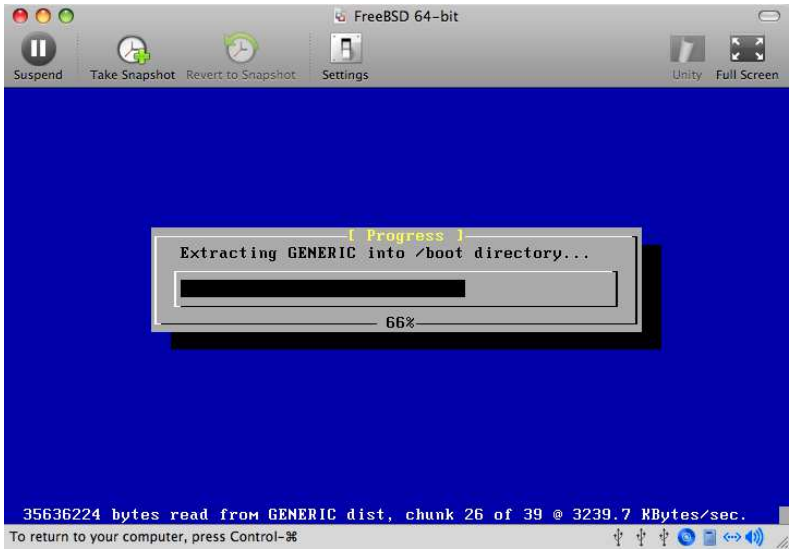
Vergeben Sie einen Namen für virtuelle Maschine an und legen Sie den Speicherort fest:

Legen Sie die Größe Ihrer virtuellen Festplatte fest:

Nachdem Sie auf "Finish" geklickt haben, wird die virtuelle Maschine gestartet:

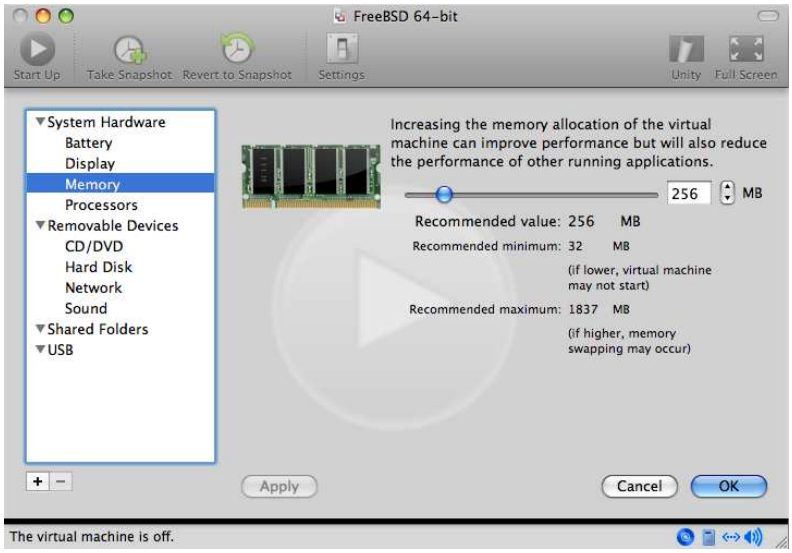


Nun können Sie FreeBSD wie gewohnt installieren (lesen Sie dazu auch Kapitel 2 des Handbuchs):



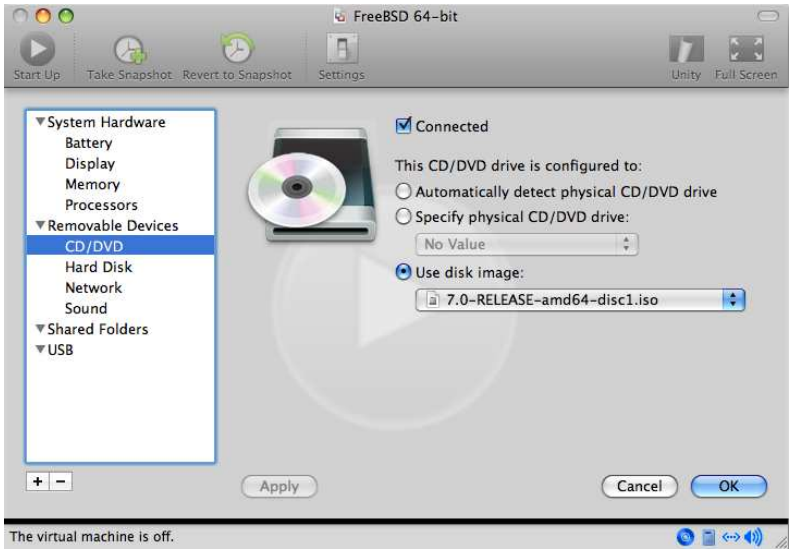
Nachdem die Installation abgeschlossen ist, können Sie noch verschiedene Parameter der virtuellen Maschine, etwa den Speicherverbrauch, konfigurieren:

> **Anmerkung:** Die Hardware der virtuellen Maschine kann nicht geändert werden, solange die virtuelle Maschine läuft.

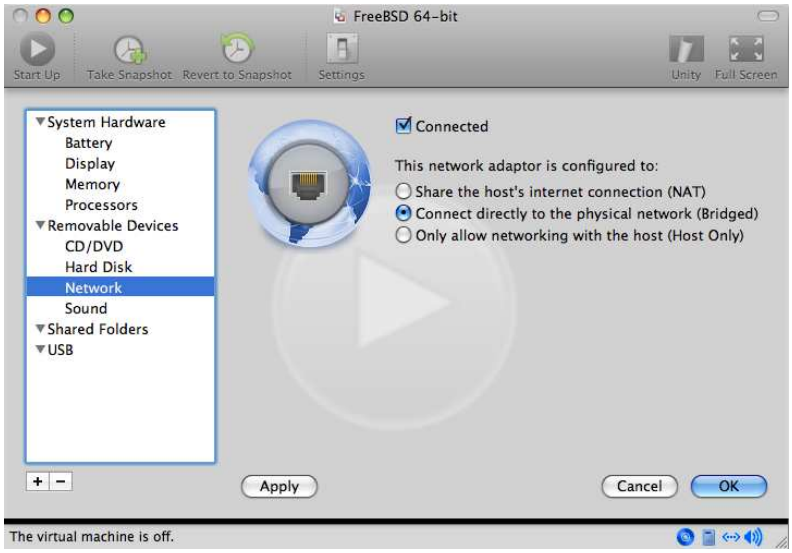


Die Anzahl der CPUs der virtuellen Maschine:

Den Status des CD-Laufwerks. Sie können das CD-Laufwerk von der virtuellen Maschine lösen, wenn Sie es nicht benötigen.



Zuletzt sollten Sie noch festlegen, wie sich die virtuelle Maschine mit dem Netzwerk verbinden soll. Sollen neben dem Gastsystem auch andere Rechner auf Ihre virtuelle Maschine zugreifen können, müssen Sie die Option Connect directly to the physical network (Bridged) wählen. Ist dies nicht der Fall, sollten Sie die Option Share the host's internet connection (NAT) wählen. In dieser Einstellung kann die virtuelle Maschine zwar auf auf das Internet zugreifen, andere Rechner dürfen aber nicht auf die virtuelle Maschine zugreifen.



Nachdem Sie die Konfiguration abgeschlossen haben, können Sie FreeBSD starten.

23.2.3.2. FreeBSD unter Mac OS X/VMware konfigurieren

Nachdem Sie FreeeBSD erfolgreich unter **VMware** für Mac OS X installiert haben, sollten Sie ihr virtuelles FreeBSD noch anpassen, um eine optimale Funktion zu gewährleisten.

1. Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der **VMware**-Umgebung zu verringern.

`kern.hz=100`

Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter **VMware** trotzdem rund 15 Prozent der CPU-Leistung eines Single Prozessor iMac's verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 5 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

   Sie können alle FireWire- und USB-Laufwerks-Treiber entfernen. **VMware** stellt einen virtuellen Netzwerkadapter bereit, der den em(4)-Treiber verwendet. Daher können alle Netzwerkgeräte bis auf em(4) und miibus(4) aus dem Kernel entfernt werden.

3. Netzwerkbetrieb einrichten

   Die einfachste Netzwerkkonfiguration ist der Einsatz von DHCP, um Ihre virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich der Host-Mac befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_em0="DHCP"` in die Datei `/etc/rc.conf` ein. Weitere Informationen zur Konfiguration des Netzwerks unter FreeBSD finden Sie im Kapitel 32 des Handbuchs.

23.3. FreeBSD als Host-Betriebssystem

Übersetzt von Benedict Reuschling und Christoph Sold.

Seit einigen Jahren wurde FreeBSD nicht offiziell von irgendeiner der verfügbaren Virtualisierungslösungen als Host-Betriebssystem unterstützt. Viele Anwender verwenden aber noch ältere **VMware**-Versionen (z.B. `emulators/vmware3`), welches die Linux-Kompatibilitätsschicht nutzt. Kurz nach der Veröffentlichung von FreeBSD 7.2 erschien **VirtualBox™** als Open-Source Edition (OSE) von Sun in der Ports-Sammlung als ein direkt auf FreeBSD lauffähiges Programm.

VirtualBox ist ein vollständiges Virtualisierungspaket, das aktiv weiterentwickelt wird und für die meisten Betriebssysteme einschliesslich Windows, Mac OS, Linux und FreeBSD zur Verfügung steht. Es kann sowohl Windows als auch UNIX-ähnliche Gastsysteme betreiben. Es ist als Open Source und als proprietäre Edition erhältlich. Die wichtigste Einschränkung der OSE aus Anwendersicht ist die fehlende USB-Unterstützung. Weitere Unterschiede können von der "Editions"-Seite des **VirtualBox**-Wikis, das unter http://www.virtualbox.org/wiki/Editions zu finden ist, entnommen werden. Momentan steht nur OSE unter FreeBSD zur Verfügung.

23.3.1. VirtualBox™ installieren

VirtualBox steht als FreeBSD-Port in `emulators/virtualbox-ose` bereit und kann über den folgenden Befehl installiert werden:

```
# cd /usr/ports/emulators/virtualbox-ose
# make install clean
```

Eine nützliche Option im Konfigurationsdialog ist die `GuestAdditions`-Programmsammlung. Diese stellen eine Reihe von nützlichen Eigenschaften in den Gastbetriebssystemen zur Verfügung, wie beispielsweise Mauszeigerintegration (was es ermöglicht, die Maus zwischen dem Host und dem Gast zu teilen ohne eine spezielle Tastenkombination für diesen Wechsel zu drücken), sowie schnelleres Rendern von Videos, besonders in Windows

Gästen. Diese Gastzusätze sind im Devices-Menü zu finden, nachdem die Installation des Gastbetriebssystem abgeschlossen ist.

Ein paar Konfigurationsänderungen sind notwendig, bevor **VirtualBox** das erste Mal gestartet wird. Der Port installiert ein Kernelmodul in `/boot/modules`, das in den laufenden Kernel geladen werden muss:

```
# kldload vboxdrv
```

Um sicherzustellen, dass das Modul immer nach einem Neustart geladen wird, fügen Sie die folgende Zeile in die Datei `/boot/loader.conf` ein:

```
vboxdrv_load="YES"
```

Ältere Versionen als 3.1.2 von **VirtualBox** benötigen auch das eingehängte `proc`-Dateisystem. Dies wird in aktuellen Versionen nicht mehr benötigt, da dort die Funktionen von der sysctl(3) Bibliothek bereitgestellt werden.

Wenn Sie eine ältere Version aus den Ports benutzen, befolgen Sie die unten stehenden Anweisungen und stellen Sie sicher, dass `proc` eingehangen ist.

```
# mount -t procfs proc /proc
```

Um auch diese Einstellung nach einem Neustart zu erhalten, wird die folgende Zeile in `/etc/fstab` eingefügt:

```
proc      /proc   procfs  rw      0       0
```

> **Anmerkung:** Möglicherweise erscheint eine Fehlermeldung ähnlich der Folgenden, wenn **VirtualBox** von einem Terminal aus gestartet wird:
>
> ```
> VirtualBox: supR3HardenedExecDir: couldn't read "", errno=2 cchLink=-1
> ```
>
> Wahrscheinlich ist der Übeltäter das `proc`-Dateisystem. Verwenden Sie bitte das `mount`-Kommando um zu überprüfen, ob es korrekt eingehängt ist.

Die Gruppe `vboxusers` wird während der Installation von **VirtualBox** angelegt. Alle Benutzer, die Zugriff auf **VirtualBox** haben sollen, müssen in diese Gruppe aufgenommen werden. Der `pw`-Befehl kann benutzt werden, um neue Mitglieder hinzuzufügen:

```
# pw groupmod vboxusers -m yourusername
```

Um **VirtualBox** zu starten, wählen Sie entweder den Eintrag Sun VirtualBox aus dem Menü Ihrer graphischen Benutzeroberfläche, oder geben Sie den folgenden Befehl in ein Terminal ein:

```
% VirtualBox
```

Besuchen Sie die offizielle Webseite von **VirtualBox** unter http://www.virtualbox.org, um weitere Informationen zur Konfiguration und Verwendung zu erhalten. Da der FreeBSD-Port noch recht neu ist, befindet er sich noch unter ständiger Entwicklung. Um die aktuellen Nachrichten und Anleitungen zur Fehlerbehebung zu erhalten, besuchen Sie die entsprechende Seite im FreeBSD-Wiki unter http://wiki.FreeBSD.org/VirtualBox.

Kapitel 24. Lokalisierung – I18N/L10N einrichten und benutzen

Beigesteuert von Andrey Chernov. Überarbeitet von Michael C. Wu. Übersetzt von Alexander Langer und Martin Heinen.

24.1. Übersicht

FreeBSD ist ein über die ganze Welt verteiltes Projekt. Dieses Kapitel behandelt die Internationalisierung und Lokalisierung von FreeBSD, mit denen nicht englisch sprechende Benutzer FreeBSD an ihre Bedürfnisse anpassen können. Die Internationalisierung betrifft sowohl die System- als auch die Anwendungsebene, daher wird im Laufe des Texts auf genauere Anwendungsdokumentationen verwiesen.

Nachdem Sie dieses Kapitel durchgearbeitet haben, werden Sie wissen

- wie verschiedene Sprachen und Lokalisierungen in modernen Betriebssystemen codiert werden,
- wie Sie die Locale Ihrer Login-Shell setzen,
- wie Sie die Konsole für nicht-englische Sprachen konfigurieren,
- wie Sie das X Window System mit verschiedenen Sprachen benutzen,
- wo Sie mehr Informationen über das Erstellen von I18N-konformen Anwendungen erhalten.

Bevor Sie dieses Kapitel lesen, sollten Sie

- wissen, wie Sie zusätzliche Anwendungen installieren (Kapitel 5).

24.2. Grundlagen

24.2.1. Was ist I18N/L10N?

Entwickler kürzen das Wort *internationalization* (englisch für Internationalisierung) mit I18N ab, weil sich zwischen dem ersten und letzten Buchstaben des Worts 18 Buchstaben befinden. L10N benutzt die gleiche Namensgebung und ist eine Abkürzung des Worts *localization* (englisch für Lokalisierung). Mit I18N/L10N-Methoden, -Protokollen und -Anwendungen können Benutzer eine Sprache ihrer Wahl verwenden.

I18N-Anwendungen werden mit Hilfe von I18N-Bibliotheken programmiert. Diese erlauben es Entwicklern, eine einfache Sprachdatei zu schreiben und Menüs und Texte an jede Sprache anzupassen. Wir möchten Programmierern empfehlen, für ihre eigenen Anwendungen auf diese Techniken zurückzugreifen.

24.2.2. Wieso soll ich I18N/L10N benutzen?

I18N/L10N wird immer dann benutzt, wenn Sie Daten in anderen Sprachen als Englisch anzeigen, eingeben oder verarbeiten möchten.

24.2.3. Welche Sprachen werden von I18N unterstützt?

I18N und L10N sind nichts FreeBSD spezifisches. Momentan können Sie unter den meisten der verbreitetsten Sprachen der Welt wählen, unter anderen Chinesisch, Japanisch, Koreanisch, Französisch, Russisch und Deutsch.

24.3. Lokale Anpassungen benutzen

In seiner ganzen Schönheit ist L10N nichts, was auf FreeBSD alleine beschränkt ist, im Gegenteil, es ist eine Konvention, an die sich viele Programme für verschiedene Betriebssysteme halten. Wir möchten Sie anregen, FreeBSD bei der Unterstützung dieser Konvention zu helfen.

Lokale Anpassungen werden durch die Angabe von drei Werten erreicht: dem Sprachcode, dem Ländercode und der Codierung. Die Zusammenfassung dieser Werte wird “Locale” genannt und sieht wie folgt aus:

```
Sprachcode_Ländercode.Codierung
```

24.3.1. Sprach- und Ländercodes

Um FreeBSD (oder ein anderes UNIX System, das I18N unterstützt) an lokale Gegebenheiten und Sprachen anzupassen, muss der Benutzer herausfinden, welche Codes für sein Land und seine Sprache benutzt werden. Ländercodes geben den Anwendungen dabei vor, welche Variation einer bestimmten Sprache zu benutzen ist. Eine Variation von Deutsch wäre zum Beispiel de_CH, das eine lokale Anpassung an das in der Schweiz gesprochene Deutsch meint. Außerdem benutzen Webbrowser, SMTP/POP Server, Webserver usw. diese, um Entscheidungen über die Sprache zu fällen. Im Folgenden sind einige Beispiele für Sprach- und Ländercodes aufgelistet:

Sprachcode/Ländercode	Beschreibung
en_US	Englisch - USA
ru_RU	Russisch für Russland
zh_TW	Traditionelles Chinesisch für Taiwan

24.3.2. Codierungen

Einige Sprachen benutzen Codierungen, die nicht dem 7-Bit breitem ASCII-Standard entsprechen, wie 8-Bit Codierungen, Wide- oder Multibyte Zeichen (multibyte(3) geht darauf näher ein). Ältere Anwendungen erkennen diese Zeichen nicht und halten sie fälschlicherweise für Steuerzeichen. Neuere Anwendungen erkennen für gewöhnlich 8-Bit Zeichen. Es hängt allerdings von der Implementierung ab, ob man eine Anwendung neu kompilieren muss, um in den Genuss von lokalen Zeichensätzen zu kommen, oder ob man es sie nur nachträglich konfigurieren muss. Um es möglich zu machen, Wide- oder Multibyte-Zeichen einzugeben und zu verarbeiten, unterstützt die FreeBSD-Ports-Sammlung (http://www.FreeBSD.org/de/ports/index.html) verschiedene Sprachen für diverse Programme. Bitte konsultieren Sie die I18N-Dokumentation des entsprechenden FreeBSD-Ports.

In den meisten Fällen muss der Benutzer in die Dokumentation des Programms schauen, um herauszufinden, wie man es entsprechend für die eigene Sprache und den eigenen Zeichensatz konfiguriert, oder welche Optionen beim Übersetzen anzugeben sind.

Einige Dinge, die man im Hinterkopf behalten sollte, sind:

- Sprachbezogene C-char Zeichensätze[1] (siehe multibyte(3)), zum Beispiel ISO8859-1, ISO8859-15, KOI8-R, CP437.
- Wide- oder Multibyte-Codierungen, zum Beispiel EUC, Big5.

Eine aktuelle Liste der Zeichensätze ist in der IANA Registry (http://www.iana.org/assignments/character-sets). verfügbar.

> **Anmerkung:** Ab FreeBSD 4.5 werden X11-kompatible Codierungen verwendet.

24.3.3. I18N-Anwendungen

Im FreeBSD-Ports- und Paket-System werden I18N-Anwendungen mit einem `I18N` im Namen gekennzeichnet, damit man sie leicht identifizieren kann. Trotzdem kann es vorkommen, dass die benötigte Sprache nicht immer unterstützt wird.

24.3.4. Einstellen der Locale

Zum Aktivieren der Lokalisierung reicht es, die Umgebungsvariable `LANG` in Ihrer Login-Shell auf den Wert der Locale zu setzen und die Variable zu exportieren. Dies geschieht normalerweise in Ihrer `~/.login_conf` oder der Startdatei Ihrer Shell (`~/.profile`, `~/.bashrc`, `~/.cshrc`). Wenn `LANG` gesetzt ist, brauchen die speziellen Variablen wie `LC_CTYPE` oder `LC_CTIME` in der Regel nicht gesetzt zu werden. Sie sollten sprachbezogene FreeBSD-Dokumentation zu Rate ziehen, wenn Sie mehr Informationen wünschen.

Setzen Sie die zwei folgenden Umgebungsvariablen in Ihren Konfigurationsdateien:

- `LANG` für Funktionen der POSIX setlocale(3) Familie
- `MM_CHARSET` gibt den den MIME Zeichensatz von Anwendungen an

Damit ist die Locale für die Shell, jede Anwendung und X11 eingestellt.

24.3.4.1. Verfahren zum Einstellen der Locale

Es gibt zwei Wege, die Locale zu setzen, die im Folgenden beschrieben werden. Die erste und empfohlene Methode ist, die Umgebungsvariablen in der Login-Klasse zu setzen, die zweite ist, sie in den Startdateien der Shell zu setzen.

24.3.4.1.1. Lokalisierung in der Login-Klasse

Wenn Sie diese Methode verwenden, werden die Umgebungsvariablen für die Locale und den MIME Zeichensatz einmal für alle Shells, anstatt einzeln für jede Shell, gesetzt. Die Lokalisierung kann von einem Benutzer selbst oder von einem Administrator mit Superuser-Rechten für alle eingestellt werden.

24.3.4.1.1.1. Einrichten als Benutzer

`.login_conf` im Heimatverzeichnis eines Benutzers sollte mindestens die folgenden Einträge enthalten, damit beide Variablen für den Gebrauch der Latin-1 Codierung gesetzt werden:

```
me:\
       :charset=ISO-8859-1:\
       :lang=de_DE.ISO8859-1:
```

Damit traditionelles Chinesisch (BIG-5 Codierung) verwendet werden kann, sind in `.login_conf` die nachstehenden Ergänzungen vorzunehmen. Einige Programme behandeln die Lokalisierung für Chinesisch, Japanisch und Koreanisch falsch, daher müssen mehr Variablen als üblich gesetzt werden:

```
#Users who do not wish to use monetary units or time formats
#of Taiwan can manually change each variable
me:\
       :lang=zh_TW.Big5:\
       :setenv=LC_ALL=zh_TW.Big5:\
       :setenv=LC_COLLATE=zh_TW.Big5:\
       :setenv=LC_CTYPE=zh_TW.Big5:\
       :setenv=LC_MESSAGES=zh_TW.Big5:\
       :setenv=LC_MONETARY=zh_TW.Big5:\
       :setenv=LC_NUMERIC=zh_TW.Big5:\
       :setenv=LC_TIME=zh_TW.Big5:\
       :charset=big5:\
       :xmodifiers="@im=gcin": #Set gcin as the XIM Input Server
```

Weitere Informationen entnehmen Sie bitte login.conf(5).

24.3.4.1.1.2. Einrichten als Administrator

Stellen Sie sicher, dass in der Login-Klasse der Benutzer in `/etc/login.conf` die richtige Sprache eingestellt ist. Die folgenden Einstellungen müssen in `/etc/login.conf` vorgenommen werden:

```
Sprache|Account-Typ-Beschreibung:\
       :charset=MIME_Zeichensatz:\
       :lang=Locale:\
       :tc=default:
```

Die für Latin-1 erforderlichen Einträge sehen wie folgt aus:

```
german|German Users Accounts:\
       :charset=ISO-8859-1:\
       :lang=de_DE.ISO8859-1:\
       :tc=default:
```

Bevor Sie die Login-Klasse eines Benutzers ändern, müssen Sie den folgenden Befehl ausführen:

```
# cap_mkdb /etc/login.conf
```

Erst danach werden Ihre Änderungen in `/etc/login.conf` im System sichtbar.

Ändern der Login-Klasse mit vipw(8)

Wenn Sie neue Accounts mit `vipw` anlegen, erstellen Sie Einträge in folgender Art:

`user:password:1111:11:`*`Sprache`*`:0:0:Benutzername:/home/user:/bin/sh`

Ändern der Login-Klasse mit adduser(8)

Wenn Sie neue Accounts mit `adduser` anlegen, stehen Ihnen die folgenden Möglichkeiten zur Verfügung:

- Geben Sie in `/etc/adduser.conf` mit `defaultclass =` *`Sprache`* eine Sprache vor. In diesem Fall müssen Sie für Benutzer anderer Sprachen eine andere Login-Klasse angeben.
- Geben Sie die Sprache jedes Mal ein, wenn Sie dazu von adduser(8) aufgefordert werden:

  ```
  Enter login class: default []:
  ```

- Sie können die Login-Klasse auch auf der Kommandozeile von adduser(8) übergeben:

  ```
  # adduser -class Sprache
  ```

Ändern der Login-Klasse mit pw(8)

Wenn Sie neue Accounts mit pw(8) anlegen, benutzen Sie die folgende Kommandozeile:

```
# pw useradd Account -L Sprache
```

24.3.4.1.2. Lokalisierung in den Startdateien der Shells

> **Anmerkung:** Da Sie jede Shell unterschiedlich einrichten müssen, sollten Sie diese Methode nicht verwenden. Benutzen Sie stattdessen bitte Login-Klassen.

Um die Locale und den MIME Zeichensatz anzugeben, setzen Sie die unten aufgeführten Variablen in den Startdateien der Shells (`/etc/profile` und `/etc/csh.login`). In den folgenden Beispielen verwenden wir die deutsche Sprache.

Einstellungen in `/etc/profile`:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Einstellungen in `/etc/csh.login`:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Alternativ können Sie die Einstellungen in den Vorgabedateien der Shells vornehmen. Die oben gezeigten Einstellungen aus `/etc/profile` tragen Sie dann in `/usr/share/skel/dot.profile` und die Einstellungen aus `/etc/csh.login` in `/usr/share/skel/dot.login` ein.

Die Einstellungen für X11 in `$HOME/.xinitrc` sind von der verwendeten Login-Shell abhängig. Mit Bourne Shells verwenden Sie den folgenden Eintrag:

```
LANG=de_DE.ISO8859-1; export LANG
```

Mit C-Shells verwenden Sie den nachstehenden Eintrag:

```
setenv LANG de_DE.ISO8859-1
```

24.3.5. Einrichten der Konsole

Wenn Sie C-char Zeichensätze[1] verwenden, müssen Sie die richtigen Zeichensätze für die gewählte Sprache in `/etc/rc.conf` angeben:

```
font8x16=Zeichensatz
font8x14=Zeichensatz
font8x8=Zeichensatz
```

Dabei ist *`Zeichensatz`* der Name der passenden Datei aus `/usr/share/syscons/fonts` ohne die Endung `.fnt`.

Setzen Sie bei Bedarf die richtige Tasten- und Bildschirmzuordnung (keymap und screenmap). Dies können Sie in `sysinstall` einstellen, indem Sie Configure und dann Console wählen. Sie können die Zuordnungen aber auch direkt in `/etc/rc.conf` angeben:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

`screenmap_name` ist der Name einer Datei aus `/usr/share/syscons/scrnmaps` ohne die Endung `.scm`. Eine Bildschirmzuordnung und der zugehörige Zeichensatz verbreitert die Zeichenmatrix von VGA Karten im Pseudographik Modus von 8 Bit auf 9 Bit. Sie wird benötigt, wenn der Zeichensatz des Bildschirms 8 Bit verwendet.

Lesen Sie den nächsten Absatz, wenn Sie in `/etc/rc.conf` den **moused** Dæmon mit der nachstehenden Anweisung aktiviert haben:

```
moused_enable="YES"
```

Der Mauszeiger des syscons(4) Treibers belegt in der Voreinstellung den Bereich von 0xd0 bis 0xd3 des Zeichensatzes. Wenn dieser Bereich ebenfalls von der eingestellten Sprache benötigt wird, müssen Sie den Mauszeiger verschieben. Dazu fügen Sie die folgende Zeile in Ihre Kernelkonfigurationsdatei ein:

```
mousechar_start=3
```

`keymap_name` ist der Name einer Datei aus `/usr/share/syscons/keymaps` ohne die Endung `.kbd`. Welche Tastenzuordnung Sie benutzen müssen, können Sie ohne einen Neustart mit kbdmap(1) ausprobieren.

Mit `keychange` können die Funktionstasten so programmiert werden, dass Sie zu dem ausgesuchten Terminal passen. Die Sequenzen der Funktionstasten können nicht in Tastenzuordnungen definiert werden.

Stellen Sie sicher, dass der richtige Terminaltyp für die `ttyv*` Konsolen in `/etc/ttys` angegeben ist. Momentan sind die folgenden Terminaltypen definiert:

Zeichensatz	Terminaltyp

Zeichensatz	Terminaltyp
ISO8859-1 oder ISO8859-15	`cons25l1`
ISO8859-2	`cons25l2`
ISO8859-7	`cons25l7`
KOI8-R	`cons25r`
KOI8-U	`cons25u`
CP437 (VGA default)	`cons25`
US-ASCII	`cons25w`

Mit Wide- oder Multibyte-Zeichensätzen müssen Sie den richtigen Port aus dem Verzeichnis `/usr/ports/`*`Sprache`* verwenden. Einige Ports erscheinen als Konsolen werden aber vom System als serielle vtty's betrachtet. Achten Sie daher darauf, dass Sie genügend vtty's für X11 und die Pseudo-seriellen Konsolen definiert haben. Nachstehend finden Sie eine unvollständige Liste der Ports, die eine andere Sprache als Englisch auf der Konsole verwenden:

Sprache	Port
traditionelles Chinesisch (BIG-5)	`chinese/big5con`
Japanisch	`japanese/kon2-16dot` oder `japanese/mule-freewnn`
Koreanisch	`korean/han`

24.3.6. Einrichten von X11

Obwohl X11 nicht Teil des FreeBSD Projects ist, stellen wir hier einige Hinweise für FreeBSD-Benutzer zusammen. Weitere Details entnehmen Sie bitte der Xorg Website (http://www.x.org/) oder der Dokumentation Ihres X11 Servers.

Anwendungsspezifische I18N-Einstellungen (Zeichensätze, Menüs, usw.) können Sie in `~/.Xresources` vornehmen.

24.3.6.1. Zeichensätze

Installieren Sie den **Xorg**-Server (`x11-servers/xorg-server`) und die TrueType Zeichensätze Ihrer Sprache. Wenn Sie die Locale gesetzt haben, sollten die Menüs in Ihrer Sprache erscheinen.

24.3.6.2. Eingabe von nicht-englischen Zeichen

Das X11 Input Method (XIM) Protokoll ist ein neuer Standard für alle X11-Clients. Jede X11-Anwendung sollte als XIM-Client, der Eingaben von einem XIM-Server entgegen nimmt, implementiert sein. XIM-Server sind für verschiedene Sprachen erhältlich.

24.3.7. Einrichten eines Druckers

Drucker verfügen normalerweise schon über einige C-char Zeichensätze[1]. Wide- oder Multibyte-Zeichensätze müssen gesondert eingerichtet werden. Wir empfehlen Ihnen, dazu **apsfilter** zu benutzen. Weiterhin können Sie mit

sprachspezifischen Konvertern Ihre Dokumente auch in PostScript oder PDF umwandeln.

24.3.8. Kernel und Dateisysteme

Das FreeBSD-Dateisystem (FFS) unterstützt 8-Bit, so dass es mit C-char Zeichensätzen[1] (siehe multibyte(3)) verwendet werden kann. Der Zeichensatz wird allerdings nicht im Dateisystem gespeichert, das heißt es werden nur die 8-Bit Werte gespeichert und die Codierung wird nicht berücksichtigt. Offiziell werden Wide- oder Multibyte-Zeichensätze noch nicht unterstützt, für einige Zeichensätze existieren Patche, die eine solche Unterstützung aktivieren. Sie sind allerdings nicht im Quelltext enthalten, da sie nur schwer pflegbare Übergangslösungen sind. Die Patche und weitere Informationen erhalten Sie auf den Webseiten der betreffenden Sprache.

Das MS-DOS Dateisystem von FreeBSD kann von MS-DOS- und Unicode-Zeichensätzen nach frei wählbaren FreeBSD Zeichensätzen konvertieren. Weitere Details entnehmen Sie bitte mount_msdosfs(8).

24.4. I18N-Programme übersetzen

Viele FreeBSD-Ports besitzen I18N-Unterstützung, einige davon enthalten `-I18N` im Namen. Für diese und viele andere Programme ist keine spezielle Konfiguration notwendig.

Einige Anwendungen wie **MySQL** müssen allerdings speziell für einen Zeichensatz in ihrem `Makefile` konfiguriert werden. Normalerweise wird dazu das `Makefile` angepasst oder **configure** mit einem speziellen Parameter aufgerufen.

24.5. Lokalisierung für einzelne Sprachen

24.5.1. Russisch (KOI8-R Codierung)

Beigetragen von Andrey Chernov.

Weitere Informationen über die KOI8-R Codierung erhalten Sie auf der Webseite KOI8-R References (Russian Net Character Set) (http://koi8.pp.ru/).

24.5.1.1. Einrichten der Locale

Fügen Sie die folgenden Zeilen in `~/.login_conf` ein:

```
me:My Account:\
       :charset=KOI8-R:\
       :lang=ru_RU.KOI8-R:
```

Weitere Erklärungen finden Sie in Einstellen der Locale.

24.5.1.2. Einrichten der Konsole

- Fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
mousechar_start=3
```

- Nehmen Sie zusätzlich die folgenden Einstellungen in `/etc/rc.conf` auf:

```
keymap="ru.koi8-r"
scrnmap="koi8-r2cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
```

- Benutzen Sie `cons25r` als Terminaltyp für jeden `ttyv*` Eintrag in `/etc/ttys`.

Weitere Beispiele finden Sie in Einrichten der Konsole.

24.5.1.3. Einrichten eines Druckers

Die meisten Drucker mit russischen Zeichen besitzen die Codetabelle CP866, so dass ein spezielles Programm zur Übersetzung von KOI8-R nach CP866 benötigt wird. Zu diesem Zweck ist `/usr/libexec/lpr/ru/koi2alt` im Basissystem enthalten. Der Eintrag für einen Drucker mit russischer Sprachunterstützung in `/etc/printcap` sieht wie folgt aus:

```
lp|Russian local line printer:\
       :sh:of=/usr/libexec/lpr/ru/koi2alt:\
       :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Näheres erfahren Sie in printcap(5).

24.5.1.4. MS-DOS Dateisystem und russische Dateinamen

Russische Dateinamen auf MS-DOS Dateisystemen werden mit dem folgenden Eintrag in `/etc/fstab` erkannt:

```
/dev/ad0s2      /dos/c  msdos   rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

Die Option `-L` legt die Locale fest. Die Option `-W` legt die Zeichenumwandlung fest. Stellen Sie sicher, dass `/usr` eingehangen ist, bevor Sie die MS-DOS-Partition einhängen, da die Tabellen zur Zeichenumwandlung in `/usr/libdata/msdosfs` liegen. Weitere Informationen erhalten Sie in der Hilfeseite mount_msdosfs(8).

24.5.1.5. Einrichten von X11

1. Richten Sie zunächst die normale Lokalisierung ein.
2. Wenn Sie **Xorg** verwenden, installieren Sie den Port `x11-fonts/xorg-fonts-cyrillic`.

   Im Abschnitt `"Files"` von `/etc/X11/xorg.conf` fügen Sie den folgende Eintrag *vor* allen anderen `FontPath` Einträgen ein:

   ```
   FontPath   "/usr/local/lib/X11/fonts/cyrillic"
   ```

   > **Anmerkung:** Zusätzliche kyrillische Schriftarten finden Sie in der Ports-Sammlung.

3. Die Unterstützung für eine russische Tastatur aktivieren Sie im `"Keyboard"` Abschnitt von `xorg.conf`:

   ```
   Option "XkbLayout"   "us,ru"
   Option "XkbOptions"  "grp:toggle"
   ```

   Stellen Sie zudem sicher, dass `XkbDisable` deaktiviert (auskommentiert) ist.

   Beim Einsatz von `grp:toggle` können Sie mit **Right Alt** (Alt Gr) zwischen dem RUS- und LAT-Modus wechseln, verwenden Sie hingegen `grp:ctrl_shift_toggle`, so erfolgt der Wechsel mit **Ctrl**+**Shift**. Für `grp:caps_toggle` ist zum Wechseln des RUS/LAT-Modus **CapsLock** zuständig. Die alte Funktion von **CapsLock** steht nur im LAT-Modus mit der Tastenkombination **Shift**+**CapsLock** zur Verfügung. `grp:caps_toggle` funktioniert aus unbekannten Gründen unter **Xorg** nicht.

   Wenn Ihre Tastatur Windows-Tasten besitzt und nicht-alphanumerische Tasten im RUS-Modus nicht funktionieren, fügen Sie die folgende Zeile in `xorg.conf` ein:

   ```
   Option "XkbVariant" ",winkeys"
   ```

   Anmerkung: Die russische XKB-Tastatur funktioniert vielleicht nicht mit nicht-lokalisierten Anwendungen.

Anmerkung: Lokalisierte Anwendungen sollten mindestens die Funktion `XtSetLanguageProc (NULL, NULL, NULL);` frühzeitig aufrufen.

Weitere Informationen über die Lokalisierung von X11-Anwendungen erhalten Sie auf der Webseite KOI8-R for X Window (http://koi8.pp.ru/xwin.html).

24.5.2. Traditionell chinesische Lokalisierung für Taiwan

Das taiwanesische FreeBSD Project stellt ein Tutorium unter http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/ zur Verfügung, das viele chinesische Anwendungen benutzt. Der Editor des `FreeBSD Chinese HOWTOs` ist Shen Chuan-Hsing <`statue@freebsd.sinica.edu.tw`>.

Chuan-Hsing Shen <`statue@freebsd.sinica.edu.tw`> hat mithilfe des Tutoriums die Chinese FreeBSD Collection (CFC) (http://netlab.cse.yzu.edu.tw/~statue/cfc/) geschaffen. Die Pakete und Skripten stehen unter ftp://freebsd.csie.nctu.edu.tw/pub/taiwan/CFC/.

24.5.3. Deutsche Lokalisierung (für alle ISO 8859-1 Sprachen)

Von Slaven Rezic <`eserte@cs.tu-berlin.de`> stammt ein Tutorium, das die Benutzung von Umlauten mit FreeBSD beschreibt. Das Tutorium ist in Deutsch verfasst und unter http://user.cs.tu-berlin.de/~eserte/FreeBSD/doc/umlaute/umlaute.html verfügbar.

24.5.4. Griechische Lokalisierung

Nikos Kokkalis <nickkokkalis@gmail.com> hat einen ganzen Artikel über die Griechisch-Unterstützung in FreeBSD geschrieben. Er ist als Teil der offiziellen FreeBSD Dokumentation auf Griechisch erhältlich unter http://www.freebsd.org/doc/el_GR.ISO8859-7/articles/greek-language-support/index.html (http://www.FreeBSD.org/doc/el_GR.ISO8859-7/articles/greek-language-support/index.html). Bitte beachten Sie, dass dies *nur* für Griechisch gilt.

24.5.5. Japanische und koreanische Lokalisierung

Informationen über die japanische Lokalisierung entnehmen Sie bitte http://www.jp.FreeBSD.org/, Informationen über die koreanische Lokalisierung erhalten Sie unter http://www.kr.FreeBSD.org/.

24.5.6. Nicht-englische FreeBSD-Dokumentation

Teile vor FreeBSD Dokumentation wurden in andere Sprachen übersetzt. Folgen Sie bitte den Links auf der FreeBSD-Webseite (http://www.FreeBSD.org/de/) oder schauen Sie in `/usr/share/doc` nach.

Fußnoten

1. Mit C-char Zeichensätzen werden Zeichensätze bezeichnet, die zur Codierung den C-Datentyp `char` verwenden.

Kapitel 25. FreeBSD aktualisieren

Umstrukturiert und aktualisiert von Jim Mock. Im Original von Jordan Hubbard, Poul-Henning Kamp, John Polstra und Nik Clayton. Übersetzt von Martin Heinen.

25.1. Übersicht

FreeBSD wird zwischen einzelnen Releases ständig weiter entwickelt. Manche Leute bevorzugen die offiziellen Release-Versionen, während andere wiederum lieber auf dem aktuellen Stand der Entwicklung bleiben möchten. Wie dem auch sei, sogar offizielle Release-Versionen werden oft mit Sicherheitsaktualisierungen und anderen kritischen Fehlerbereinigungen versorgt. Unabhängig von der eingesetzten Version bringt FreeBSD alle nötigen Werkzeuge mit, um ihr System aktuell zu halten und es innerhalb verschiedener Versionen zu aktualisieren. Dieses Kapitel hilft Ihnen bei der Entscheidung, ob Sie mit dem Entwicklungssystem Schritt halten oder ein Release verwenden wollen. Die zugrundeliegenden Werkzeuge um Ihr System aktuell zu halten werden ebenfalls vorgestellt.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- wissen, welche Werkzeuge verwendet werden können, um das System und die Port-Sammlung zu aktualisieren.
- wissen, wie Sie Ihr System mit **freebsd-update**, **CVSup**, **CVS** oder **CTM** aktualisieren.
- wissen, wie man das aktuell installierte System mit einer ursprünglichen Version vergleicht.
- wissen, wie Sie ihre Dokumentation mit **CVSup** oder Dokumentations-Ports aktuell halten können.
- den Unterschied zwischen den beiden Entwicklungszweigen FreeBSD-STABLE und FreeBSD-CURRENT kennen.
- Wissen, wie Sie das komplette Basissystem mit `make buildworld` neu bauen und installieren.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Ihr Netzwerk richtig konfiguriert haben (Kapitel 32) und
- wissen, wie Sie Software Dritter installieren (Kapitel 5).

> **Anmerkung:** Im gesamten Kapitel wird der Befehl `cvsup` verwendet, um die FreeBSD Quellen zu beziehen und zu aktualisieren. Um es zu verwenden, benötigen Sie einen Port oder ein Paket wie `net/cvsup` (falls Sie den graphischen `cvsup`-Client nicht benötigen, können Sie auch nur den Port `net/cvsup-without-gui` installieren). Alternativ können Sie auch csup(1) verwenden, das bereits Teil des Basissystems ist.

25.2. FreeBSD-Update

Geschrieben von Tom Rhodes. Basierend auf bereitgestellten Mitschriften von Colin Percival. Übersetzt von Benedict Reuschling.

Das Einspielen von Sicherheitsaktualisierungen ist ein wichtiger Bestandteil bei der Wartung von Computersoftware, besonders wenn es um das Betriebssystem geht. Für lange Zeit war dieser Prozess unter FreeBSD nicht einfach.

Fehlerbehebungen mussten auf den Quellcode angewendet werden, danach wurde der Code zu neuen Binärdateien übersetzt und schliesslich mussten diese Dateien neu installiert werden.

Das ist seit längerem nicht mehr der Fall, da FreeBSD jetzt ein Werkzeug namens `freebsd-update` enthält. Dieses Werkzeug bringt zwei getrennte Funktionen mit sich. Die erste Funktion ermöglicht die Anwendung von Sicherheitsaktualisierungen im Binärformat auf das FreeBSD Basissystem, ohne dieses neu zu übersetzen und zu installieren. Die zweite Funktion unterstützt Aktualisierungen zwischen Haupt- und Unterversionen.

> **Anmerkung:** Binäre Aktualisierungen sind für alle Architekturen und Releases verfügbar, die aktuell vom FreeBSD Security Team betreut werden. Vor der Aktualisierung auf eine neue Release-Version sollten die aktuellen Ankündigungen zu dem Release gelesen werden, da diese wichtige Informationen zu der gewünschten Version enthalten. Diese Ankündigungen finden Sie unter dem folgenden Link: http://www.FreeBSD.org/releases/.

Wenn eine `crontab` existiert, welche die Eigenschaften von `freebsd-update` verwendet, muss diese deaktiviert werden, bevor die folgende Aktion gestartet wird.

25.2.1. Die Konfigurationsdatei

Manche Anwender möchten sicherlich Einstellungen in der Standard-Konfigurationsdatei unter `/etc/freebsd-update.conf` vornehmen, um bessere Kontrolle über den gesamten Prozess zu besitzen. Die Optionen sind sehr gut dokumentiert, jedoch benötigen die folgenden ein paar zusätzliche Erklärungen:

```
# Components of the base system which should be kept updated.
Components src world kernel
```

Dieser Parameter kontrolliert, welche Teile von FreeBSD auf dem aktuellen Stand gehalten werden sollen. Die Voreinstellung ist es, den Quellcode zu aktualisieren, das gesamte Basissystem sowie den Kernel. Die Komponenten sind die gleichen wie während der Installation, also würde beispielsweise das hinzufügen von `world/games` an dieser Stelle es erlauben, Aktualisierungen für Spiele anzuwenden. Die Verwendung von `src/bin` erlaubt es, den Quellcode in `src/bin` aktuell zu halten.

Die beste Einstellung ist, diese Option so zu belassen, da eine Änderung es bedingt, dass man als Benutzer jede Komponente auflisten muss, die aktualisiert werden soll. Dies könnte katastrophale Folgen nach sich ziehen, da der Quellcode und die Binärdateien dadurch nicht mehr synchron wären.

```
# Paths which start with anything matching an entry in an IgnorePaths
# statement will be ignored.
IgnorePaths
```

Fügen Sie Pfade wie `/bin` oder `/sbin` hinzu, um diese speziellen Verzeichnisse während des Aktualisierungsprozesses unberührt zu lassen. Diese Option kann verwendet werden, um zu verhindern, dass `freebsd-update` lokale Änderungen überschreibt.

```
# Paths which start with anything matching an entry in an UpdateIfUnmodified
# statement will only be updated if the contents of the file have not been
# modified by the user (unless changes are merged; see below).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

Aktualisieren Sie Konfigurationsdateien in den angegebenen Verzeichnissen nur, wenn diese nicht geändert wurden. Jegliche Änderung, die der Benutzer daran vorgenommen hat, wird die automatische Aktualisierung dieser Dateien ungültig machen. Es gibt eine weitere Option `KeepModifiedMetadata`, die `freebsd-update` instruiert, die Änderungen während der Zusammenführung zu speichern.

```
# When upgrading to a new FreeBSD release, files which match MergeChanges
# will have any local changes merged into the version from the new release.
MergeChanges /etc/ /var/named/etc/
```

Eine Liste von Verzeichnissen mit Konfigurationsdateien, in denen `freebsd-update` Zusammenführungen versuchen soll. Dieser Verschmelzungsprozess von Dateien ist eine Serie von diff(1)-Korrekturen, ähnlich wie mergemaster(8) mit weniger Optionen. Die Änderungen werden entweder akzeptiert, öffnen einen Editor oder `freebsd-update` bricht ab. Wenn Sie im Zweifel sind, sichern Sie das `/etc` Verzeichnis und akzeptieren einfach die Änderungen. Lesen Sie Abschnitt 25.7.11.1, um Informationen über das `mergemaster`-Kommando zu erhalten.

```
# Directory in which to store downloaded updates and temporary
# files used by FreeBSD Update.
# WorkDir /var/db/freebsd-update
```

In diesem Verzeichnis werden alle Korrekturen und temporären Dateien abgelegt. Für Fälle in denen der Anwender eine Versionsaktualisierung vornimmt, sollte diesem Verzeichnis mindestens ein Gigabyte Festplattenspeicher zur Verfügung stehen.

```
# When upgrading between releases, should the list of Components be
# read strictly (StrictComponents yes) or merely as a list of components
# which *might* be installed of which FreeBSD Update should figure out
# which actually are installed and upgrade those (StrictComponents no)?
# StrictComponents no
```

Wenn dies auf `yes` gesetzt ist, wird `freebsd-update` annehmen, dass die `Components`-Liste vollständig ist und nicht versuchen, Änderungen ausserhalb dieser Liste zu tätigen. Tatsächlich wird `freebsd-update` versuchen, jede Datei zu aktualisieren, die zu der `Components`-Liste gehört.

25.2.2. Sicherheitsaktualisierungen

Sicherheitsaktualisierungen sind auf einer entfernten Maschine abgelegt und können durch das folgende Kommando heruntergeladen und installiert werden:

```
# freebsd-update fetch
# freebsd-update install
```

Wenn irgendeine Änderung auf den Kernel angewendet wurde benötigt das System einen Neustart. Wenn alles gut verlaufen ist, sollte das System aktualisiert sein und `freebsd-update` kann als nächtlicher cron(8)-Job ablaufen. Ein Eintrag in der Datei `/etc/crontab` ist für diese Aufgabe ausreichend:

```
@daily                                  root    freebsd-update cron
```

Dieser Eintrag besagt, dass einmal am Tag `freebsd-update` ausgeführt wird. Auf diese Weise kann `freebsd-update` nur durch die Verwendung des `cron`-Arguments prüfen, ob Aktualisierungen vorliegen. Wenn Korrekturen existieren, werden diese automatisch auf die lokale Festplatte heruntergeladen, aber nicht eingespielt. Der `root`-Benutzer bekommt eine Nachricht, damit dieser die Korrekturen manuell installiert.

Sollte irgendetwas schief gelaufen sein, kann `freebsd-update` den letzten Satz von Änderungen mit dem folgenden Befehl zurückrollen:

```
# freebsd-update rollback
```

Sobald dieser Vorgang abgeschlossen ist, sollte das System neu gestartet werden, wenn der Kernel oder ein beliebiges Kernelmodul geändert wurde. Dies ermöglicht es FreeBSD, die neuen Binärdateien in den Hauptspeicher zu laden.

Das `freebsd-update`-Werkzeug kann nur den `GENERIC`-Kernel automatisch aktualisieren. Wenn ein selbstkonfigurierter Kernel verwendet wird, muss dieser neu erstellt und installiert werden, nachdem `freebsd-update` den Rest der Aktualisierungen durchgeführt hat. Allerdings wird `freebsd-update` den `GENERIC`-Kernel in `/boot/GENERIC` erkennen und aktualisieren (falls dieser existiert), sogar dann, wenn dies nicht der aktuell verwendete Kernel des Systems ist.

> **Anmerkung:** Es ist eine gute Idee, immer eine Kopie des `GENERIC`-Kernels in `/boot/GENERIC` aufzubewahren. Das wird bei der Diagnose von verschiedenen Problemen eine grosse Hilfe sein, sowie bei der Durchführung von Versionsaktualisierungen mit `freebsd-update`, wie in Abschnitt 25.2.3 beschrieben ist.

Solange die Standardkonfiguration in `/etc/freebsd-update.conf` nicht geändert wurde, wird `freebsd-update` die aktualisierten Quellcodedateien des Kernels zusammen mit dem Rest der Neuerungen installieren. Die erneute Übersetzung und Installation ihres neuen, selbstkonfigurierten Kernels kann dann auf die übliche Art und Weise durchgeführt werden.

> **Anmerkung:** Die Aktualisierungen, die über `freebsd-update` verteilt werden, betreffen nicht immer den Kernel. Es ist nicht notwendig, den selbstkonfigurierten Kernel neu zu erstellen, wenn die Kernelquellen nicht durch die Ausführung von `freebsd-update install` geändert wurden. Allerdings wird `freebsd-update` auf alle Fälle die Datei `/usr/src/sys/conf/newvers.sh` aktualisieren. Der aktuelle Patch-Level (angegeben durch die `-p`-Nummer, die von dem Kommando `uname -r` ausgegeben wird) wird aus dieser Datei ausgelesen. Die Neuinstallation des selbstkonfigurierten Kernels, selbst wenn sich daran nichts geändert hat, erlaubt es uname(1), den aktuellen Patch-Level des Systems korrekt wiederzugeben. Dies ist besonders hilfreich, wenn mehrere Systeme gewartet werden, da es eine schnelle Einschätzung der installierten Aktualisierungen in jedem einzelnen System ermöglicht.

25.2.3. Aktualisierungen an Haupt- und Unterversionen

Dieser Prozess entfernt alte Objekt-Dateien und Bibliotheken, was dazu führt, dass die meisten Anwendungen von Drittherstellern nicht mehr funktionieren. Es wird empfohlen, dass alle installierten Ports entweder entfernt und neu installiert oder zu einem späteren Zeitpunkt mittels `ports-mgmt/portupgrade` aktualisiert werden. Die meisten Anwender werden wahrscheinlich einen Testlauf mittels des folgenden Kommandos durchführen wollen:

```
# portupgrade -af
```

Dies sorgt dafür, dass alles korrekt neu installiert wird. Beachten Sie, dass das Setzen der `BATCH`-Umgebungsvariable auf `yes` während dieses Prozesses auf jede Eingabe mit `ja` antwortet, was es nicht mehr notwendig macht, manuell eingreifen zu müssen.

Wenn ein selbstkonfigurierter Kernel verwendet wird, ist der Aktualisierungsprozess ein kleines bisschen aufwändiger. Eine Kopie des `GENERIC`-Kernels wir benötigt und sollte in `/boot/GENERIC` abgelegt sein. Wenn der `GENERIC`-Kernel nicht bereits im System vorhanden ist, kann dieser über eine der folgenden Methoden bezogen werden:

- Wenn ein eigener Kernel genau einmal gebaut wurde, ist der Kernel im Verzeichnis `/boot/kernel.old` in Wirklichkeit der `GENERIC`-Kernel. Benennen Sie einfach dieses Verzeichnis in `/boot/GENERIC` um.
- Angenommen, direkter Zugriff auf die Maschine ist möglich, so kann eine Kopie des `GENERIC`-Kernels von den CD-ROM-Medien installiert werden. Legen Sie die Installations-CD ein und benutzen Sie die folgenden Befehle:

  ```
  # mount /cdrom
  # cd /cdrom/X.Y-RELEASE/kernels
  # ./install.sh GENERIC
  ```

  Ersetzen Sie *`X.Y-RELEASE`* mit der richtigen Version der Veröffentlichung, die Sie verwenden. Der `GENERIC`-Kernel wird standardmässig in `/boot/GENERIC` installiert.
- Falls alle obigen Schritte fehlschlagen, kann der `GENERIC`-Kernel folgendermassen aus den Quellen neu gebaut und installiert werden:

  ```
  # cd /usr/src
  # env DESTDIR=/boot/GENERIC make kernel
  # mv /boot/GENERIC/boot/kernel/* /boot/GENERIC
  # rm -rf /boot/GENERIC/boot
  ```

  Damit dieser Kernel als `GENERIC`-Kernel von `freebsd-update` erkannt wird, darf die `GENERIC`-Konfigurationsdatei in keiner Weise geändert worden sein. Es wird ebenfalls empfohlen, dass dieser ohne irgendwelche speziellen Optionen erstellt wird (bevorzugt mit einer leeren `/etc/make.conf`).

Der Neustart in den `GENERIC`-Kernel ist zu diesem Zeitpunkt nicht notwendig.

Aktualisierungen an Haupt- und Unterversionen können durchgeführt werden, wenn man `freebsd-update` eine Release-Version als Ziel übergibt. Beispielsweise wird das folgende Kommando das System auf FreeBSD 8.1 aktualisieren:

```
# freebsd-update -r 8.1-RELEASE upgrade
```

Nachdem das Kommando empfangen wurde, überprüft `freebsd-update` die Konfigurationsdatei und das aktuelle System, um die nötigen Informationen für die Systemaktualisierung zu sammeln. Eine Bildschirmausgabe wird anzeigen, welche Komponenten erkannt und welche nicht erkannt wurden. Zum Beispiel:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 8.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
```

```
world/proflibs

Does this look reasonable (y/n)? y
```

An diesem Punkt wird `freebsd-update` versuchen, alle notwendigen Dateien für die Aktualisierung herunter zu laden. In manchen Fällen wird der Benutzer mit Fragen konfrontiert, um festzustellen, was installiert werden soll oder auf welche Art und Weise fortgesetzt werden soll.

Wenn ein selbstkonfigurierter Kernel benutzt wird, produziert der vorherige Schritt eine Warnung ähnlich zu der folgenden:

```
WARNING: This system is running a "MYKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 8.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

Diese Warnung kann an dieser Stelle problemlos ignoriert werden. Der aktualisierte `GENERIC`-Kernel wird als ein Zwischenschritt im Aktualisierungsprozess verwendet.

Nachdem alle Korrekturen auf das lokale System heruntergeladen wurden, werden diese nun eingespielt. Dieser Prozess kann eine gewisse Zeit in Anspruch nehmen, abhängig von der Geschwindigkeit und Auslastung der Maschine. Konfigurationsdateien werden ebenfalls zusammengefügt - dieser Teil der Prozedur benötigt einige Benutzereingaben, da eine Datei möglicherweise von Hand zusammengefasst werden muss oder ein Editor erscheint auf dem Bildschirm zum manuellen bearbeiten. Die Ergebnisse von jeder erfolgreichen Zusammenfassung werden dem Benutzer angezeigt, während der Prozess weiterläuft. Eine fehlgeschlagene oder ignorierte Zusammenfassung wird den Prozess sofort beenden. Benutzer sollten eine Sicherung von `/etc` anlegen und wichtige Dateien später manuell vereinen, beispielsweise `master.passwd` oder `group`.

> **Anmerkung:** Das System ist noch nicht verändert worden, alle Korrekturen und Vereinigungen sind in einem anderen Verzeichnis vorgenommen worden. Wenn alle Korrekturen erfolgreich eingespielt, alle Konfigurationsdateien zusammengefügt wurden und es den Anschein hat, dass der Prozess problemlos verlaufen wird, müssen die Änderungen vom Anwender noch angewendet werden.

Sobald dieser Prozess abgeschlossen ist, können die Aktualisierungen über das folgende Kommando auf die Platte geschrieben werden:

```
# freebsd-update install
```

Der Kernel und die Module werden zuerst aktualisiert. Zu diesem Zeitpunkt muss die Maschine neu gestartet werden. Wenn das System einen selbstkonfigurierten Kernel verwendet, benutzen Sie das nextboot(8)-Kommando, um den Kernel für den nächsten Neustart auf `/boot/GENERIC` zu setzen (welcher aktualisiert wurde):

```
# nextboot -k GENERIC
```

> **Warnung:** Bevor mit dem `GENERIC`-Kernel das System neu gestartet wird, vergewissern Sie sich, dass alle notwendigen Treiber für ihr System enthalten sind, um korrekt zu starten (und schliessen Sie ihn ans Netzwerk an, falls auf die Maschine, die aktualisiert wird, von der Ferne aus zugegriffen wird). Achten Sie besonders darauf, dass wenn der vorherige selbstkonfigurierte Kernel Funktionalität beinhaltet, die von Kernelmodulen zur Verfügung gestellt wurde, dass diese temporär in den `GENERIC`-Kernel über die Datei `/boot/loader.conf`

übernommen werden. Sie sollten ebenfalls nicht benötigte Dienste, eingehängte Platten, verbundene Netzlaufwerke, usw. deaktivieren, bis der Aktualisierungsprozess abgeschlossen ist.

Die Maschine sollte nun mit dem aktualisierten Kernel neu gestartet werden:

```
# shutdown -r now
```

Sobald das System wieder hochgefahren wurde, muss `freebsd-update` erneut gestartet werden. Der Zustand des Prozesses wurde zuvor gesichert und deshalb wird `freebsd-update` nicht von vorne beginnen, jedoch alle alten Shared-Libraries und Objektdateien löschen. Um zu diesem Zustand zu gelangen, setzen Sie das folgende Kommando ab:

```
# freebsd-update install
```

Anmerkung: Abhängig davon, ob irgendwelche Bibliotheksversionen erhöht wurden, kann es sein, dass nur zwei Installationsphasen anstatt drei durchlaufen werden.

Nun muss alle Drittanbieter-Software neu erstellt und neu installiert werden. Dies ist notwendig, da die installierte Software möglicherweise Abhängigkeiten zu Bibliotheken enthält, die während der Aktualisierung entfernt wurden. Der `ports-mgmt/portupgrade`-Befehl kann verwendet werden, um diesen Vorgang zu automatisieren. Die folgenden Kommandos können verwendet werden, um diesen Prozess zu starten:

```
# portupgrade -f ruby
# rm /var/db/pkg/pkgdb.db
# portupgrade -f ruby18-bdb
# rm /var/db/pkg/pkgdb.db /usr/ports/INDEX-*.db
# portupgrade -af
```

Sobald dies abgeschlossen ist, beenden Sie den Aktualisierungsprozess mit einem letzten Aufruf von `freebsd-update`. Geben Sie den folgenden Befehl ein, um alle losen Enden des Aktualisierungsprozesses miteinander zu verknüpfen:

```
# freebsd-update install
```

Wenn der `GENERIC`-Kernel temporär Verwendung fand, ist dies der richtige Zeitpunkt, einen neuen, selbstkonfigurierten Kernel zu bauen und über die übliche Methode zu installieren.

Booten Sie anschliessend die Maschine in die neue FreeBSD-Version. Der Prozess ist damit abgeschlossen.

25.2.4. Vergleich des Systemzustands

Das `freebsd-update`-Werkzeug kann verwendet werden, um den Zustand der installierten FreeBSD-Version gegenüber einer bekannten und funktionierenden Kopie zu vergleichen. Diese Option vergleicht die aktuelle Version von Systemwerkzeugen, Bibliotheken und Konfigurationsdateien. Um diesen Vergleich zu starten, geben Sie den folgenden Befehl ein:

```
# freebsd-update IDS >> outfile.ids
```

> **Warnung:** Obwohl der Befehlsname IDS lautet, sollte er in keiner Weise als Ersatz für ein Intrusion Detection System wie `security/snort` angesehen werden. Da `freebsd-update` seine Daten auf Platte ablegt, ist die Möglichkeit von Verfälschungen offensichtlich. Obwohl diese Möglichkeit durch die Verwendung von `kern.securelevel` oder die Ablage von `freebsd-update` auf einem Nur-Lese Dateisystem, wenn es gerade nicht gebraucht wird, eingedämmt werden kann, besteht eine bessere Lösung darin, das System gegen ein gesichertes Medium, wie eine DVD oder einen externen, separat aufbewahrten USB-Plattenspeicher, zu vergleichen.

Das System wird jetzt untersucht und eine Liste von Dateien ausgegeben, zusammen mit deren sha256(1)-Hashwerten, sowohl der von der Release-Version bekannte Wert als auch der des aktuell installierten Systems. Das ist der Grund dafür, warum die Ausgabe an die Datei `outfile.ids` geschickt wurde. Es scrollt zu schnell vorbei, um diese mit den Augen zu vergleichen und bald wird auch der Konsolenpuffer damit überfüllt.

Diese Zeilen sind dazu noch extrem lang, aber das Ausgabeformat kann sehr einfach verarbeitet werden. Um beispielsweise eine Liste von allen Dateien zu erhalten, die sich vom aktuellen Release unterscheiden, geben Sie das folgende Kommando ein:

```
# cat outfile.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

Diese Ausgabe wurde abgeschnitten, es existieren noch viel mehr Dateien dazu. Manche dieser Dateien besitzen ganz selbstverständliche Veränderungen, `/etc/passwd` wurde beispielsweise geändert, um Benutzer zum System hinzuzufügen. In manchen Fällen kann es anderen Dateien wie Kernelmodule geben, welche sich geändert haben, weil `freebsd-update` diese aktualisiert hat. Um bestimmte Dateien oder Verzeichnisse auszuschliessen, hängen Sie diese an die `IDSIgnorePaths`-Option in `/etc/freebsd-update.conf` an.

Diese Vorgehensweise kann als Teil einer ausgeklügelten Aktualisierungsmethode benutzt werden, unabhängig von der zuvor angesprochenen Variante.

25.3. Portsnap: Ein Werkzeug zur Aktualisierung der Ports-Sammlung

Geschrieben von Tom Rhodes. Basierend auf bereitgestellten Mitschriften von Colin Percival. Übersetzt von Benedict Reuschling.

Das Basissystem von FreeBSD enthält auch ein Programm zum Aktualisieren der Ports-Sammlung: das portsnap(8) Werkzeug. Wenn es ausgeführt wird, verbindet es sich mit einem entfernten Rechner, überprüft den Sicherungsschlüssel und lädt eine neue Kopie der Ports-Sammlung herunter. Der Schlüssel wird dazu verwendet, um die Integrität aller heruntergeladenen Dateien zu prüfen und um sicherzustellen, dass diese unterwegs nicht verändert wurden. Um die aktuellsten Dateien der Ports-Sammlung herunter zu laden, geben Sie das folgende Kommando ein:

```
# portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 3 mirrors found.
Fetching snapshot tag from portsnap1.FreeBSD.org... done.
Fetching snapshot metadata... done.
```

```
Updating from Wed Aug  6 18:00:22 EDT 2008 to Sat Aug 30 20:24:11 EDT 2008.
Fetching 3 metadata patches.. done.
Applying metadata patches... done.
Fetching 3 metadata files... done.
Fetching 90 patches.....10....20....30....40....50....60....70....80....90. done.
Applying patches... done.
Fetching 133 new ports or files... done.
```

Dieses Beispiel zeigt, dass portsnap(8) mehrere Korrekturen für die aktuellen Ports-Daten gefunden und verifiziert hat. Es zeigt auch, dass das Programm zuvor schon einmal gestartet wurde. Wäre es das erste Mal, würde nur die Ports-Sammlung heruntergeladen werden.

Wenn portsnap(8) erfolgreich die `fetch`-Operation abgeschlossen hat, befinden sich die Ports-Sammlung und die dazugehörigen Korrekturen auf dem lokalen System, welches die Überprüfung bestanden hat. Wenn Sie `portsnap` das erste Mal ausgeführt haben, müssen Sie den Befehl `portsnap extract` verwenden, um die Ports-Sammlung zu installieren:

```
# portsnap extract
/usr/ports/.cvsignore
/usr/ports/CHANGES
/usr/ports/COPYRIGHT
/usr/ports/GIDs
/usr/ports/KNOBS
/usr/ports/LEGAL
/usr/ports/MOVED
/usr/ports/Makefile
/usr/ports/Mk/bsd.apache.mk
/usr/ports/Mk/bsd.autotools.mk
/usr/ports/Mk/bsd.cmake.mk
...
```

Um Ihre bereits installierte Ports-Sammlung zu aktualisieren, verwenden Sie hingegen den Parameter `update`:

```
# portsnap update
```

Der Prozess ist jetzt abgeschlossen und Anwendungen können mittels der aktuellen Ports-Sammlung installiert oder aktualisiert werden.

Die Operationen `fetch` und `extract` oder `update` können auch nacheinander ausgeführt werden, wie im folgenden Beispiel gezeigt:

```
# portsnap fetch update
```

Dieser Befehl lädt die aktuelle Version der Ports-Sammlung herunter und aktualisiert anschließend Ihre lokale Version im Verzeichnis `/usr/ports`.

25.4. Aktualisieren der Dokumentationssammlung

Übersetzt von Benedict Reuschling.

Neben dem Basissystem und der Ports-Sammlung ist die Dokumentation ein wichtiger Bestandteil des FreeBSD Betriebssystems. Obwohl eine aktuelle Version der FreeBSD Dokumentation jederzeit auf der FreeBSD Webseite

(http://www.freebsd.org/doc/) verfügbar ist, verfügen manche Benutzer nur über eine langsame oder überhaupt keine Netzwerkverbindung. Glücklicherweise gibt es mehrere Möglichkeiten, die Dokumentation, welche mit jeder Version ausgeliefert wird, zu aktualisieren, indem eine lokale Kopie der aktuellen FreeBSD-Dokumentationssammlung verwendet wird.

25.4.1. Verwenden von CVSup um die Dokumentation zu aktualisieren

Die Quellen und die installierte Kopie der FreeBSD Dokumentation kann mittels **CVSup** aktualisiert werden, indem ein ähnlicher Mechanismus angewendet wird, wie derjenige für die Betriebssystemquellen (vergleichen Sie mit Abschnitt 25.7). Dieser Abschnitt beschreibt:

- Wie die Dokumentations-Werkzeugsammlung installiert wird, welche die Werkzeuge enthält, die nötig sind, um die FreeBSD Dokumentation aus den Quellen neu zu erstellen.
- Wie man eine Kopie der Dokumentationsquellen nach `/usr/doc` herunterlädt, unter Verwendung von **CVSup**.
- Wie man die FreeBSD Dokumentation aus den Quellen baut und unter `/usr/share/doc` installiert.
- Manche der Optionen zum Erstellen, die vom System zum Bauen der Dokumentation unterstützt werden, z.B. die Optionen welche nur ein paar der unterschiedlichen Sprachübersetzungen der Dokumentation erstellen oder die Optionen, die ein bestimmtes Ausgabeformat auswählen.

25.4.2. CVSup und die Werkzeugsammlung der Dokumentation installieren

Die FreeBSD Dokumentation aus dem Quellen zu erstellen benötigt eine ziemlich grosse Anzahl an Werkzeugen. Diese Werkzeuge sind nicht Teil des FreeBSD Basissystems, da sie eine grosse Menge an Plattenplatz verbrauchen und nicht von allen FreeBSD-Anwendern benötigt werden. Sie sind nur für diejenigen Benutzer notwendig, die aktiv an neuer Dokumentation fü FreeBSD schreiben oder häufig ihre Dokumentation aus den Quellen bauen lassen.

Alle benötigten Werkzeuge sind als Teil der Ports-Sammlung verfügbar. Der Port `textproc/docproj` dient als Masterport, der vom FreeBSD Documentation Project entwickelt wurde, um die initiale Installation und zukünftige Aktualisierungen dieser Werkzeuge zu vereinfachen.

> **Anmerkung:** Wenn Sie die Dokumentation nicht als PostScript oder PDF benötigen, können Sie alternativ die Installation des `textproc/docproj-nojadetex`-Ports in Erwägung ziehen. Diese Version der Dokumentations-Werkzeugsammlung enthält alles ausser das **teTeX**-Textsatzsystem. **teTeX** ist eine sehr grosse Sammlung an Werkzeugen, deshalb ist es vernünftig, deren Installation auszulassen, wenn die Ausgabe von PDF nicht unbedingt gebraucht wird.

Für weitere Informationen über das Installieren und Verwenden von **CVSup**, lesen Sie CVSup verwenden.

25.4.3. Die Dokumentationsquellen aktualisieren

Das Programm **CVSup** kann eine saubere Kopie der Dokumentationsquellen holen, indem es die Datei `/usr/share/examples/cvsup/doc-supfile` als Konfigurationsvorlage verwendet. Der Standard-Host zum Aktualisieren ist auf einen Platzhalterwert im `doc-supfile` gesetzt, aber cvsup(1) akzeptiert auch einen Hostnamen über die Kommandozeile. Somit können die Dokumentationsquellen von einem der **CVSup**-Server geholt werden, indem man eingibt:

```
# cvsup -h cvsup.FreeBSD.org -g -L 2 /usr/share/examples/cvsup/doc-supfile
```

Ändern Sie *cvsup.FreeBSD.org* auf den Ihnen am nächsten gelegenen **CVSup**-Server. Eine vollständige Liste von Spiegelservern finden Sie unter Abschnitt A.6.7.

Es dauert eine Weile, wenn die Dokumentationsquellen das allererste Mal heruntergeladen werden. Lassen Sie es laufen, bis es fertig ist.

Zukünftige Aktualisierungen der Dokumentationsquellen können Sie über den gleichen Befehl bekommen. Das Programm **CVSup** lädt und kopiert nur diejenigen Aktualisierungen herunter, die seit seinem letzten Aufruf hinzugekommen sind. Deshalb sollte jeder weitere Aufruf von **CVSup** nach dem Ersten wesentlich schneller abgeschlossen sein.

Nachdem die Quellen einmal ausgecheckt wurden, besteht ein anderer Weg, die Dokumentation zu aktualisieren, darin, das `Makefile` im Verzeichnis `/usr/doc` anzupassen. Durch setzen von `SUP_UPDATE`, `SUPHOST` und `DOCSUPFILE` in der Datei `/etc/make.conf` ist es jetzt möglich, folgendes zu tun:

```
# cd /usr/doc
# make update
```

Ein typischer Satz dieser make(1)-Optionen für `/etc/make.conf` ist:

```
SUP_UPDATE= yes
SUPHOST?= cvsup.freebsd.org
DOCSUPFILE?= /usr/share/examples/cvsup/doc-supfile
```

> **Anmerkung:** Das Setzen des Werts von `SUPHOST` und `DOCSUPFILE` auf `?=` erlaubt es, diese in der Kommandozeile von make zu überschreiben. Diese Methode wird empfohlen, um Optionen zu `make.conf` hinzuzufügen, um zu verhindern, dass man die Datei jedes Mal bearbeiten muss, um einen anderen Wert für die Option auszuprobieren.

25.4.4. Einstellbare Optionen der Dokumentationsquellen

Das System zum aktualisieren und erstellen der FreeBSD-Dokumentation unterstützt ein paar Optionen, welche den Prozess der Aktualisierung von Teilen der Dokumentation oder einer bestimmten Übersetzung erleichtert. Diese Optionen lassen sich entweder systemweit in der Datei `/etc/make.conf` setzen, oder als Kommandozeilenoptionen, die dem make(1)-Werkzeug übergeben werden.

Die folgenden Optionen sind ein paar davon:

`DOC_LANG`

Eine Liste von Sprachen und Kodierungen, die gebaut und installiert werden sollen, z.B. `en_US.ISO8859-1`, um nur die englische Dokumentation zu erhalten.

`FORMATS`

Ein einzelnes Format oder eine Liste von Ausgabeformaten, das gebaut werden soll. Momentan werden `html`, `html-split`, `txt`, `ps`, `pdf`, und `rtf` unterstützt.

`SUPHOST`

Der Hostname des **CVSup**-Servers, der verwendet werden soll, um Aktualisierungen zu holen.

`DOCDIR`

Wohin die Dokumentation installiert werden soll. Der Standardpfad ist `/usr/share/doc`.

Für weitere make-Variablen, die als systemweite Optionen in FreeBSD unterstützt werden, lesen Sie make.conf(5).

Für weitere make-Variablen, die vom System zum Erstellen der FreeBSD-Dokumentation unterstützt werden, lesen Sie die Fibel für neue Mitarbeiter des FreeBSD-Dokumentationsprojekts (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/fdp-primer).

25.4.5. Die FreeBSD-Dokumentation aus den Quellen installieren

Wenn ein aktueller Schnappschuss der Dokumentationsquellen nach `/usr/doc` heruntergeladen wurde, ist alles bereit für eine Aktualisierung der bestehenden Dokumentation.

Eine komplette Aktualisierung aller Sprachoptionen, definiert durch die `DOC_LANG` Makefile-Option, kann durch folgende Eingabe erreicht werden:

```
# cd /usr/doc
# make install clean
```

Wenn `make.conf` mit den richtigen Optionen `DOCSUPFILE`, `SUPHOST` und `SUP_UPDATE` eingerichtet wurde, kann der Installationsschritt mit einer Aktualisierung der Dokumentationsquellen kombiniert werden, indem man eingibt:

```
# cd /usr/doc
# make update install clean
```

Wenn nur eine Aktualisierung einer bestimmten Sprache gewünscht wird, kann make(1) in einem sprachspezifischen Unterverzeichnis von `/usr/doc` aufgerufen werden, z.B.:

```
# cd /usr/doc/en_US.ISO8859-1
# make update install clean
```

Die zu installierenden Ausgabeformate können durch das Setzen der make-Variablen `FORMATS` angegeben werden, z.B.:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

25.4.6. Verwendung von Dokumentations-Ports

Basierend auf der Arbeit von Marc Fonvieille.

Im vorherigen Abschnitt wurde eine Methode gezeigt, wie die FreeBSD-Dokumentation aus den Quellen gebaut werden kann. Allerdings sind quellbasierte Aktualisierungen möglicherweise nicht für alle FreeBSD-Systeme geeignet oder praktikabel. Das Erstellen der Dokumentationsquellen benötigt eine grosse Anzahl an Werkzeugen, Programmen und Hilfsmitteln, die *documentation toolchain*, ein gewisser Grad an Vertrautheit mit **CVS** und ausgecheckte Quellen von einem Repository, sowie ein paar manuelle Schritte, um diese ausgecheckten Quellen zu

bauen. In diesem Abschnitt wird eine alternative Art und Weise vorgestellt, wie man die installierte Kopie der FreeBSD-Dokumentation aktualisieren kann. Diese Methode verwendet die Ports-Sammlung und erlaubt es:

- vorgefertigte Schnappschüsse der Dokumentation herunter zu laden und zu installieren, ohne vorher irgendetwas lokal zu erstellen (dadurch ist es nicht mehr notwendig, den kompletten Werkzeugkasten der Dokumentation zu installieren).
- die Dokumentationsquellen herunterzuladen und durch das Ports-System erstellen zu lassen (was die Schritte zum Auschecken und Erstellen etwas erleichtert).

Diese beiden Methoden der Aktualisierung der FreeBSD-Dokumentation werden durch eine Menge von *Dokumentations-Ports* unterstützt, die von Documentation Engineering Team <doceng@FreeBSD.org> monatlich aktualisiert wird. Diese sind in der Ports-Sammlung unter der virtuellen Kategorie, docs (http://www.freshports.org/docs/) genannt, gelistet.

25.4.6.1. Erstellen und Installieren von Dokumentations-Ports

Die Dokumentations-Ports nutzen das Ports-System, um das Erstellen von Dokumentation wesentlich einfacher zu machen. Es automatisiert den Prozess des Auscheckens der Dokumentationsquellen, aufrufen von make(1) mit den passenden Umgebungsvariablen und Kommandozeilenoptionen und macht die Installation und Deinstallation von Dokumentation so einfach wie die Installation von jedem anderen Port oder Paket.

> **Anmerkung:** Als zusätzliche Eigenschaft zeichnen sie eine Abhängigkeit zum *Dokumentations-Werkzeugsatz* auf, wenn die Dokumentations-Ports lokal erstellt werden, weshalb dieser auch automatisch mitinstalliert wird.

Die Dokumentations-Ports sind wie folgt organisiert:

- Es existiert ein “Master-Port”, `misc/freebsd-doc-en`, in dem alle Dateien zu den Dokumentations-Ports abgelegt sind. Es dient als Basis für alle Dokumentations-Ports. Als Voreinstellung wird nur die englische Dokumentation gebaut.
- Es gibt einen “Alles-in-Einem-Port”, `misc/freebsd-doc-all`, welcher die komplette Dokumentation in allen verfügbaren Sprachen erstellt und installiert.
- Schliesslich gibt es noch einen sogenannten “slave port” für jede Übersetzung, z.B.: `misc/freebsd-doc-hu` für Dokumentation in ungarischer Sprache. All diese benötigen den Master-Port und installieren die übersetzte Dokumentation in der entsprechenden Sprache.

Um einen Dokumentations-Port aus den Quellen zu installieren, geben Sie das folgende Kommando (als `root`) ein:

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

Auf diese Weise wird die englische Dokumentation gebaut und als getrenntes HTML-Format im Verzeichnis `/usr/local/share/doc/freebsd` installiert (genau wie unter http://www.FreeBSD.org zu finden).

25.4.6.1.1. Gebräuchliche Schalter und Optionen

Es gibt viele Optionen, um das Standarderhalten der Dokumentations-Ports zu verändern. Im Folgenden sind nur ein paar davon aufgeführt:

`WITH_HTML`

Erlaubt das Erstellen im HTML-Format: eine einzige HTML-Datei pro Dokument. Die formatierte Dokumentation wird als Datei mit dem Namen `article.html` gespeichert, oder, je nachdem, als `book.html`, zuzuüglich der Bilder.

`WITH_PDF`

Erlaubt das Erstellen von Adobe Portable Document Format, für die Verwendung mit Adobe Acrobat Reader, **Ghostscript** oder anderen PDF-Betrachtern. Die formatierte Dokumentation wird als Datei mit dem Namen `article.pdf` oder, soweit angemessen, als `book.pdf` gespeichert.

`DOCBASE`

Wohin die Dokumentation installiert werden soll. Der Standardpfad ist `/usr/local/share/doc/freebsd`.

> **Anmerkung:** Beachten Sie, dass sich der Standardpfad von dem Verzeichnis unterscheidet, das von der **CVSup**-Methode verwendet wird. Das liegt daran, dass ein Port installiert wird und diese üblicherweise im Verzeichnis `/usr/local` abgelegt werden. Durch setzen der `PREFIX`-Variablen kann dieses Verhalten geändert werden.

Es folgt ein kurzes Beispiel, wie die Variablen verwendet werden, um die oben erwähnte ungarische Dokumentation als Portable Document Format zu installieren:

```
# cd /usr/ports/misc/freebsd-doc-hu
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

25.4.6.2. Verwendung von Dokumentations-Paketen

Das Erstellen der Dokumentations-Ports aus den Quellen, wie im vorherigen Abschnitt beschrieben, benötigt die lokale Installation der Dokumentations-Werkzeugsammlung und ein wenig Festplattenspeicher für das Bauen der Ports. Sollten die Ressourcen zum Bauen der Dokumentations-Werkzeugsammlung nicht zur Verfügung stehen, oder weil das erstellen zuviel Plattenplatz benötigen würde, ist es trotzdem möglich, bereits zuvor gebaute Schnappschüsse der Dokumentations-Ports zu installieren.

Documentation Engineering Team <`doceng@FreeBSD.org`> erstellt monatliche Schnappschüsse der Dokumentations-Pakete von FreeBSD. Diese Binärpakete können mit jedem der mitgelieferten Paketwerkzeuge installiert werden, beispielsweise pkg_add(1), pkg_delete(1) und so weiter.

> **Anmerkung:** Wenn Binärpakete zu Einsatz kommen, wird die FreeBSD-Dokumentation in *allen* verfügbaren Formaten in der gegebenen Sprache installiert.

Zum Beispiel installiert das folgende Kommando das aktuelle, vorgefertigte Paket der ungarischen Dokumentation:

```
# pkg_add -r hu-freebsd-doc
```

Anmerkung: Pakete haben das folgende Namensformat, welches sich von dem Namen des dazugehörigen Ports unterscheidet: *`lang`*`-freebsd-doc`. *`lang`* entspricht hier der Kurzform des Sprachcodes, z.B. `hu` für Ungarisch, oder `zh_cn` für vereinfachtes Chinesisch.

25.4.6.3. Dokumentations-Ports aktualisieren

Um einen zuvor installierten Dokumentations-Port zu aktualisieren, kann jedes Werkzeug, das auch zum Aktualisieren von Ports verwendet wird, eingesetzt werden. Beispielsweise aktualisiert das folgende Kommando die installierte ungarische Dokumentation mittels des Programms `ports-mgmt/portupgrade` indem nur Pakete verwendet werden sollen:

```
# portupgrade -PP hu-freebsd-doc
```

25.5. Einem Entwicklungszweig folgen

FreeBSD besitzt zwei Entwicklungszweige: FreeBSD-CURRENT und FreeBSD-STABLE. Dieser Abschnitt beschreibt beide Zweige und erläutert, wie Sie Ihr System auf dem aktuellen Stand eines Zweiges halten. Zuerst wird FreeBSD-CURRENT vorgestellt, dann FreeBSD-STABLE.

25.5.1. FreeBSD-CURRENT

Beachten Sie im Folgenden, dass FreeBSD-CURRENT die Spitze der Entwicklung von FreeBSD ist. Benutzer von FreeBSD-CURRENT sollten über sehr gute technische Fähigkeiten verfügen und in der Lage sein, schwierige Probleme alleine zu lösen. Wenn FreeBSD neu für Sie ist, überlegen Sie sich genau, ob Sie FreeBSD-CURRENT benutzen wollen.

25.5.1.1. Was ist FreeBSD-CURRENT?

FreeBSD-CURRENT besteht aus den neuesten Quellen des FreeBSD-Systems. Es enthält Sachen, an denen gerade gearbeitet wird, experimentelle Änderungen und Übergangsmechanismen, die im nächsten offiziellen Release der Software enthalten sein können oder nicht. Obwohl FreeBSD-CURRENT täglich von vielen Entwicklern gebaut wird, gibt es Zeiträume, in denen sich das System nicht bauen lässt. Diese Probleme werden so schnell wie möglich gelöst, aber ob Sie mit FreeBSD-CURRENT Schiffbruch erleiden oder die gewünschten Verbesserungen erhalten, kann von dem Zeitpunkt abhängen, an dem Sie sich den Quelltext besorgt haben!

25.5.1.2. Wer braucht FreeBSD-CURRENT?

FreeBSD-CURRENT wird hauptsächlich für 3 Interessengruppen zur Verfügung gestellt:

1. Entwickler, die an einem Teil des Quellbaums arbeiten und daher über die aktuellen Quellen verfügen müssen.
2. Tester, die bereit sind, Zeit in das Lösen von Problemen zu investieren und sicherstellen, dass FreeBSD-CURRENT so stabil wie möglich bleibt. Weiterhin Leute, die Vorschläge zu Änderungen oder der generellen Entwicklung von FreeBSD machen und Patches bereitstellen, um diese Vorschläge zu realisieren.

3. Für Leute, die die Entwicklung im Auge behalten wollen, oder die Quellen zu Referenzzwecken (zum Beispiel darin lesen, aber nicht verwenden) benutzen wollen. Auch diese Gruppe macht Vorschläge oder steuert Quellcode bei.

25.5.1.3. Was FreeBSD-CURRENT *nicht* ist!

1. Der schnellste Weg, neue Sachen vor dem offiziellen Release auszuprobieren. Bedenken Sie, dass der erste, der die neuen Sachen ausprobiert, auch der erste ist, der die neuen Fehler findet.
2. Ein schneller Weg, um an Fehlerbehebungen (engl. *bug fixes*) zu kommen. Jede Version von FreeBSD-CURRENT führt mit gleicher Wahrscheinlichkeit neue Fehler ein, mit der sie alte behebt.
3. In irgendeiner Form “offiziell unterstützt”. Wir tun unser Bestes, um Leuten aus den drei “legitimen” Benutzergruppen von FreeBSD-CURRENT zu helfen, aber wir *haben einfach nicht die Zeit*, technische Unterstützung zu erbringen. Das kommt nicht daher, dass wir kleinliche, gemeine Leute sind, die anderen nicht helfen wollen (wenn wir das wären, würden wir FreeBSD nicht machen), wir können einfach nicht jeden Tag Hunderte Nachrichten beantworten *und* an FreeBSD arbeiten! Vor die Wahl gestellt, FreeBSD zu verbessern oder jede Menge Fragen zu experimentellem Code zu beantworten, haben sich die Entwickler für ersteres entschieden.

25.5.1.4. Benutzen von FreeBSD-CURRENT

1. Es ist *essentiell*, die Mailinglisten freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) und svn-src-head (http://lists.FreeBSD.org/mailman/listinfo/svn-src-head) zu lesen. Wenn Sie *freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)* nicht lesen, verpassen Sie die Kommentare anderer über den momentanen Zustand des Systems und rennen demzufolge in viele bekannte Probleme, die schon gelöst sind. Noch kritischer ist, dass Sie wichtige Bekanntmachungen verpassen, die erhebliche Auswirkungen auf die Stabilität Ihres Systems haben können.

   In der svn-src-head (http://lists.FreeBSD.org/mailman/listinfo/svn-src-head) Mailingliste sehen Sie zu jeder Änderung das Commit-Log, das Informationen zu möglichen Seiteneffekten enthält.

   Um diese Listen zu abonnieren (oder zu lesen) besuchen Sie bitte die Seite http://lists.FreeBSD.org/mailman/listinfo. Weitere Informationen erhalten Sie, wenn Sie dort auf die gewünschte Liste klicken. Wenn Sie daran interessiert sind, die Änderungen am gesamten Quellbaum mit zu verfolgen, schlagen wir vor, die Liste svn-src-all (http://lists.FreeBSD.org/mailman/listinfo/svn-src-all) zu abonnieren.
2. Beschaffen Sie sich die Quellen von einem FreeBSD-Spiegel. Sie haben dazu zwei Möglichkeiten:
   a. Benutzen Sie das Programm cvsup mit der Datei `standard-supfile` aus dem Verzeichnis `/usr/share/examples/cvsup`. Dies ist die empfohlene Methode, da Sie die ganzen Quellen nur einmal herunterladen und danach nur noch Änderungen beziehen. Viele lassen `cvsup` aus `cron` heraus laufen, um ihre Quellen automatisch auf Stand zu bringen. Sie müssen die obige Sup-Datei anpassen und cvsup in Ihrer Umgebung konfigurieren.

      Anmerkung: Die `standard-supfile`-Beispieldatei ist dafür vorgesehen, einen bestimmten Sicherheitszweig zu verfolgen und nicht FreeBSD-CURRENT. Sie müssen diese Datei bearbeiten und die folgende Zeile:

      ```
      *default release=cvs tag=RELENG_X_Y
      ```

durch diese ersetzen:

```
*default release=cvs tag=.
```

Lesen Sie den Abschnitt über CVS Tags im Handbuch, um eine genaue Beschreibung von verwendbaren Tags zu erhalten.

b.

CTM kommt in Frage, wenn Sie über eine schlechte Internet-Anbindung (hoher Preis oder nur E-Mail Zugriff) verfügen. Der Umgang mit **CTM** ist allerdings recht mühsam und Sie können beschädigte Dateien erhalten. Daher wird es selten benutzt, was wiederum dazu führt, dass es über längere Zeit nicht funktioniert. Wir empfehlen jedem mit einem 9600 bps oder schnellerem Modem, **CVSup** zu benutzen.

3. Wenn Sie die Quellen einsetzen und nicht nur darin lesen wollen, besorgen Sie sich bitte die *kompletten* Quellen von FreeBSD-CURRENT und nicht nur ausgesuchte Teile. Der Grund hierfür ist, dass die verschiedenen Teile der Quellen voneinander abhängen. Es ist ziemlich sicher, dass Sie in Schwierigkeiten geraten, wenn Sie versuchen, nur einen Teil der Quellen zu übersetzen.

   Sehen Sie sich das `Makefile` in `/usr/src` genau an, bevor Sie FreeBSD-CURRENT übersetzen. Wenn Sie FreeBSD das erste Mal aktualisieren, sollten Sie sowohl einen Kernel als auch das System neu installieren. Lesen Sie bitte die Mailingliste FreeBSD-CURRENT (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) und `/usr/src/UPDATING`, um über Änderungen im Installationsverfahren, die manchmal vor der Einführung eines neuen Releases notwendig sind, informiert zu sein.

4. Seien Sie aktiv! Wenn Sie FreeBSD-CURRENT laufen lassen, wollen wir wissen, was Sie darüber denken, besonders wenn Sie Verbesserungsvorschläge oder Fehlerbehebungen haben. Verbesserungsvorschläge, die Code enthalten, werden übrigens begeistert entgegengenommen.

25.5.2. FreeBSD-STABLE

25.5.2.1. Was ist FreeBSD-STABLE?

FreeBSD-STABLE ist der Entwicklungszweig, auf dem Releases erstellt werden. Dieser Zweig ändert sich langsamer als FreeBSD-CURRENT und alle Änderungen hier sollten zuvor in FreeBSD-CURRENT ausgetestet sein. Beachten Sie, dass dies *immer noch* ein Entwicklungszweig ist und daher zu jedem Zeitpunkt die Quellen von FreeBSD-STABLE verwendbar sein können oder nicht. FreeBSD-STABLE ist Teil des Entwicklungsprozesses und nicht für Endanwender gedacht.

25.5.2.2. Wer braucht FreeBSD-STABLE?

Wenn Sie den FreeBSD-Entwicklungsprozess, besonders im Hinblick auf das nächste Release, verfolgen oder dazu beitragen wollen, sollten Sie erwägen, FreeBSD-STABLE zu benutzen.

Auch wenn sicherheitsrelevante Fehlerbehebungen in den FreeBSD-STABLE Zweig einfließen, müssen Sie deswegen noch lange nicht FreeBSD-STABLE verfolgen. Jeder der FreeBSD Sicherheitshinweise beschreibt für

jedes betroffene Release, [1] wie sie einen sicherheitsrelevanten Fehler beheben. Wenn Sie den Entwicklungszweig aus Sicherheitsgründen verfolgen wollen, bedenken Sie, dass Sie neben Fehlerbehebungen auch eine Vielzahl unerwünschter Änderungen erhalten werden.

Obwohl wir versuchen sicherzustellen, dass der FreeBSD-STABLE Zweig sich jederzeit übersetzen lässt und läuft, können wir dafür keine Garantie übernehmen. Auch wenn Neuentwicklungen in FreeBSD-CURRENT stattfinden, ist es jedoch so, dass mehr Leute FreeBSD-STABLE benutzen als FreeBSD-CURRENT und es daher unvermeidlich ist, dass Fehler und Grenzfälle erst in FreeBSD-STABLE auffallen.

Aus diesen Gründen empfehlen wir Ihnen *nicht*, blindlings FreeBSD-STABLE zu benutzen. Es ist wichtig, dass Sie FreeBSD-STABLE zuerst sorgfältig in einer Testumgebung austesten, bevor Sie Ihre Produktion auf FreeBSD-STABLE migrieren.

Wenn Sie dies nicht leisten können, empfehlen wir Ihnen, das aktuelle FreeBSD-Release zu verwenden. Benutzen Sie dann den binären Update-Mechanismus, um auf neue Releases zu migrieren.

25.5.2.3. Benutzen von FreeBSD-STABLE

1. Lesen Sie Mailingliste freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable), damit Sie über Abhängigkeiten beim Bau von FreeBSD-STABLE und Sachen, die besondere Aufmerksamkeit erfordern, informiert sind. Umstrittene Fehlerbehebungen oder Änderungen werden von den Entwicklern auf dieser Liste bekannt gegeben. Dies erlaubt es den Benutzern, Einwände gegen die vorgeschlagenen Änderungen vorzubringen.

   Abonnieren Sie die passende **SVN**-Liste für den jeweiligen Branch, den Sie verfolgen. Wenn Sie beispielsweise den Zweig 7-STABLE verfolgen, lesen Sie die svn-src-stable-7 (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-7). Dort sehen Sie zu jeder Änderung das Commit-Log, das Informationen zu möglichen Seiteneffekten enthält.

   Um diese Listen oder andere Listen zu abonnieren besuchen Sie bitte die Seite http://lists.FreeBSD.org/mailman/listinfo. Weitere Informationen erhalten Sie, wenn Sie dort auf die gewünschte Liste klicken. Wenn Sie daran interessiert sind, Änderungen am gesamten Quellbaum zu verfolgen, dann empfehlen wir, dass Sie svn-src-all (http://lists.FreeBSD.org/mailman/listinfo/svn-src-all) abonnieren.

2. Wenn Sie ein neues System installieren und dazu einen der monatlich aus FreeBSD-STABLE erzeugten Snapshots verwenden wollen, sollten Sie zuerst die Snapshot Website (http://www.FreeBSD.org/../snapshots/) auf aktuelle Informationen überprüfen. Alternativ können Sie auch das neueste FreeBSD-STABLE-Release von den Spiegeln beziehen und Ihr System nach den folgenden Anweisungen aktualisieren.

   Wenn Sie schon ein älteres Release von FreeBSD und das System mit dem Quellcode aktualisieren wollen, benutzen Sie einen der FreeBSD-Spiegel. Sie haben dazu zwei Möglichkeiten:

   a.

      Benutzen Sie das Programm cvsup mit der Datei `stable-supfile` aus dem Verzeichnis `/usr/share/examples/cvsup`. Dies ist die empfohlene Methode, da Sie die ganzen Quellen nur einmal herunterladen und danach nur noch Änderungen beziehen. Viele lassen `cvsup` aus `cron` heraus laufen, um ihre Quellen automatisch auf Stand zu bringen. Sie müssen das oben erwähnte `supfile` anpassen und cvsup konfigurieren.

   b.

Benutzen Sie **CTM**. Wenn Sie über keine schnelle und billige Internet-Anbindung verfügen, sollten Sie diese Methode in Betracht ziehen.

3. Benutzen Sie `cvsup` oder `ftp`, wenn Sie schnellen Zugriff auf die Quellen brauchen und die Bandbreite keine Rolle spielt, andernfalls benutzen Sie **CTM**.
4. Bevor Sie FreeBSD-STABLE übersetzen, sollten Sie sich das `Makefile` in `/usr/src` genau anschauen. Wenn Sie FreeBSD das erste Mal aktualisieren, sollten Sie sowohl einen Kernel als auch das System neu installieren. Lesen Sie bitte die Mailingliste FreeBSD-STABLE (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable) und `/usr/src/UPDATING`, um über Änderungen im Installationsverfahren, die manchmal vor der Einführung eines neuen Releases notwendig sind, informiert zu sein.

25.6. Synchronisation der Quellen

Sie können eine Internet-Verbindung (oder E-Mail) dazu nutzen, Teile von FreeBSD, wie die Quellen zu einzelnen Projekten, oder das Gesamtsystem, aktuell zu halten. Dazu bieten wir die Dienste AnonymousCVS, CVSup und CTM an.

> **Warnung:** Obwohl es möglich ist, nur Teile des Quellbaums zu aktualisieren, ist die einzige unterstütze Migrationsprozedur, den kompletten Quellbaum zu aktualisieren und alles, das heißt das Userland (z.B. alle Programme in `/bin` und `/sbin`) und die Kernelquellen, neu zu übersetzen. Wenn Sie nur einen Teil der Quellen, zum Beispiel nur den Kernel oder nur die Programme aus dem Userland, aktualisieren, werden Sie oft Probleme haben, die von Übersetzungsfehlern über Kernel-Panics bis hin zu Beschädigungen Ihrer Daten reichen können.

Anonymous CVS und **CVSup** benutzen die *Pull*-Methode [2], um die Quellen zu aktualisieren. Im Fall von **CVSup** ruft der Benutzer oder ein `cron`-Skript `cvsup` auf, das wiederum mit einem `cvsupd` Server interagiert, um Ihre Quellen zu aktualisieren. Mit beiden Methoden erhalten Sie aktuelle Updates zu einem genau von Ihnen bestimmten Zeitpunkt. Sie können die Prozedur auf bestimmte Dateien oder Verzeichnisse einschränken, so dass Sie nur die Updates bekommen, die für Sie von Interesse sind. Die Updates werden zur Laufzeit, abhängig von den Sachen, die Sie schon haben und den Sachen, die Sie haben wollen, auf dem Server generiert. **Anonymous CVS** ist eine Erweiterung von **CVS**, die es Ihnen erlaubt, Änderungen direkt aus einem entfernten CVS-Repository zu ziehen. **Anonymous CVS** ist leichter zu handhaben als **CVSup**, doch ist letzteres sehr viel effizienter.

Im Gegensatz dazu vergleicht **CTM** Ihre Quellen nicht mit denen auf einem Server. Stattdessen läuft auf dem Server ein Skript, das Änderungen an Dateien gegenüber seinem vorigen Lauf bemerkt, die Änderungen komprimiert, mit einer Sequenznummer versieht und für das Verschicken per E-Mail kodiert (es werden nur druckbare ASCII-Zeichen verwendet). Wenn Sie diese “CTM-Deltas” erhalten haben, können Sie sie mit ctm_rmail(1) benutzen, welches die Deltas dekodiert, verifiziert und dann die Änderungen an Ihren Quellen vornimmt. Dieses Verfahren ist viel effizienter als **CVSup** und erzeugt auch weniger Last auf unseren Servern, da es die *Push*-Methode [3] verwendet.

Es gibt natürlich noch weitere Unterschiede, die Sie beachten sollten. Wenn Sie unabsichtlich Teile Ihres Archivs löschen, wird das von **CVSup** wie **Anonymous CVS** erkannt und repariert. Wenn sich fehlerhafte Dateien in Ihrem Quellbaum befinden, löschen Sie diese einfach und synchronisieren erneut. **CTM** leistet das nicht, wenn Sie Teile

des Quellbaums gelöscht haben und keine Sicherung besitzen, müssen Sie von neuem, das heißt vom letzten "Basis-Delta", starten und die Änderungen wieder mit **CTM** nachziehen.

25.7. Das komplette Basissystem neu bauen

Wenn Sie Ihren lokalen Quellbaum mit einer bestimmten FreeBSD Version (FreeBSD-STABLE, FreeBSD-CURRENT, usw.) synchronisiert haben, können Sie diesen benutzen, um das System neu zu bauen.

Erstellen Sie eine Sicherungskopie!: Es kann nicht oft genug betont werden, wie wichtig es ist, Ihr System zu sichern, *bevor* Sie die nachfolgenden Schritte ausführen. Obwohl der Neubau des Systems eine einfache Aufgabe ist, wenn Sie sich an die folgende Anleitung halten, kann es dennoch vorkommen, dass Sie einen Fehler machen, oder dass Ihr System nicht mehr bootet, weil andere Entwickler Fehler in den Quellbaum eingeführt haben.

Stellen Sie sicher, dass Sie eine Sicherung erstellt haben und über eine Fixit-Floppy oder eine startfähige CD verfügen. Wahrscheinlich werden Sie die Startmedien nicht benötigen, aber gehen Sie auf Nummer sicher!

Abonnieren Sie die richtige Mailingliste: Die FreeBSD-STABLE und FreeBSD-CURRENT Zweige befinden sich in *ständiger Entwicklung*. Die Leute, die zu FreeBSD beitragen, sind Menschen und ab und zu machen sie Fehler.

Manchmal sind diese Fehler harmlos und lassen Ihr System eine Warnung ausgeben. Die Fehler können allerdings auch katastrophal sein und dazu führen, dass Sie Ihr System nicht mehr booten können, Dateisysteme beschädigt werden oder Schlimmeres passiert.

Wenn solche Probleme auftauchen, wird ein "heads up" an die passende Mailingliste geschickt, welches das Problem erklärt und die betroffenen Systeme benennt. Eine "all clear" Meldung wird versendet, wenn das Problem gelöst ist.

Wenn Sie FreeBSD-STABLE oder FreeBSD-CURRENT benutzen und nicht die Mailinglisten FreeBSD-STABLE (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable) beziehungsweise FreeBSD-CURRENT (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current) lesen, bringen Sie sich nur unnötig in Schwierigkeiten.

Finger weg von `make world`: Ältere Dokumentationen empfehlen, das Kommando `make world` für den Neubau. Das Kommando überspringt wichtige Schritte. Setzen Sie es nur ein, wenn Sie wissen was Sie tun. In fast allen Fällen ist `make world` falsch, benutzen Sie stattdessen die nachstehende Anleitung.

25.7.1. Richtig aktualisieren

Um Ihr System zu aktualisieren, sollten Sie zuerst `/usr/src/UPDATING` lesen, und eventuelle, für Ihre Quellcodeversion nötigen Aufgaben erledigen, bevor Sie das System bauen. Danach aktualisieren Sie Ihr System mit den folgenden Schritten.

Bei den hier dargestellten Aktualisierungsschritten wird davon ausgegangen, dass Sie momentan eine alte FreeBSD-Version verwenden, die aus einem alten Compiler, Kernel, sowie einem alten Basissystem und veralteten Konfigurationsdateien besteht. Mit "Basissystem" sind hier die zentralen Binärdateien, Bibliotheken und Entwicklerdateien gemeint. Der Compiler ist Teil des "Basissystems", beinhaltet aber ein paar Besonderheiten.

Es wird ausserdem davon ausgegangen, dass Sie bereits die Quellen für ein neues System bezogen haben. Falls die Quellen in dem vorliegenden System zu alt sind, lesen Sie Abschnitt 25.6, um detaillierte Hilfe über die Aktualisierung der Quellen zu erhalten.

Die Aktualisierung des Systems aus den Quellen ist ein wenig ausgetüftelter als es zunächst den Anschein hat. Die Entwickler von FreeBSD haben es über die Jahre für Nötig befunden, den vorgeschlagenen Ablauf ziemlich stark zu verändern, da neue Arten von unvermeidlichen Abhängigkeiten mit der Zeit ans Licht kamen. Der übrige Teil dieses Abschnitts beschreibt die Überlegungen hinter der aktuell empfohlenen Aktualisierungsreihenfolge.

Jede erfolgreiche Aktualisierung muss sich mit den folgenden Sachverhalten auseinandersetzen:

- Der alte Compiler ist möglicherweise nicht in der Lage, den neuen Kernel zu übersetzen (alte Compiler besitzen manchmal Fehler). Deshalb sollte der neue Kernel mit dem neuen Compiler übersetzt werden. Ganz besonders muss darauf geachtet werden, dass der neue Compiler vor dem neuen Kernel gebaut wird. Das bedeutet nicht unbedingt, dass der neue Compiler auch *installiert* werden muss, bevor der neue Kernel gebaut wird.
- Das neue Basissystem benötigt eventuell neue Eigenschaften des Kernels. Also muss der neue Kernel installiert sein, bevor das neue Basissystem installiert wird.

Diese ersten beiden Sachverhalte sind die Grundlage für die zentrale Sequenz von `buildworld`, `buildkernel`, `installkernel` und `installworld`, die in den folgenden Abschnitten beschrieben wird. Dies ist keine vollständige Liste all der Gründe, warum Sie den aktuell empfohlenen Prozess der Aktualisierung bevorzugen sollten. Ein paar der weniger naheliegenden Gründe sind im folgenden aufgezählt:

- Das alte Basissystem wird möglicherweise nicht korrekt mit dem neuen Kernel funktionieren, weshalb Sie das neue Basissystem sofort nach der Installation des neuen Kernels installieren müssen.
- Manche Änderungen an der Konfiguration müssen erledigt worden sein, bevor das neue Basissystem installiert wird, jedoch können andere die Funktionalität des alten Basissystems beeinträchtigen. Aus diesem Grund sind zwei verschiedene Schritte notwendig, um eine Aktualisierung der Konfiguration durchzuführen.
- Der Aktualisierungsprozess ersetzt zum Grossteil Dateien oder fügt neue hinzu, bestehende Dateien werden nicht gelöscht. In wenigen Ausnahmefällen kann dies Probleme verursachen. Aus diesem Grund wird der Aktualisierungsprozess manchmal bestimmte Dateien zum manuellen Löschen vorschlagen. Dies wird eventuell in der Zukunft automatisch durchgeführt.

Diese Bedenken haben zu der folgenden Reihenfolge geführt. Beachten Sie, dass der genaue Ablauf für bestimmte Aktualisierungen zusätzliche Schritte nach sich zieht, jedoch sollte der Kernprozess davon nicht beeinträchtigt werden:

1. `make buildworld`

   Dieser Schritt übersetzt zuerst den neuen Compiler und ein paar damit zusammenhängende Werkzeuge und verwendet dann den neuen Compiler, um den Rest des Basissystems zu erstellen. Das Ergebnis landet dann in `/usr/obj`.

2. `make buildkernel`

   Statt dem alten Ansatz, config(8) und make(1) zu verwenden, nutzt dieser den *neuen* Compiler, der in `/usr/obj` abgelegt ist. Das schützt Sie vor falschen Compiler-Kernel-Kombinationen.

3. `make installkernel`

   Platziert den neuen Kernel und Kernelmodule auf der Platte, was es erlaubt, mit dem frisch aktualisierten Kernel zu starten.

4. Starten Sie das System neu in den Single-User-Modus.

   Der Single-User-Modus minimiert Probleme mit der Aktualisierung von Programmen, die bereits gestartet sind. Ebenso minimiert es Probleme, die mit der Verwendung des alten Basissystems und des neuen Kernels zu tun haben könnten.

5. `mergemaster -p`

   Dieser Schritt aktualisiert ein paar initiale Konfigurationsdateien als Vorbereitung für das neue Basissystem. Beispielsweise fügt es neue Benutzergruppen zum System oder neue Benutzernamen in die Passwortdatenbank hinzu. Dies wird oftmals benötigt, wenn neue Gruppen oder bestimmte Systembenutzerkonten seit der letzten Aktualisierung hinzu gekommen sind, so dass der `installworld`-Schritt in der Lage ist, auf dem neu installierten System die Benutzer oder Systemgruppennamen ohne Probleme zu verwenden.

6. `make installworld`

   Kopiert das Basissystem aus `/usr/obj`. Sie haben jetzt den neuen Kernel und das neue Basissystem auf der Festplatte.

7. `mergemaster`

   Sie können nun die verbleibenden Konfigurationsdateien aktualisieren, da Sie nun das neue Basissystem auf der Platte haben.

8. Starten Sie das System neu.

   Ein kompletter Systemneustart ist notwendig, um den neuen Kernel und das neue Basissystem mit den neuen Konfigurationsdateien zu laden.

Beachten Sie, dass wenn Sie von einem Release des gleichen FreeBSD-Zweigs auf ein aktuelleres Release des gleichen Zweigs, z.B. von 7.0 auf 7.1, aktualisieren, dann ist diese Vorgehensweise nicht unbedingt notwendig, da Sie nur sehr unwahrscheinlich in ungünstige Kombinationen zwischen Compiler, Kernel, Basissystem und den Konfigurationsdateien geraten werden. Die ältere Vorgehensweise von `make world`, gefolgt von der Erstellung und Installation des neuen Kernels funktioniert möglicherweise gut genug, um kleinere Aktualisierungen vorzunehmen.

Wenn Sie allerdings zwischen Hauptversionen aktualisieren wollen und befolgen diese Schritte nicht, sollten Sie sich auf Probleme gefasst machen.

Es ist auch wichtig zu wissen, dass viele Aktualisierungen, z.B. von 4.*X* auf 5.0, viele spezielle und zusätzliche Schritte benötigt, wie beispielsweise das umbennen oder löschen von speziellen Dateien vor installworld. Lesen Sie die Datei `/usr/src/UPDATING` gründlich, besonders am Ende, wo die aktuell vorgeschlagene Aktualisierungssequenz explizit aufgelistet ist.

Diese Prozedur hat sich mit der Zeit weiterentwickelt, da die Entwickler es für unmöglich erachtet haben, bestimmte Arten von Kombinationsproblemen vollständig auszuschliessen. Hoffentlich wird die aktuelle Aktualisierungsprozedur für lange Zeit stabil bleiben.

Als Zusammenfassung ist hier nochmal die aktuell vorgeschlagene Vorgehensweise für die Aktualisierung von FreeBSD aus den Quellen aufgelistet:

```
# cd /usr/src
# make buildworld
# make buildkernel
# make installkernel
# shutdown -r now
```

> **Anmerkung:** Es gibt einige, sehr seltene Situationen, in denen Sie `mergemaster -p` zusätzlich ausführen müssen, bevor Sie das System mit `buildworld` bauen. Diese Situationen werden in `UPDATING` beschrieben. Solche Situationen treten aber in der Regel nur dann auf, wenn Sie Ihr FreeBSD-System um eine oder mehrere Hauptversionen aktualisieren.

Nachdem `installkernel` erfolgreich abgeschlossen wurde, starten Sie das System im Single-User-Modus (etwa durch die Eingabe von `boot -s` am Loaderprompt). Danach führen Sie die folgenden Anweisungen aus:

```
# mount -u /
# mount -a -t ufs
# adjkerntz -i
# mergemaster -p
# cd /usr/src
# make installworld
# mergemaster
# reboot
```

> **Lesen Sie bitte weiter:** Die obige Vorschrift ist nur eine Gedächtnisstütze. Um die einzelnen Schritte zu verstehen, lesen Sie bitte die folgenden Abschnitte, insbesondere wenn Sie einen angepassten Kernel erstellen.

25.7.2. Lesen Sie `/usr/src/UPDATING`

Bevor Sie etwas anderes tun, lesen Sie bitte `/usr/src/UPDATING` (oder die entsprechende Datei, wenn Sie den Quellcode woanders installiert haben). Die Datei enthält wichtige Informationen zu Problemen, auf die Sie stoßen könnten oder gibt die Reihenfolge vor, in der Sie bestimmte Kommandos laufen lassen müssen. Die Anweisungen in `UPDATING` sind aktueller als die in diesem Handbuch. Im Zweifelsfall folgen Sie bitte den Anweisungen aus `UPDATING`.

> **Wichtig:** Das Lesen von `UPDATING` ersetzt nicht das Abonnieren der richtigen Mailingliste. Die beiden Voraussetzungen ergänzen sich, es reicht nicht aus, nur eine zu erfüllen.

25.7.3. Überprüfen Sie `/etc/make.conf`

Überprüfen Sie die Dateien `/usr/share/examples/etc/make.conf` und `/etc/make.conf`. Die erste enthält Vorgabewerte, von denen die meisten auskommentiert sind. Um diese während des Neubaus des Systems zu nutzen, tragen Sie die Werte in `/etc/make.conf` ein. Beachten Sie, dass alles, was Sie in `/etc/make.conf` eintragen, bei jedem Aufruf von `make` angezogen wird. Es ist also klug, hier etwas Sinnvolles einzutragen.

Typischerweise wollen Sie die Zeilen, die `CFLAGS` und `NO_PROFILE` enthalten, aus `/usr/share/examples/etc/make.conf` nach `/etc/make.conf` übertragen und dort aktivieren.

Sehen Sie sich auch die anderen Definitionen, wie `COPTFLAGS` oder `NOPORTDOCS` an und entscheiden Sie, ob Sie diese aktivieren wollen.

25.7.4. Aktualisieren Sie die Dateien in `/etc`

Das Verzeichnis `/etc` enthält den Großteil der Konfigurationsdateien des Systems und Skripten, die beim Start des Systems ausgeführt werden. Einige dieser Skripten ändern sich bei einer Migration auf eine neue FreeBSD-Version.

Einige der Konfigurationsdateien, besonders `/etc/group`, werden für den Normalbetrieb des Systems gebraucht.

Es gab Fälle, in denen das Kommando `make installworld` auf bestimmte Accounts oder Gruppen angewiesen war, die aber während der Aktualisierung fehlten. Demzufolge kam es zu Problemen bei der Aktualisierung. In einigen Fällen prüft `make buildworld` ob die Accounts oder Gruppen vorhanden sind.

Ein Beispiel dafür trat beim Anlegen des Benutzers `smmsp` auf. Die Installationsprozedur schlug an der Stelle fehl, an der mtree(8) versuchte, `/var/spool/clientmqueue` anzulegen.

Um dieses Problem zu umgehen,rufen Sie mergemaster(8) prä-buildworld-Modus auf, der mit `-p` aktiviert wird. In diesem Modus werden nur Dateien verglichen, die für den Erfolg von `buildworld` oder `installworld` essentiell sind.

> **Tipp:** Wenn Sie besonders paranoid sind, sollten Sie Ihr System nach Dateien absuchen, die der Gruppe, die Sie umbenennen oder löschen, gehören:
>
> ```
> # find / -group GID -print
> ```
>
> Das obige Kommando zeigt alle Dateien an, die der Gruppe *`GID`* (dies kann entweder ein Gruppenname oder eine numerische ID sein) gehören.

25.7.5. Wechseln Sie in den Single-User-Modus

Sie können das System im Single-User-Modus übersetzen. Abgesehen davon, dass dies etwas schneller ist, werden bei der Installation des Systems viele wichtige Dateien, wie die Standard-Systemprogramme, die Bibliotheken und Include-Dateien, verändert. Sie bringen sich in Schwierigkeiten, wenn Sie diese Dateien auf einem laufenden System verändern, besonders dann, wenn zu dieser Zeit Benutzer auf dem System aktiv sind.

Eine andere Methode übersetzt das System im Mehrbenutzermodus und wechselt für die Installation in den Single-User-Modus. Wenn Sie diese Methode benutzen wollen, warten Sie mit den folgenden Schritten, bis der Bau des Systems fertig ist und Sie mit `installkernel` oder `installworld` installieren wollen.

Als Superuser können Sie mit dem folgenden Kommando ein laufendes System in den Single-User-Modus bringen:

```
# shutdown now
```

Alternativ können Sie das System mit der Option “single user” in den Single-User-Modus booten. Danach geben Sie die folgenden Befehle ein:

```
# fsck -p
# mount -u /
# mount -a -t ufs
# swapon -a
```

Die Kommandos überprüfen die Dateisysteme, hängen `/` wieder beschreibbar ein, hängen dann alle anderen UFS Dateisysteme aus `/etc/fstab` ein und aktivieren den Swap-Bereich.

Anmerkung: Zeigt Ihre CMOS-Uhr die lokale Zeit und nicht GMT an, dies erkennen Sie daran, dass date(1) die falsche Zeit und eine falsche Zeitzone anzeigt, setzen Sie das folgende Kommando ab:

```
# adjkerntz -i
```

Dies stellt sicher, dass Ihre Zeitzone richtig eingestellt ist. Ohne dieses Kommando werden Sie vielleicht später Probleme bekommen.

25.7.6. Entfernen Sie `/usr/obj`

Die neu gebauten Teile des Systems werden in der Voreinstellung unter `/usr/obj` gespeichert. Die Verzeichnisse dort spiegeln die Struktur unter `/usr/src`.

Sie können den `make buildworld` Prozess beschleunigen, indem Sie dieses Verzeichnis entfernen. Dies erspart Ihnen zudem einigen Ärger aufgrund von Abhängigkeiten.

Einige Dateien unter `/usr/obj` sind vielleicht durch die `immutable`-Option (siehe chflags(1)) schreibgeschützt, die vor dem Löschen entfernt werden muss.

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

25.7.7. Übersetzen der Quellen des Basissystems

25.7.7.1. Sichern der Ausgaben

Für den Fall, dass etwas schief geht, sollten Sie die Ausgaben von make(1) in einer Datei sichern, damit Sie eine Kopie der Fehlermeldung besitzen. Das mag Ihnen nicht helfen, den Fehler zu finden, kann aber anderen helfen, wenn Sie Ihr Problem in einer der FreeBSD-Mailinglisten schildern.

Dazu können Sie einfach das Kommando script(1) benutzen, dem Sie beim Aufruf als Parameter den Dateinamen für die Ausgaben mitgeben. Setzen Sie das Kommando unmittelbar vor dem Neubau ab und geben Sie **`exit`** ein, wenn der Bau abgeschlossen ist:

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
# make TARGET
... Ausgaben des Kommandos ...
# exit
Script done, ...
```

Sichern Sie die Ausgaben nicht in `/tmp`, da dieses Verzeichnis beim nächsten Boot aufgeräumt werden kann. Ein geeigneteres Verzeichnis ist `/var/tmp`, wie im vorigen Beispiel gezeigt, oder das Heimatverzeichnis von `root`.

25.7.7.2. Übersetzen des Basissystems

Wechseln Sie in das Verzeichnis, in dem die Quellen liegen (in der Voreinstellung ist das `/usr/src`):

```
# cd /usr/src
```

Zum Neubau der Welt benutzen Sie make(1). Dieses Kommando liest ein `Makefile`, das Anweisungen enthält, wie die Programme, aus denen FreeBSD besteht, zu bauen sind und in welcher Reihenfolge diese zu bauen sind.

Ein typischer Aufruf von `make` sieht wie folgt aus:

```
# make -x -DVARIABLE target
```

In diesem Beispiel ist *`-x`* eine Option, die Sie an make(1) weitergeben wollen. Eine Liste gültiger Optionen finden Sie in der make(1) Manualpage.

Das Verhalten eines `Makefile`s wird von Variablen bestimmt. Mit `-D`*`VARIABLE`* setzen Sie eine Variable. Diese Variablen sind dieselben, die auch in `/etc/make.conf` gesetzt werden, dies ist nur ein alternativer Weg, Variablen zu setzen.

Um zu verhindern, dass die "profiled" Bibliotheken gebaut werden, rufen Sie `make` wie folgt auf:

```
# make -DNO_PROFILE target
```

Dieser Aufruf entspricht dem folgenden Eintrag in `/etc/make.conf`:

```
NO_PROFILE=    true     #    Avoid compiling profiled libraries
```

Jedes `Makefile` definiert einige "Ziele", die festlegen, was genau zu tun ist. Mit *`target`* wählen Sie eins dieser Ziele aus.

Einige Ziele im `Makefile` sind nicht für den Endanwender gedacht, sondern unterteilen den Bauprozess in eine Reihe von Einzelschritten.

Im Regelfall müssen Sie make(1) keine Parameter mitgeben, so dass Ihre Kommandozeile wie folgt aussehen wird:

```
# make target
```

`target` steht dabei für die verschiedenen Ziele. Das erste Ziel sollte immer `buildworld` sein.

Mit `buildworld` wird ein kompletter Baum unterhalb von `/usr/obj` gebaut, der mit `installworld`, einem weiteren Ziel, auf dem System installiert werden kann.

Über separate Optionen zu verfügen, ist aus mehreren Gründen nützlich. Erstens können Sie das System auf einem laufenden System bauen, da die Bauprozedur abgekapselt vom Rest des Systems ist. Das System lässt sich im Mehrbenutzermodus ohne negative Seiteneffekte bauen. Die Installation mit `installworld` sollte aber immer noch im Single-User-Modus erfolgen.

Zweitens können Sie NFS benutzen, um mehrere Maschinen in Ihrem Netzwerk zu aktualisieren. Wenn Sie die Maschinen `A`, `B` und `C` aktualisieren wollen, lassen sie `make buildworld` und `make installworld` auf `A` laufen. Auf den Maschinen `B` und `C` können Sie die Verzeichnisse `/usr/src` und `/usr/obj` von `A` einhängen und brauchen dort nur noch `make installworld` auszuführen, um die Bauresultate zu installieren.

Obwohl das Ziel `world` noch existiert, sollten Sie es wirklich nicht mehr benutzen.

Um das System zu bauen, setzen Sie das folgende Kommando ab:

```
# make buildworld
```

Mit `-j` können Sie `make` anweisen, mehrere Prozesse zu starten. Besonders effektiv ist das auf Mehrprozessor-Systemen. Da aber der Übersetzungsprozess hauptsächlich von IO statt der CPU bestimmt wird, ist diese Option auch auf Einprozessor-Systemen nützlich.

Auf einem typischen Einprozessor-System können Sie den folgenden Befehl absetzen:

```
# make -j4 buildworld
```

make(1) wird dann bis zu vier Prozesse gleichzeitig laufen lassen. Erfahrungsberichte aus den Mailinglisten zeigen, dass dieser Aufruf typischerweise den besten Geschwindigkeitsgewinn bringt.

Wenn Sie ein Mehrprozessor-System besitzen und SMP in Ihrem Kernel konfiguriert ist, probieren Sie Werte zwischen 6 und 10 aus.

25.7.7.3. Laufzeiten

Die Laufzeit eines Baus wird von vielen Faktoren beeinflusst, ein aktuelles System benötigt aber etwa zwei Stunden um FreeBSD-STABLE zu bauen. Der Bau von FreeBSD-CURRENT dauert etwas länger.

25.7.8. Übersetzen und Installation des Kernels

Um das Beste aus Ihrem System zu holen, sollten Sie einen neuen Kernel kompilieren. Praktisch gesehen ist das sogar notwendig, da sich einige Datenstrukturen geändert haben und Programme wie ps(1) oder top(1) nur mit einem Kernel zusammen arbeiten, der auch zu dem entsprechenden Quellcode passt.

Am einfachsten und sichersten bauen Sie dazu den `GENERIC` Kernel. Obwohl der `GENERIC` Kernel vielleicht nicht alle Ihre Geräte unterstützt, sollte er alles enthalten, um das System in den Single-User-Modus zu booten. Dies ist auch ein guter Test, um zu sehen, dass das System ordnungsgemäß funktioniert. Nachdem Sie mit `GENERIC` gebootet und sichergestellt haben, dass Ihr System funktioniert, können Sie einen neuen Kernel mit Ihrer Konfigurationsdatei bauen.

In aktuellen FreeBSD-Versionen müssen Sie das Basissystem neu bauen, bevor Sie einen neuen Kernel erstellen.

> **Anmerkung:** Wenn Sie einen angepassten Kernel erstellen wollen und bereits über eine Konfigurationsdatei verfügen, geben Sie diese, wie im folgenden Beispiel gezeigt, auf der Kommandozeile an:
>
> ```
> # cd /usr/src
> # make buildkernel KERNCONF=MYKERNEL
> # make installkernel KERNCONF=MYKERNEL
> ```

Wenn `kern.securelevel` einen Wert größer als `1` besitzt *und* der Kernel mit `noschg` oder ähnlichen Optionen geschützt ist, müssen Sie `installkernel` im Einbenutzermodus ausführen. Wenn das nicht der Fall ist, sollten die beiden Kommandos problemlos im Mehrbenutzermodus laufen. Weitere Informationen über `kern.securelevel` finden Sie in init(8) und chflags(1) erläutert Optionen, die Sie auf Dateien setzen können.

25.7.9. Booten Sie in den Single-User-Modus

Um zu prüfen, ob der neue Kernel funktioniert, sollten Sie in den Single-User-Modus booten. Folgen Sie dazu der Anleitung aus Abschnitt 25.7.5.

25.7.10. Installation des Systems

Nun können Sie das neue System mit `installworld` installieren. Rufen Sie dazu das folgende Kommando auf:

```
# cd /usr/src
# make installworld
```

> **Anmerkung:** Wenn Sie mit dem `make buildworld` Kommando Variablen verwendet haben, müssen Sie dieselben Variablen auch bei dem `make installworld` Kommando angeben. Auf die anderen Optionen trifft das nur bedingt zu: `-j` darf mit `installworld` nicht benutzt werden.
>
> Sie haben zum Bauen die folgende Kommandozeile verwendet:
>
> ```
> # make -DNO_PROFILE buildworld
> ```
>
> Bei der Installation setzen Sie dann das folgende Kommando ab:
>
> ```
> # make -DNO_PROFILE installworld
> ```
>
> Würden Sie die Variable bei der Installation weglassen, so würde das System versuchen, die “profiled” Bibliotheken, die aber gar nicht gebaut wurden, zu installieren.

25.7.11. Aktualisieren der von `make installworld` ausgelassenen Dateien

Neue oder geänderte Konfigurationsdateien aus einigen Verzeichnissen, besonders `/etc`, `/var` und `/usr` werden bei der Installationsprozedur nicht berücksichtigt.

Sie können diese Dateien mit mergemaster(8) aktualisieren. Alternativ können Sie das auch manuell durchführen, obwohl wir diesen Weg nicht empfehlen. Egal welchen Weg Sie beschreiten, sichern Sie vorher den Inhalt von `/etc` für den Fall, dass etwas schief geht.

25.7.11.1. `mergemaster`

Beigetragen von Tom Rhodes.

Das Bourne-Shell Skript mergemaster(8) hilft Ihnen dabei, die Unterschiede zwischen den Konfigurationsdateien in `/etc` und denen im Quellbaum unter `/usr/src/etc` zu finden. `mergemaster` ist der empfohlene Weg, Ihre Systemkonfiguration mit dem Quellbaum abzugleichen.

Rufen Sie `mergemaster` einfach auf und schauen Sie zu. Ausgehend von `/` wird `mergemaster` einen virtuellen Root-Baum aufbauen und darin die neuen Konfigurationsdateien ablegen. Diese Dateien werden dann mit den auf Ihrem System installierten verglichen. Unterschiede zwischen den Dateien werden im diff(1)-Format dargestellt. Neue oder geänderte Zeilen werden mit `+` gekennzeichnet. Zeilen die gelöscht oder ersetzt werden, sind mit einem `-` gekennzeichnet. Das Anzeigeformat wird in diff(1) genauer erklärt.

mergemaster(8) zeigt Ihnen jede geänderte Datei an und Sie haben die Wahl, die neue Datei (in `mergemaster` wird sie temporäre Datei genannt) zu löschen, sie unverändert zu installieren, den Inhalt der neuen Datei mit dem Inhalt der alten Datei abzugleichen, oder die diff(1) Ausgabe noch einmal zu sehen. Sie können die aktuelle Datei auch überspringen, sie wird dann noch einmal angezeigt, nachdem alle anderen Dateien abgearbeitet wurden. Sie erhalten Hilfe, wenn Sie bei der Eingabeaufforderung von `mergemaster` ein **?** eingeben.

Wenn Sie die temporäre Datei löschen, geht `mergemaster` davon aus, dass Sie Ihre aktuelle Datei behalten möchten. Wählen Sie die Option bitte nur dann, wenn Sie keinen Grund sehen, die aktuelle Datei zu ändern.

Wenn Sie die temporäre Datei installieren, wird Ihre aktuelle Datei mit der neuen Datei überschrieben. Sie sollten alle unveränderten Konfigurationsdateien auf diese Weise aktualisieren.

Wenn Sie sich entschließen den Inhalt beider Dateien abzugleichen, wird ein Texteditor aufgerufen, indem Sie beide Dateien nebeneinander betrachten können. Mit der Taste **l** übernehmen Sie die aktuelle Zeile der links dargestellten Datei, mit der Taste **r** übernehmen Sie die Zeile der rechts dargestellten Datei. Das Ergebnis ist eine Datei, die aus Teilen der beiden ursprünglichen Dateien besteht und installiert werden kann. Dieses Verfahren wird gewöhnlich bei veränderten Dateien genutzt.

Haben Sie sich entschieden die Differenzen noch einmal anzuzeigen, zeigt Ihnen mergemaster(8) dieselbe Ausgabe, die Sie gesehen haben, bevor die Eingabeaufforderung ausgegeben wurde.

Wenn mergemaster(8) alle Systemdateien abgearbeitet hat, werden weitere Optionen abgefragt. Sie werden unter Umständen gefragt, ob Sie die Passwort-Datei neu bauen lassen wollen. Am Ende haben Sie die Möglichkeit, den Rest der temporären Dateien zu löschen.

25.7.11.2. Manueller Abgleich der Konfigurationsdateien

Wenn Sie den Abgleich lieber selbst ausführen wollen, beachten Sie bitte, dass Sie nicht einfach die Dateien aus `/usr/src/etc` nach `/etc` kopieren können. Einige dieser Dateien müssen zuerst *installiert* werden, bevor sie benutzt werden können. Das liegt daran, dass `/usr/src/etc` keine exakte Kopie von `/etc` ist. Zudem gibt es Dateien, die sich in `/etc` befinden aber nicht in `/usr/src/etc`.

Wenn Sie, wie empfohlen, `mergemaster` benutzen, können Sie direkt in den nächsten Abschnitt wechseln.

Am einfachsten ist es, wenn Sie die neuen Dateien in ein temporäres Verzeichnis installieren und sie nacheinander auf Differenzen zu den bestehenden Dateien durchsehen.

> **Sichern Sie die Inhalte von `/etc`:** Obwohl bei dieser Prozedur keine Dateien in `/etc` automatisch verändert werden, sollten Sie dessen Inhalt an einen sicheren Ort kopieren:
>
> ```
> # cp -Rp /etc /etc.old
> ```
>
> Mit `-R` wird rekursiv kopiert und `-p` erhält die Attribute der kopierten Dateien, wie Zugriffszeiten und Eigentümer.

Sie müssen die neuen Dateien in einem temporären Verzeichnis installieren. `/var/tmp/root` ist eine gute Wahl für das temporäre Verzeichnis, in dem auch noch einige Unterverzeichnisse angelegt werden müssen.

```
# mkdir /var/tmp/root
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root distrib-dirs distribution
```

Die obigen Kommandos bauen die nötige Verzeichnisstruktur auf und installieren die neuen Dateien in diese Struktur. Unterhalb von `/var/tmp/root` wurden einige leere Verzeichnisse angelegt, die Sie am besten wie folgt entfernen:

```
# cd /var/tmp/root
# find -d . -type d | xargs rmdir 2>/dev/null
```

Im obigen Beispiel wurde die Fehlerausgabe nach `/dev/null` umgeleitet, um die Warnungen über nicht leere Verzeichnisse zu unterdrücken.

`/var/tmp/root` enthält nun alle Dateien, die unterhalb von `/` installiert werden müssen. Sie müssen nun jede dieser Dateien mit den schon existierenden Dateien vergleichen.

Einige der installierten Dateien unter `/var/tmp/root` beginnen mit einem ".". Als dieses Kapitel verfasst wurde, waren das nur die Startdateien für die Shells in `/var/tmp/root/` und `/var/tmp/root/root/`. Abhängig davon, wann Sie dieses Handbuch lesen, können mehr Dateien dieser Art existieren. Verwenden Sie `ls -a` um sicherzustellen, dass Sie alle derartigen Dateien finden.

Benutzen Sie diff(1) um Unterschiede zwischen zwei Dateien festzustellen:

```
# diff /etc/shells /var/tmp/root/etc/shells
```

Das obige Kommando zeigt Ihnen die Unterschiede zwischen der installierten Version von `/etc/shells` und der neuen Version in `/var/tmp/root/etc/shells`. Entscheiden Sie anhand der Unterschiede, ob Sie beide Dateien abgleichen oder die neue Version über die alte kopieren wollen.

> **Versehen Sie das temporäre Verzeichnis mit einem Zeitstempel:** Wenn Sie das System oft neu bauen, müssen Sie `/etc` genauso oft aktualisieren. Dies kann mit der Zeit sehr lästig werden.
>
> Sie können das Verfahren beschleunigen, wenn Sie sich eine Kopie der Dateien behalten, die Sie zuletzt nach `/etc` installiert haben. Das folgende Verfahren zeigt Ihnen, wie das geht.

1. Folgen Sie der normalen Prozedur um das System zu bauen. Wenn Sie `/etc` und die anderen Verzeichnisse aktualisieren wollen, geben Sie dem temporären Verzeichnis einen Namen, der das aktuelle Datum enthält. Wenn Sie dies zum Beispiel am 14. Februar 1998 durchführten, hätten Sie die folgenden Kommandos abgesetzt:

   ```
   # mkdir /var/tmp/root-19980214
   # cd /usr/src/etc
   # make DESTDIR=/var/tmp/root-19980214 \
       distrib-dirs distribution
   ```

2. Gleichen Sie die Änderungen entsprechend der Anleitung von oben ab.

   Wenn Sie fertig sind, entfernen Sie das Verzeichnis `/var/tmp/root-19980214` *nicht*.

3. Wenn Sie nun neue Quellen heruntergeladen und gebaut haben, folgen Sie bitte Schritt 1. Wenn Sie zwischen den Updates eine Woche gewartet haben, haben Sie nun ein Verzeichnis mit dem Namen `/var/tmp/root-19980221`.

4. Sie können nun die Unterschiede, die sich in einer Woche ergeben haben, sehen, indem Sie diff(1) rekursiv anwenden:

   ```
   # cd /var/tmp
   # diff -r root-19980214 root-19980221
   ```

Üblicherweise sind die Differenzen, die Sie jetzt sehen, kleiner als die Differenzen zwischen `/var/tmp/root-19980221/etc` und `/etc`. Da die angezeigten Differenzen kleiner sind, ist es jetzt einfacher den Abgleich der Dateien durchzuführen.

5. Sie können nun das älteste der beiden `/var/tmp/root-*` Verzeichnisse entfernen:

   ```
   # rm -rf /var/tmp/root-19980214
   ```

6. Wiederholen Sie diesen Prozess jedes Mal wenn Sie Dateien in `/etc` abgleichen müssen.

Mit date(1) können Sie den Verzeichnisnamen automatisch erzeugen:

```
# mkdir /var/tmp/root-‘date "+%Y%m%d"‘
```

25.7.12. Das System neu starten

Sie sind nun am Ende der Prozedur angelangt. Nachdem Sie sich davon überzeugt haben, dass Ihr System funktioniert, starten Sie Ihr System mit shutdown(8) neu:

```
# shutdown -r now
```

25.7.13. Ende

Herzlichen Glückwunsch! Sie haben gerade erfolgreich Ihr FreeBSD System aktualisiert.

Es ist übrigens leicht einen Teil des Systems wiederherzustellen, für den Fall, dass Ihnen ein kleiner Fehler unterlaufen ist. Wenn Sie beispielsweise während des Updates oder Abgleichs `/etc/magic` aus Versehen gelöscht haben, wird file(1) nicht mehr funktionieren. In diesem Fall können Sie das Problem mit dem folgenden Kommando beheben:

```
# cd /usr/src/usr.bin/file
# make all install
```

25.7.14. Fragen

1. Muss ich wirklich immer alles neu bauen, wenn sich etwas geändert hat?

Darauf gibt es keine einfache Antwort. Was zu tun ist, hängt von den Änderungen ab. Es lohnt wahrscheinlich nicht, alles neu zu bauen, wenn sich bei einem **CVSup**-Lauf nur die folgenden Dateien geändert haben:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/share/mk/bsd.port.mk
```

In diesem Fall können Sie in die entsprechenden Unterverzeichnisse wechseln und dort `make all install` ausführen. Wenn sich allerdings etwas Wichtiges, wie `src/lib/libc/stdlib`, geändert hat, sollten Sie die Welt oder mindestens die statisch gelinkten Teile des Systems (sowie Ihre statisch gelinkten Ergänzungen) neu bauen.

Letztendlich ist das Ihre Entscheidung. Sie sind vielleicht damit zufrieden, das System alle zwei Wochen neu zu bauen und in der Zwischenzeit die anfallenden Änderungen zu sammeln. Wenn Sie sich zutrauen, alle Abhängigkeiten zu erkennen, bauen Sie vielleicht auch nur die geänderten Sachen neu.

Das hängt natürlich auch noch davon ab, wie oft Sie ein Update durchführen wollen und ob Sie FreeBSD-STABLE oder FreeBSD-CURRENT benutzen.

2. Der Bau bricht mit vielen `Signal 11`-Fehlern (oder anderen Signalnummern) ab. Was ist da passiert?

Normalerweise zeigen diese Meldungen Hardwarefehler an. Ein Neubau der Welt ist ein guter Belastungstest für Ihre Hardware und zeigt oft Probleme mit dem Speicher auf. Dies äußert sich darin, dass der Compiler mit dem Erhalt von seltsamen Signalen abbricht.

Es liegt garantiert ein Hardwarefehler vor, wenn ein neuer Übersetzungslauf an einer anderen Stelle abbricht.

In diesem Fall können Sie nur einzelne Komponenten Ihres Systems tauschen, um zu bestimmen, welche Komponente den Fehler verursacht.

3. Kann ich `/usr/obj` löschen, wenn ich fertig bin?

Kurze Antwort: Ja.

In `/usr/obj` werden alle Dateien abgelegt, die während der Übersetzungsphase erstellt wurden. Dieses Verzeichnis wird in einem der ersten Schritte der Bauprozedur entfernt. Es macht daher wenig Sinn, dieses Verzeichnis zu behalten und Sie setzen eine Menge Plattenplatz, momentan ungefähr 2 GB, frei, wenn Sie es löschen.

Wenn Sie allerdings genau wissen, was Sie tun, können Sie diesen Schritt bei `make buildworld` auslassen. Nachfolgende Bauprozeduren werden dadurch erheblich schneller, da die meisten Quelldateien nicht mehr neu übersetzt werden. Dafür können aber subtile Abhängigkeitsprobleme entstehen, die dazu führen, dass der Bau auf merkwürdige Weise abbrechen kann. Dies führt häufig zu unnötigen Diskussionen auf den FreeBSD Mailinglisten, wenn sich jemand über einen kaputten Bau beschwert, aber nicht sieht, dass er Probleme hat, weil er eine Abkürzung genommen hat.

4. Kann ein abgebrochener Bau weitergeführt werden?

Das hängt davon ab, wieweit der Bauprozess fortgeschritten ist.

Üblicherweise werden essentielle Werkzeuge, wie gcc(1) und make(1), und die Systembibliotheken während des Bauprozesses neu erstellt (dies ist aber keine allgemein gültige Regel). Die neu erstellen Werkzeuge und Bibliotheken werden dann benutzt, um sich selbst noch einmal zu bauen, und wieder installiert. Anschließend wird das Gesamtsystem mit den neu erstellten Systemdateien gebaut.

Wenn Sie sich im letzten Schritt befinden und Sie wissen, dass Sie dort sind, weil Sie durch die Ausgaben, die Sie ja sichern, der Bauprozedur gesehen haben, können Sie mit ziemlicher Sicherheit den Bau weiterführen:

```
... Fehler beheben ...
# cd /usr/src
# make -DNO_CLEAN all
```

Diese Variablen verhindern, dass `make buildworld` die vorher erstellten Dateien löscht.

Das Sie sich im letzten Schritt der Bauprozedur befinden, erkennen Sie daran, dass Sie in der Ausgabe die folgenden Zeilen finden:

```
------------------------------------------------------------
Building everything..
------------------------------------------------------------
```

Wenn Sie diese Meldung nicht finden, oder sich nicht sicher sind, dann ist es besser, noch einmal ganz von Vorne anzufangen.

5. Wie kann ich den Bauprozess beschleunigen?

- Bauen Sie im Single-User-Modus.
- Legen Sie `/usr/src` und `/usr/obj` in getrennte Dateisysteme auf unterschiedliche Festplatten. Benutzen Sie nach Möglichkeit auch getrennte Platten-Controller.
- Noch besser ist es, diese Dateisysteme auf mehrere Festplatten mit ccd(4) zu verteilen.
- Bauen Sie die “profiled”-Bibliotheken, die Sie wahrscheinlich sowieso nicht brauchen, nicht. `/etc/make.conf` sollte dazu `NO_PROFILE=true` enthalten.
- Setzen Sie die `CFLAGS` in `/etc/make.conf` auf `-O -pipe`. Die Optimierungsstufe `-O2` ist deutlich langsamer und die Performance-Unterschiede zwischen `-O` und `-O2` sind vernachlässigbar klein. `-pipe` veranlasst den Compiler Pipes anstelle von Dateien für die Kommunikation zu benutzen. Dies spart einige Plattenzugriffe, geht aber auf Kosten des Speichers.
- Benutzen Sie `-j`*n*, um mehrere Prozesse parallel laufen zu lassen. Normalerweise beschleunigt dies den Bauprozess unabhängig davon, ob Sie ein Einprozessor- oder Mehrprozessorsystem einsetzen.
- Sie können das Dateisystem `/usr/src` mit der Option `noatime` einhängen. Dies verhindert, dass die Zugriffszeiten der Dateien aktualisiert werden (eine Information, die Sie vielleicht gar nicht brauchen).

  ```
  # mount -u -o noatime /usr/src
  ```

  > **Warnung:** Das Beispiel geht davon aus, dass sich `/usr/src` auf einem separaten Dateisystem befindet. Wenn das nicht der Fall ist, weil das Verzeichnis beispielsweise Teil des `/usr` Dateisystems ist, müssen Sie anstelle von `/usr/src` den Mountpoint des Dateisystems angeben.

- Das Dateisystem, in dem sich `/usr/obj` befindet, kann mit der Option `async` eingehangen werden. Dies bewirkt, dass Schreibzugriffe auf die Platte asynchron stattfinden, das heißt ein Schreibzugriff ist sofort beendet, die Daten werden allerdings erst einige Sekunden später geschrieben. Dadurch können Schreibzugriffe zusammengefasst werden, was einen erheblichen Geschwindigkeitszuwachs mit sich bringen kann.

  > **Warnung:** Beachten Sie, dass dies Ihr Dateisystem anfälliger für Fehler macht. Im Fall eines Stromausfalls besteht eine erhöhte Wahrscheinlichkeit, dass das Dateisystem beim Start der Maschine zerstört ist.

Wenn sich `/usr/obj` auf einem extra Dateisystem befindet, ist das kein Problem. Wenn sich allerdings auf diesem Dateisystem noch andere wertvolle Daten befinden, stellen Sie sicher, dass Sie aktuelle Sicherungen besitzen.

```
# mount -u -o async /usr/obj
```

Warnung: Ersetzen Sie `/usr/obj` durch den Mountpoint des entsprechenden Dateisystems, wenn es sich nicht auf einem eigenen Dateisystem befindet.

6. Was mache ich, wenn etwas nicht funktioniert?

Stellen Sie sicher, dass sich in Ihrer Umgebung keine Reste eines vorherigen Baus befinden. Das geht ganz einfach:

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
# make cleandir
```

Ja, `make cleandir` muss wirklich zweimal aufgerufen werden.

Nachdem Sie aufgeräumt haben, starten Sie den Bauprozess wieder mit `make buildworld`.

Wenn Sie immer noch Probleme haben, schicken Sie die Fehlermeldungen und die Ausgabe von `uname -a` an die Mailingliste ’Fragen und Antworten zu FreeBSD’ <`de-bsd-questions@de.FreeBSD.org`>. Bereiten Sie sich darauf vor, weitere Fragen zu Ihrer Umgebung zu beantworten.

25.8. Veraltete Dateien, Verzeichnisse und Bibliotheken löschen

Basiert auf Notizen von Anton Shterenlikht.

Aufgrund der ständigen Weiterentwicklung von FreeBSD kann es dazu kommen, dass Dateien (und deren Inhalte) obsolet werden, weil deren Funktionalität entweder in anderen Dateien implementiert wurde, sich die Versionsnummer der Bibliothek geändert hat oder deren Funktion nicht mehr benötigt wird. Dies kann sowohl Dateien und Verzeichnis, aber auch Bibliotheken betreffen. Diese veralteten Dateien sollten daher entfernt werden, bevor Sie Ihr System aktualisieren. Der Vorteil für den Benutzer ist darin zu sehen, dass dessen System (sowie dessen Backup) von nicht mehr benötigten Dateien gereinigt wird. Falls die obsolete Bibliothek Sicherheits- oder Stabilitätsprobleme aufweist, sollte das System ebenfalls aktualisiert werden, um Ihr System sicher zu halten und/oder durch die fehlerhafte Bibliothek verursachte Systemabstürze zu vermeiden. Veraltete Dateien, Verzeichnisse und Bibliotheken sind in der Datei `/usr/src/ObsoleteFiles.inc` aufgelistet. Die folgenden Anweisungen sollen Ihnen dabei helfen, diese Dateien während der Systemaktualisierung zu entfernen.

Im Folgenden wird angenommen, dass Sie den Anweisungen von Abschnitt 25.7.1 folgen. Nachdem Sie `make installworld` sowie `mergemaster` erfolgreich ausgeführt haben, sollten Sie Ihr System auf veraltete Dateien und Bibliotheken hin überprüfen:

```
# cd /usr/src
# make check-old
```

Werden dabei veraltete Dateien gefunden, können diese im nächsten Schritt entfernt werden:

```
# make delete-old
```

> **Tipp:** Weitere interessante Targets finden sich in der Datei `/usr/src/Makefile`.

Bei jeder Datei wird nachgefragt, ob Sie diese wirklich löschen wollen. Es ist aber auch möglich, alle Dateien automatisch löschen zu lassen. Dies erreichen Sie, indem Sie die Umgebungsvariable `BATCH_DELETE_OLD_FILES` entsprechend setzen:

```
# make -DBATCH_DELETE_OLD_FILES delete-old
```

Alternativ können Sie auch den folgenden Befehl einsetzen (und jeweils die Antwort `yes` an die einzelnen Abfragen weiterreichen):

```
# yes | make delete-old
```

> **Warnung:** Das Löschen veralteter Dateien kann dazu führen, dass Programme, die auf diese Dateien angewiesen sind, nicht mehr funktionieren. Dies gilt insbesondere für veraltete Bibliotheken. In den meisten Fällen ist es dann notwendig, Programme, Ports und Bibliotheken, welche die veraltete Bibliothek verwenden, neu zu bauen, bevor Sie den Befehl `make delete-old-libs` ausführen.

Die Ports-Sammlung enthält Werkzeuge, die Ihnen beim Prüfen von Bibliothek-Abhängigkeiten helfen können: `sysutils/libchk` sowie `sysutils/bsdadminscripts`.

Veraltete Bibliotheken können zu Konflikten mit neueren Bibliotheken führen und beispielsweise folgende Meldungen verursachen:

```
/usr/bin/ld: warning: libz.so.4, needed by /usr/local/lib/libtiff.so, may conflict with libz.so.5
/usr/bin/ld: warning: librpcsvc.so.4, needed by /usr/local/lib/libXext.so, may conflict with libr
```

Um diese Probleme zu lösen, müssen Sie zuerst herausfinden, welcher Port die Bibliothek installiert hat:

```
# pkg_info -W  /usr/local/lib/libtiff.so
/usr/local/lib/libtiff.so was installed by package tiff-3.9.4
# pkg_info -W /usr/local/lib/libXext.so
/usr/local/lib/libXext.so was installed by package libXext-1.1.1,1
```

Danach deinstallieren Sie den Port und bauen ihn neu, um ihn danach erneut zu installieren. Dieser Vorgang kann durch den Einsatz der Werkzeuge `ports-mgmt/portmaster` oder `ports-mgmt/portupgrade` automatisiert werden. Nachdem Sie alle Ports erfolgreich neu gebaut haben (und Sie daher keine veralteten Bibliotheken mehr verwenden) können Sie die veralteten Bibliotheken endgültig entfernen:

```
# make delete-old-libs
```

25.9. Installation mehrerer Maschinen

Beigetragen von Mike Meyer.

Wenn Sie mehrere Maschinen besitzen, die Sie alle auf dem gleichen Stand halten wollen, ist es eine Verschwendung von Ressourcen, die Quellen auf jeder Maschine vorzuhalten und zu übersetzen. Die Lösung dazu ist, eine Maschine den Großteil der Arbeit durchführen zu lassen und den anderen Maschinen das Ergebnis mit NFS zur Verfügung zu stellen. Dieser Abschnitt zeigt Ihnen wie das geht.

25.9.1. Voraussetzungen

Stellen Sie zuerst eine Liste der Maschinen zusammen, die auf demselben Stand sein sollen. Wir nennen diese Maschinen die *Baugruppe*. Jede dieser Maschinen kann mit einem eigenen Kernel laufen, doch sind die Programme des Userlands auf allen Maschinen gleich. Wählen Sie aus der Baugruppe eine Maschine aus, auf der der Bau durchgeführt wird, den *Bau-Master*. Dies sollte eine Maschine sein, die über die nötigen Ressourcen für `make buildworld` und `make installworld` verfügt. Sie brauchen auch eine *Testmaschine*, auf der Sie die Updates testen, bevor Sie sie in Produktion installieren. Dies sollte eine Maschine, eventuell der Bau-Master, sein, die über einen längeren Zeitraum nicht zur Verfügung stehen kann.

Alle Maschinen der Baugruppe müssen `/usr/obj` und `/usr/src` von derselben Maschine an gleichem Ort einhängen. Idealerweise befinden sich die beiden Verzeichnisse auf dem Bau-Master auf verschiedenen Festplatten, sie können allerdings auch auf dem Bau-Master über NFS zur Verfügung gestellt werden. Wenn Sie mehrere Baugruppen haben, sollte sich `/usr/src` auf einem Bau-Master befinden und über NFS für den Rest der Maschinen zur Verfügung gestellt werden.

Stellen Sie sicher, dass `/etc/make.conf` und `/etc/src.conf` auf allen Maschinen einer Baugruppe mit der Datei des Bau-Masters übereinstimmt. Der Bau-Master muss jeden Teil des Systems bauen, den irgendeine Maschine der Baugruppe benötigt. Auf dem Bau-Master müssen in `/etc/make.conf` alle zu bauenden Kernel mit der Variablen `KERNCONF` bekannt gegeben werden. Geben Sie dabei den Kernel des Bau-Masters zuerst an. Für jeden zu bauenden Kernel muss auf dem Bau-Master die entsprechende Konfigurationsdatei unter `/usr/src/sys/`*`arch`*`/conf` abgelegt werden.

25.9.2. Installation des Basissystems

Nach diesen Vorbereitungen können Sie mit dem Bau beginnen. Bauen Sie auf dem Bau-Master, wie in Abschnitt 25.7.7.2 beschrieben, den Kernel und die Welt, installieren Sie aber nichts. Wechseln Sie auf die Testmaschine und installieren Sie den gerade gebauten Kernel. Wenn diese Maschine `/usr/src` und `/usr/obj` über NFS bekommt, müssen Sie das Netzwerk im Single-User-Modus aktivieren und die beiden Dateisysteme einhängen. Am einfachsten ist dies, wenn Sie auf der Testmaschine ausgehend vom Mehrbenutzermodus mit `shutdown now` in den Single-User-Modus wechseln. Sie können dann mit der normalen Prozedur den neuen Kernel und das System installieren und anschließend `mergemaster` laufen lassen. Wenn Sie damit fertig sind, können Sie die Maschine wieder in den Mehrbenutzermodus booten.

Nachdem Sie sichergestellt haben, dass die Testmaschine einwandfrei funktioniert, wiederholen Sie diese Prozedur für jede Maschine in der Baugruppe.

25.9.3. Die Ports-Sammlung

Dasselbe Verfahren können Sie auch für die Ports-Sammlung anwenden. Zuerst müssen alle Maschinen einer Baugruppe `/usr/ports` von derselben Maschine über NFS zur Verfügung gestellt bekommen. Setzen Sie dann ein Verzeichnis für die Quellen auf, das sich alle Maschinen teilen. Dieses Verzeichnis können Sie in `/etc/make.conf` mit der Variablen `DISTDIR` angeben. Das Verzeichnis sollte für den Benutzer beschreibbar sein, auf den der Benutzer `root` vom NFS Subsystem abgebildet wird. Jede Maschine sollte noch `WRKDIRPREFIX` auf ein lokales Bauverzeichnis setzen. Wenn Sie vorhaben, Pakete zu bauen und zu verteilen, sollten Sie `PACKAGES` auf ein Verzeichnis mit den gleichen Eigenschaften wie `DISTDIR` setzen.

Fußnoten

1. Das stimmt nicht ganz. Obwohl wir alte FreeBSD Releases für einige Jahre unterstützen, können wir sie nicht ewig unterstützen. Eine vollständige Beschreibung der Sicherheitspolitik für alte FreeBSD Releases entnehmen Sie bitte http://www.FreeBSD.org/security/.
2. Von engl. *to pull = ziehen*. Der Client holt sich bei dieser Methode die Dateien ab.
3. Von engl. *to push = schieben*. Der Server schickt dem Client die Dateien.

Kapitel 26. DTrace

Written by Tom Rhodes. Übersetzt von Benedict Reuschling und Christoph Sold.

26.1. Überblick

DTrace, auch bekannt als Dynamic Tracing, wurde von Sun als ein Werkzeug zur Analyse von Performance-Problemen in Produktiv- und Entwicklungssystemen entwickelt. Es ist kein Debugging-Werkzeug, sondern ein Hilfsmittel für Echtzeit-Systemanalysen.

DTrace ist ein bemerkenswertes Werkzeug zur Profilerstellung, mit einer beeindruckenden Palette von Eigenschaften zur Diagnose von Systemereignissen. Es kann auch dazu verwendet werden, bestehende Skripte ablaufen zu lassen, um einen Nutzen aus deren Möglichkeiten zu ziehen. Nutzer können mittels der Programmiersprache D von DTrace ihre eigenen Hilfsmittel schreiben, was es ermöglicht, die eigenen Profile nach Ihren Bedürfnissen anzupassen.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Was DTrace ist und welche Funktionen es zur Verfügung stellt.
- Unterschiede zwischen der Solaris DTrace Implementierung und derjenigen, die FreeBSD bereitstellt.
- Wie man DTrace auf FreeBSD aktiviert und verwendet.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- UNIX und FreeBSD Grundlagen verstehen (Kapitel 4).
- Einen Kernel konfigurieren und kompilieren können (Kapitel 9).
- Vertraut sein mit Sicherheitsaspekten und wie diese FreeBSD betreffen (Kapitel 15).
- Verstehen, wie man den Quellcode von FreeBSD beziehen und das Betriebssystem neu erstellen kann (Kapitel 25).

> **Warnung:** Diese Funktion ist als experimentell anzusehen. Manche Einstellungen enthalten möglicherweise nicht alle Funktionalitäten, andere Teile könnten gar nicht laufen. Mit der Zeit, wenn diese Funktion als für den Produktivbetrieb geeignet erscheint, wird auch diese Dokumentation geändert, um diesem Umstand gerecht zu werden.

26.2. Unterschiede in der Implementierung

Obwohl DTrace in FreeBSD sehr ähnlich zu dem in Solaris ist, existieren doch Unterschiede, die vorher erklärt werden müssen. Der Hauptunterschied für die Anwender besteht darin, dass in FreeBSD DTrace explizit aktiviert werden muss. Es existieren Kerneloptionen und Module, die aktiviert sein müssen, damit DTrace korrekt arbeitet. Diese werden später genauer erlätert.

Die Kerneloption `DDB_CTF` wird dafür verwendet, um die Unterstützung im Kernel für das Laden von CTF-Daten aus Kernelmodulen und dem Kernel selbst zu ermöglichen. CTF ist das Compact C Type Format von Solaris, welches eine reduzierte Form von Debug-Informationen kapselt, ähnlich zu DWARF und den antiken Stabs. Diese CTF-Daten werden dem Binärcode von den `ctfconvert` und `ctfmerge` Befehlen den Werkzeugen zum Bauen des

Systems hinzugefügt. Das `ctfconvert`-Dienstprogramm parst die vom Compiler erstellten DWARF ELF Debug-Abschnitte und `ctfmerge` vereint CTF ELF-Abschnitte aus Objekten, entweder in ausführbare Dateien oder Shared-Libraries. In Kürze erfahren Sie, wie Sie dies für den Kernel und den Bau von FreeBSD aktivieren.

Einige Provider in FreeBSD unterscheiden sich von der Solaris-Implementierung. Am deutlichsten wird das beim `dtmalloc`-Provider, welcher das Aufzeichnen von `malloc()` nach Typen im FreeBSD-Kernel ermöglicht.

In FreeBSD darf DTrace wegen unterschiedlicher Sicherheitskonzepte nur von `root` verwendet werden. Solaris besitzt ein paar Audit-Funktionen auf den unteren Ebenen, die noch nicht in FreeBSD implementiert sind. Deshalb kann nur `root` auf `/dev/dtrace/dtrace` zugreifen.

Zum Schluss muss noch erwähnt werden, dass die DTrace-Software unter Suns CDDL Lizenz fällt. Die `Common Development and Distribution License` wird von FreeBSD mitgeliefert, sehen Sie sich dazu `/usr/src/cddl/contrib/opensolaris/OPENSOLARIS.LICENSE` an, oder lesen Sie die Online-Version unter http://www.opensolaris.org/os/licensing.

Diese Lizenz bedeutet, dass ein FreeBSD-Kernel mit den DTrace-Optionen immer noch BSD-lizenziert ist; allerdings tritt die CDDL in Kraft, wenn Module in Binärform vertrieben werden oder die Binärdateien geladen werden.

26.3. Die DTrace Unterstützung aktivieren

Um Unterstützung für DTrace zu aktivieren, fügen Sie die folgenden Zeilen zu Ihrer Kernelkonfigurationsdatei hinzu:

```
options         KDTRACE_HOOKS
options         DDB_CTF
```

> **Anmerkung:** Besitzer der AMD-Architektur werden wahrscheinlich noch die folgende Zeile zur Kernelkonfigurationsdatei hinzufügen:
>
> ```
> options         KDTRACE_FRAME
> ```
>
> Diese Option liefert die Unterstützung für die FBT-Eigenschaft. DTrace wird auch ohne diese Option funktionieren; jedoch wird dann Function Boundary Tracing nur eingeschränkt unterstützt.

Der gesamte Quellcode muss neu gebaut und mit der CTF-Option installiert werden. Um das zu erreichen, bauen Sie FreeBSD aus dem Quellcode mittels:

```
# cd /usr/src

# make WITH_CTF=1 kernel
```

Das System muss im Anschluss daran neu gestartet werden.

Nachdem das System neu gestartet und der neue Kernel in den Hauptspeicher geladen wurde, sollte die Unterstützung für die Korn-Shell hinzugefügt werden. Dies wird benötigt, da die Sammlung von DTrace-Werkzeugen mehrere Dienstprogramme enthält, die in `ksh` implementiert sind. Installieren Sie `shells/ksh93`. Es ist auch möglich, diese Werkzeuge unter `shells/pdksh` oder `shells/mksh` laufen zu lassen.

Zum Schluss sollten Sie noch den aktuellen DTrace-Werkzeugsatz beschaffen. Die aktuelle Version ist unter http://www.opensolaris.org/os/community/dtrace/dtracetoolkit/ verfügbar. Ein Mechanismus zur Installation ist

enthalten, allerdings ist eine Installation nicht unbedingt nötig, um die darin enthaltenen Dienstprogramme einzusetzen.

26.4. DTrace verwenden

Bevor die DTrace-Funktionaltät benutzt werden kann, muss das DTrace-Gerät existieren. Um das Gerät zu laden, geben Sie das folgende Kommando ein:

```
# kldload dtraceall
```

Die DTrace-Unterstützung sollte jetzt verfügbar sein. Um alle Sonden anzuzeigen, kann der Administrator nun den folgenden Befehl eingeben:

```
# dtrace -l | more
```

Alle Ausgaben werden an das `more`-Programm übergeben, da der Bildschirmpuffer sehr schnell überlaufen wird. Ab diesem Punkt kann DTrace als einsatzbereit angesehen werden. Jetzt ist es an der Zeit, sich näher mit dem Satz von Werkzeugen zu beschäftigen.

Der Werkzeugsatz ist eine Sammlung von vorgefertigten Skripten, die von DTrace ausgeführt werden können, um Systeminformationen zu sammeln. Es gibt Skripte, die offene Dateien überprüfen, den Speicher, CPU-Verbrauch und noch viel mehr. Entpacken Sie die Skripte mit dem folgenden Befehl:

```
# gunzip -c DTraceToolkit* | tar xvf -
```

Wechseln Sie mit dem `cd`-Kommando in dieses Verzeichnis und ändern Sie die Berechtigung zum Ausführen von allen Dateien, deren Name klein geschrieben ist, auf `755`.

All diese Skripte müssen inhaltlich verändert werden. Diejenigen, die auf `/usr/bin/ksh` verweisen, müssen in `/usr/local/bin/ksh` geändert werden und die Anderen, welche `/usr/bin/sh` verwenden, müssen so angepasst werden, dass sie `/bin/sh` verwenden. Schliesslich müssen noch diejenigen, die `/usr/bin/perl` enthalten, auf `/usr/local/bin/perl` umgeschrieben werden.

> **Wichtig:** Zu diesem Zeitpunkt ist es klug, den Leser noch einmal daran zu erinnern, dass die Unterstützung von DTrace in FreeBSD noch *unvollständig* und *experimentell* ist. Viele dieser Skripte werden nicht funktionieren, da diese entweder zu sehr Solaris-spezifisch sind oder Sonden verwenden, die zur Zeit noch nicht unterstützt werden.

Zum Zeitpunkt, an dem dieses Dokument geschrieben wurde, existieren nur zwei Skripte im DTrace-Werkzeugsatz, die von FreeBSD komplett unterstützt werden: die Skripte `hotkernel` und `procsystime`. Diese beiden werden in den folgenden Teilen dieses Abschnitts genauer untersucht.

`hotkernel` wurde entworfen, um zu identifizieren, welche Funktion die meiste Kernelzeit beansprucht. Normal ausgeführt, wird es Ausgaben ähnlich der Folgenden produzieren:

```
# ./hotkernel
Sampling... Hit Ctrl-C to end.
```

Der Systemadministrator muss die Tastenkombination **Ctrl**+**C** drücken, um den Prozess zu stoppen. Nach dem Abbruch wird das Skript eine Liste von Kernelfunktionen und Zeitmessungen ausgeben, aufsteigend sortiert nach den Zeiten:

```
kernel‘_thread_lock_flags                               2   0.0%
0xc1097063                                              2   0.0%
kernel‘sched_userret                                    2   0.0%
kernel‘kern_select                                      2   0.0%
kernel‘generic_copyin                                   3   0.0%
kernel‘_mtx_assert                                      3   0.0%
kernel‘vm_fault                                         3   0.0%
kernel‘sopoll_generic                                   3   0.0%
kernel‘fixup_filename                                   4   0.0%
kernel‘_isitmyx                                         4   0.0%
kernel‘find_instance                                    4   0.0%
kernel‘_mtx_unlock_flags                                5   0.0%
kernel‘syscall                                          5   0.0%
kernel‘DELAY                                            5   0.0%
0xc108a253                                              6   0.0%
kernel‘witness_lock                                     7   0.0%
kernel‘read_aux_data_no_wait                            7   0.0%
kernel‘Xint0x80_syscall                                 7   0.0%
kernel‘witness_checkorder                               7   0.0%
kernel‘sse2_pagezero                                    8   0.0%
kernel‘strncmp                                          9   0.0%
kernel‘spinlock_exit                                   10   0.0%
kernel‘_mtx_lock_flags                                 11   0.0%
kernel‘witness_unlock                                  15   0.0%
kernel‘sched_idletd                                   137   0.3%
0xc10981a5                                          42139  99.3%
```

Dieses Skript funktioniert auch mit Kernelmodulen. Um diese Eigenschaft zu verwenden, starten Sie das Skript mit dem Parameter `-m`:

```
# ./hotkernel -m
Sampling... Hit Ctrl-C to end.
^C
MODULE                                              COUNT   PCNT
0xc107882e                                              1   0.0%
0xc10e6aa4                                              1   0.0%
0xc1076983                                              1   0.0%
0xc109708a                                              1   0.0%
0xc1075a5d                                              1   0.0%
0xc1077325                                              1   0.0%
0xc108a245                                              1   0.0%
0xc107730d                                              1   0.0%
0xc1097063                                              2   0.0%
0xc108a253                                             73   0.0%
kernel                                                874   0.4%
0xc10981a5                                         213781  99.6%
```

Das `procsystime` Skript fängt die Systemaufruf-Zeiten ab und zeigt diese für eine gegebene PID oder einen Prozessnamen an. Im folgenden Beispiel wurde eine neue Instanz von `/bin/csh` erzeugt. `procsystime` wurde

ausgeführt und verbleibt so, während ein paar Befehle in die andere Instanz von `csh` eingegeben werden. Dies sind die Ergebnisse dieses Versuchs:

```
# ./procsystime -n csh
Tracing... Hit Ctrl-C to end...
^C

Elapsed Times for processes csh,

         SYSCALL          TIME (ns)
          getpid               6131
       sigreturn               8121
           close              19127
           fcntl              19959
             dup              26955
         setpgid              28070
            stat              31899
       setitimer              40938
           wait4              62717
       sigaction              67372
     sigprocmask             119091
    gettimeofday             183710
           write             263242
          execve             492547
           ioctl             770073
           vfork            3258923
      sigsuspend            6985124
            read         3988049784
```

Wie aus der Ausgabe ersichtlich ist, verbraucht der `read()`-Systemaufruf die meiste Zeit in Nanosekunden, während der Systemaufruf `getpid()` hingegen am schnellsten läft.

26.5. Die Sprache D

Der DTrace Werkzeugsatz enthält viele Skripte in der speziellen Sprache von DTrace. Diese Sprache wird als “die D Sprache” in der Dokumentation von Sun bezeichnet und ist C++ sehr ähnlich. Eine tiefergehende Betrachtung dieser Sprache würde den Rahmen dieses Dokuments sprengen. Ausführlich wird diese Sprache unter http://wikis.sun.com/display/DTrace/Documentation behandelt.

IV. Netzwerke

FreeBSD ist eins der meist benutzten Betriebssysteme für leistungsfähige Netzwerkserver. Die Kapitel in diesem Teil behandeln die nachstehenden Themen:

- Serielle Datenübertragung
- PPP und PPP over Ethernet
- Electronic-Mail
- Den Betrieb von Netzwerkdiensten
- Firewalls
- Weiterführende Netzwerkthemen

Diese Kapitel sollten Sie lesen, wenn Sie die Informationen darin benötigen. Sie brauchen Sie nicht in einer bestimmten Reihenfolge zu lesen, noch müssen Sie die Kapitel lesen, bevor Sie anfangen, FreeBSD zu benutzen.

Kapitel 27. Serielle Datenübertragung

Übersetzt von Martin Heinen.

27.1. Übersicht

UNIX Systeme unterstützten schon immer die serielle Datenübertragung. Tatsächlich wurden Ein- und Ausgaben auf den ersten UNIX Maschinen über serielle Leitungen durchgeführt. Seit der Zeit, in der ein durchschnittlicher "Terminal" aus einem seriellen Drucker mit 10 Zeichen/Sekunde und einer Tastatur bestand, hat sich viel verändert. Dieses Kapitel behandelt einige Möglichkeiten, serielle Datenübertragung unter FreeBSD zu verwenden.

Nachdem Sie dieses Kapitel durchgearbeitet haben, werden Sie Folgendes wissen:

- Wie Sie Terminals an Ihr FreeBSD anschließen.
- Wie Sie sich mit einem Modem auf einem entfernten Rechner einwählen.
- Wie Sie entfernten Benutzern erlauben, sich mit einem Modem in Ihr System einzuwählen.
- Wie Sie Ihr System über eine serielle Konsole booten.

Bevor Sie dieses Kapitel lesen, sollten Sie

- einen neuen Kernel konfigurieren und installieren können (Kapitel 9).
- Das Berechtigungskonzept von UNIX und Prozesse verstehen (Kapitel 4).
- Zudem sollten Sie Zugriff auf die Handbücher der seriellen Komponenten (Modem oder Multiportkarte) haben, die Sie mit FreeBSD verwenden wollen.

27.2. Einführung

> **Warnung:** Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

27.2.1. Begriffe

bps

Bits pro Sekunde – Einheit für die Übertragungsgeschwindigkeit.

DEE (DTE)

Datenendeinrichtung (Data Terminal Equipment) – zum Beispiel Ihr Computer.

DÜE (DCE)

Datenübertragungseinrichtung (Data Communications Equipment) – Ein Modem.

RS-232

EIA (Electronic Industries Association) Norm für die serielle Datenübertragung.

In diesem Abschnitt wird der Begriff “Baud” nicht für Übertragungsgeschwindigkeiten gebraucht. Baud bezeichnet elektrische Zustandswechsel pro Zeiteinheit, die Taktfrequenz, während “bps” (Bits pro Sekunde) der *richtige* Begriff für die Übertragungsgeschwindigkeit ist (die meisten Pedanten sollten damit zufrieden sein).

27.2.2. Kabel und Schnittstellen

Um ein Modem oder einen Terminal an Ihr FreeBSD-System anzuschließen, muss Ihr Computer über eine serielle Schnittstelle verfügen. Zusätzlich brauchen Sie noch das passende Kabel, um das Gerät mit der Schnittstelle zu verbinden. Wenn Sie mit Ihren Geräten und den nötigen Kabeln schon vertraut sind, können Sie diesen Abschnitt überspringen.

27.2.2.1. Kabel

Es gibt verschiedene serielle Kabel. Die zwei häufigsten sind Nullmodemkabel und Standard-RS-232-Kabel. Die Dokumentation Ihrer Hardware sollte beschreiben, welchen Kabeltyp Sie benötigen.

27.2.2.1.1. Nullmodemkabel

Ein Nullmodemkabel verbindet einige Signale, wie die Betriebserde, eins zu eins, andere Signale werden getauscht: Die Sende- und Empfangsleitungen werden zum Beispiel gekreuzt.

Sie können das Kabel für die Anbindung eines Terminals auch selbst herstellen. Die folgende Tabelle enthält die Signalnamen von RS-232C sowie die Pinbelegung für einen Stecker vom Typ DB-25. Beachten Sie dabei aber, dass der Standard zwar eine direkte Verbindung beider Pin 1 (*Protective Ground*) vorschreibt, diese aber in vielen Fällen nicht vorhanden ist. Einige Terminals benötigen nur die Pins 2, 3 und 7 für eine korrekte Funktion, während andere eine unterschiedliche Konfiguration als die in den folgenden Beispielen gezeigte benötigen.

Tabelle 27-1. Nullmodemkabel vom Typ DB-25-zu-DB-25

Signal	Pin #		Pin #	Signal
SG	7	verbunden mit	7	SG
TD	2	verbunden mit	3	RD
RD	3	verbunden mit	2	TD
RTS	4	verbunden mit	5	CTS
CTS	5	verbunden mit	4	RTS
DTR	20	verbunden mit	6	DSR
DTR	20	verbunden mit	8	DCD
DSR	6	verbunden mit	20	DTR

Signal	Pin #		Pin #	Signal
DCD	8	verbunden mit	20	DTR

Die folgenden zwei Schemata werden heutzutage ebenfalls häufig eingesetzt:

Tabelle 27-2. Nullmodemkabel vom Typ DB-9-zu-DB-9

Signal	Pin #		Pin #	Signal
RD	2	verbunden mit	3	TD
TD	3	verbunden mit	2	RD
DTR	4	verbunden mit	6	DSR
DTR	4	verbunden mit	1	DCD
SG	5	verbunden mit	5	SG
DSR	6	verbunden mit	4	DTR
DCD	1	verbunden mit	4	DTR
RTS	7	verbunden mit	8	CTS
CTS	8	verbunden mit	7	RTS

Tabelle 27-3. Nullmodemkabel vom Typ DB-9-zu-DB-25

Signal	Pin #		Pin #	Signal
RD	2	verbunden mit	2	TD
TD	3	verbunden mit	3	RD
DTR	4	verbunden mit	6	DSR
DTR	4	verbunden mit	8	DCD
SG	5	verbunden mit	7	SG
DSR	6	verbunden mit	20	DTR
DCD	1	verbunden mit	20	DTR
RTS	7	verbunden mit	5	CTS
CTS	8	verbunden mit	4	RTS

> **Anmerkung:** Wird ein Pin eines Kabels mit zwei Pins des anderen Kabels verbunden, werden dazu in der Regel zuerst die beiden Pins mit einem kurzem Draht verbunden. Danach wird dieser Draht mit dem Pin des anderen Endes verbunden.

Die eben besprochenen Schemata scheinen die beliebtesten zu sein. Es gibt aber noch weitere Varianten. Im Buch *RS-232 Made Easy* wird beispielsweise SG mit SG verbunden, TD mit RD, RTS und CTS mit DCD, DTR mit DSR, und umgekehrt.

27.2.2.1.2. Standard RS-232C Kabel

Ein Standard-RS-232C-Kabel verbindet alle Signale direkt, das heißt das Signal “Transmitted Data” wird mit dem

Signal “Transmitted Data” der Gegenstelle verbunden. Dieses Kabel wird benötigt, um ein Modem mit einem FreeBSD-System zu verbinden. Manche Terminals benötigen dieses Kabel ebenfalls.

27.2.2.2. Schnittstellen

Über serielle Schnittstellen werden Daten zwischen dem FreeBSD-System und dem Terminal übertragen. Dieser Abschnitt beschreibt die verschiedenen Schnittstellen und wie sie unter FreeBSD angesprochen werden.

27.2.2.2.1. Arten von Schnittstellen

Da es verschiedene Schnittstellen gibt, sollten Sie vor dem Kauf oder Selbstbau eines Kabels sicherstellen, dass dieses zu den Schnittstellen Ihres Terminals und FreeBSD-Systems passt.

Die meisten Terminals besitzen DB-25-Stecker. Personal Computer haben DB-25- oder DB-9-Stecker. Wenn Sie eine serielle Multiportkarte für Ihren PC besitzen, haben Sie vielleicht RJ-12- oder RJ-45-Anschlüsse.

Die Dokumentation Ihrer Geräte sollte Aufschluss über den Typ der benötigten Anschlüsse geben. Oft hilft es, wenn Sie sich den Anschluss einfach ansehen.

27.2.2.2.2. Schnittstellenbezeichnung

Unter FreeBSD sprechen Sie die serielle Schnittstelle (Port) über einen Eintrag im `/dev` Verzeichnis an. Es gibt dort zwei verschiedene Einträge:

- Schnittstellen für eingehende Verbindungen werden `/dev/ttyu`*`N`* genannt. Dabei ist *`N`* die Nummer der Schnittstelle, deren Zählung bei Null beginnt. Allgemein wird diese Schnittstelle für Terminals benutzt. Diese Schnittstelle funktioniert nur, wenn ein “Data Carrier Detect” Signal (DCD) vorliegt.
- Für ausgehende Verbindungen wird `/dev/cuad`*`N`* verwendet. Dieser Port wird normalerweise nur von Modems genutzt. Sie können ihn allerdings für Terminals benutzen, die das “Data Carrier Detect” Signal nicht unterstützen.

Wenn Sie einen Terminal an die erste serielle Schnittstelle (`COM1` in MS-DOS), angeschlossen haben, sprechen Sie ihn über `/dev/ttyu0` an. Wenn er an der zweiten seriellen Schnittstelle angeschlossen ist, verwenden Sie `/dev/ttyu1`, usw.

27.2.3. Kernelkonfiguration

In der Voreinstellung benutzt FreeBSD vier serielle Schnittstellen, die in MS-DOS-Kreisen als `COM1`, `COM2`, `COM3` und `COM4` bekannt sind. Momentan unterstützt FreeBSD einfache Multiportkarten (z.B. die BocaBoard 1008 und 2016) und bessere wie die von Digiboard und Stallion Technologies. In der Voreinstellung sucht der Kernel allerdings nur nach den Standardanschlüssen.

Um zu überprüfen, ob der Kernel eine Ihrer seriellen Schnittstellen erkennt, achten Sie auf die Meldungen beim Booten, oder schauen sich diese später mit `/sbin/dmesg` an. Insbesondere sollten Sie auf Meldungen achten, die mit den Zeichen `sio` anfangen.

Tipp: Das folgende Kommando zeigt Ihnen nur die Meldungen an, die die Folge `sio` enthalten:

```
# /sbin/dmesg | grep 'sio'
```

Auf einem System mit vier seriellen Schnittstellen sollte der Kernel die folgenden Meldungen ausgeben:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

Wenn Ihr Kernel nicht alle seriellen Schnittstellen erkennt, müssen Sie Ihren Kernel über die Datei `/boot/device.hints` konfigurieren. Zusätzlich können Sie Einträge für Geräte, die auf Ihrem System nicht vorhanden sind, aus dem Kernel entfernen.

Die Hilfeseite sio(4) enthält weitere Informationen zu seriellen Schnittstellen und Multiportkarten. Seien Sie vorsichtig, wenn Sie Konfigurationsdateien von älteren FreeBSD-Versionen verwenden, da sich die Syntax und die Bedeutung der Optionen zwischen verschiedenen Versionen geändert hat.

> **Anmerkung:** `port IO_COM1` ist ein Ersatz für `port 0x3f8`, `IO_COM2` bedeutet `port 0x2f8`, `IO_COM3` bedeutet `port 0x3e8` und `IO_COM4` steht für `port 0x2e8`. Die angegebenen IO-Adressen sind genau wie die Interrupts 4, 3, 5 und 9 üblich für serielle Schnittstellen. Beachten Sie bitte, dass sich normale serielle Schnittstellen auf ISA-Bussen *keine* Interrupts teilen können. Multiportkarten besitzen zusätzliche Schaltkreise, die es allen 16550As auf der Karte erlauben, sich einen oder zwei Interrupts zu teilen.

27.2.4. Gerätedateien

Die meisten Geräte im Kernel werden durch Gerätedateien in `/dev` angesprochen. Die `sio` Geräte werden durch `/dev/ttyu`*`N`* für eingehende Verbindungen und durch `/dev/cuad`*`N`* für ausgehende Verbindungen angesprochen. Zum Initialisieren der Geräte stellt FreeBSD die Dateien `/dev/ttyu`*`N`*`.init` und `/dev/cuad`*`N`*`.init` zur Verfügung. Zusätzlich existieren Dateien für das Sperren von Gerätedateien (*Locking*). Dabei handelt es sich um die Dateien `/dev/ttyu`*`N`*`.lock` und `/dev/cuad`*`N`*`.lock`. Diese Dateien werden benutzt, um Kommunikationsparameter beim Öffnen eines Ports vorzugeben. Für Modems, die zur Flusskontrolle `RTS/CTS` benutzen, kann damit `crtscts` gesetzt werden. Die Geräte `/dev/ttyld`*`N`* und `/dev/cuala`*`N`* (locking devices) werden genutzt, um bestimmte Parameter festzuschreiben und vor Veränderungen zu schützen. Weitere Informationen zu Terminals finden Sie in termios(4), sio(4) erklärt die Dateien zum Initialisieren und Sperren der Geräte, stty(1) beschreibt schließlich Terminal-Einstellungen.

27.2.5. Konfiguration der seriellen Schnittstelle

Anwendungen benutzen normalerweise die Geräte `ttyu`*`N`* oder `cuad`*`N`*. Das Gerät besitzt einige Voreinstellungen für Terminal-I/O, wenn es von einem Prozess geöffnet wird. Mit dem folgenden Kommando können Sie sich diese Einstellungen ansehen:

```
# stty -a -f /dev/ttyu1
```

Sie können diese Einstellungen verändern, sie bleiben allerdings nur solange wirksam, bis das Gerät geschlossen wird. Wenn das Gerät danach wieder geöffnet wird, sind die Voreinstellungen wieder wirksam. Um die Voreinstellungen zu ändern, öffnen Sie das Gerät, das zum Initialisieren dient und verändern dessen Einstellungen. Um beispielsweise für `ttyu5` den `CLOCAL` Modus, 8-Bit Kommunikation und `XON/XOFF` Flusssteuerung einzuschalten, setzen Sie das folgende Kommando ab:

```
# stty -f /dev/ttyu5.init clocal cs8 ixon ixoff
```

In `/etc/rc.d/rc.serial` werden die systemweiten Voreinstellungen für serielle Geräte vorgenommen.

Um zu verhindern, dass Einstellungen von Anwendungen verändert werden, können Sie die Geräte zum Festschreiben von Einstellungen (“locking devices”) benutzen. Wenn sie beispielsweise die Geschwindigkeit von `ttyu5` auf 57600 bps festlegen wollen, benutzen Sie das folgende Kommando:

```
# stty -f /dev/ttyld5 57600
```

Eine Anwendung, die `ttyu5` öffnet, kann nun nicht mehr die Geschwindigkeit ändern und muss 57600 bps benutzen.

Die Geräte zum Initialisieren und Festschreiben von Einstellungen sollten selbstverständlich nur von `root` beschreibbar sein.

27.3. Terminals

Beigetragen von Sean Kelly.

Warnung: Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

Wenn Sie sich nicht an der Konsole oder über ein Netzwerk an Ihrem FreeBSD-System anmelden können, sind Terminals ein bequemer und billiger Weg auf Ihr System zuzugreifen. Dieser Abschnitt beschreibt wie Sie Terminals mit FreeBSD benutzen.

27.3.1. Terminaltypen

Das ursprüngliche UNIX System besaß keine Konsolen. Zum Anmelden und Starten von Programmen wurden stattdessen Terminals benutzt, die an den seriellen Schnittstellen des Rechners angeschlossen waren. Dies entspricht der Benutzung eines Modems zum Anmelden auf einem entfernten System, um dort mit einem Terminalemulator im Textmodus zu arbeiten.

Die Konsolen heutiger PCs besitzen sehr gute Grafikfähigkeiten, trotzdem gibt es in fast jedem UNIX System die Möglichkeit, sich über die serielle Schnittstelle anzumelden; FreeBSD ist da keine Ausnahme. Sie können sich an einem Terminal anmelden und dort jedes Textprogramm, das Sie normalerweise an der Konsole oder in einem `xterm` Fenster im X Window System benutzen, laufen lassen.

Im kommerziellen Umfeld können Sie viele Terminals an ein FreeBSD-System anschließen und diese auf den Arbeitsplätzen Ihrer Angestellten aufstellen. Im privaten Umfeld kann ein älterer IBM PC oder Macintosh als Terminal dienen. Damit verwandeln Sie einen Einzelarbeitsplatz in ein leistungsfähiges Mehrbenutzersystem.

FreeBSD kennt drei verschiedene Terminals:

- Dumb terminals,
- PCs, die als Terminals fungieren,
- X Terminals.

Die folgenden Abschnitte beschreiben jeden dieser Terminals.

27.3.1.1. Dumb-Terminals

Dumb-Terminals (unintelligente Datenstationen) sind Geräte, die über die serielle Schnittstelle mit einem Rechner verbunden werden. Sie werden “unintelligent” genannt, weil sie nur Text senden und empfangen und keine Programme laufen lassen können. Alle Programme, wie Texteditoren, Compiler oder Spiele befinden sich auf dem Rechner, der mit dem Terminal verbunden ist.

Es gibt viele Dumb-Terminals, die von verschiedenen Herstellern produziert werden, wie zum Beispiel der VT-100 von Digital Equipment Corporation oder der WY-75 von Wyse. So gut wie jeder der verschiedenen Terminals sollte mit FreeBSD zusammenarbeiten. Manche High-End Geräte verfügen sogar über Grafikfähigkeiten, die allerdings nur von spezieller Software genutzt werden kann.

Dumb-Terminals sind in Umgebungen beliebt, in denen keine Grafikanwendungen, wie zum Beispiel X-Programme, laufen müssen.

27.3.1.2. PCs, die als Terminal fungieren

Jeder PC kann die Funktion eines Dumb-Terminals, der ja nur Text senden und empfangen kann, übernehmen. Dazu brauchen Sie nur das richtige Kabel und eine *Terminalemulation*, die auf dem PC läuft.

Diese Konfiguration ist im privaten Umfeld sehr beliebt. Wenn Ihr Ehepartner zum Beispiel gerade an der FreeBSD-Konsole arbeitet, können Sie einen weniger leistungsstarken PC, der als Terminal mit dem FreeBSD-System verbunden ist, benutzen, um dort gleichzeitig im Textmodus zu arbeiten.

Bereits im Basissystem sind mindestens zwei Werkzeuge vorhanden, die Sie zur Arbeit über eine serielle Konsole einsetzen können: cu(1) sowie tip(1).

Um sich von einem FreeBSD-System aus über eine serielle Verbindung mit einem anderen System zu verbinden, geben Sie folgenden Befehl ein:

```
# cu -l serial-port-device
```

“serial-port-device” ist hier der Name der Gerätedatei, die einer bestimmten seriellen Schnittstelle Ihres Systems zugewiesen ist. Diese Gerätedateien werden `/dev/cuad`*`N`* genannt.

Der Buchstabe “N” muss dabei durch die Nummer des seriellen Ports Ihres Systems ersetzt werden.

> **Anmerkung:** Beachten Sie, dass die Numerierung dieses Daten (im Gegensatz etwa zu MS-DOS-kompatiblen Systemen) unter FreeBSD mit Null und nicht mit Eins beginnt. Die Schnittstelle “COM1” entspricht daher in der Regel `/dev/cuad0` unter FreeBSD.

Anmerkung: In der Ports-Sammlung finden sich weitere Programme (beispielsweise `comms/minicom`), mit denen Sie eine Verbindung über eine serielle Schnittstelle herstellen können.

27.3.1.3. X-Terminals

X-Terminals sind die ausgereiftesten der verfügbaren Terminals. Sie werden nicht mit der seriellen Schnittstelle sondern mit einem Netzwerk, wie dem Ethernet, verbunden. Diese Terminals sind auch nicht auf den Textmodus beschränkt, sondern können jede X-Anwendung darstellen.

X-Terminals sind hier nur der Vollständigkeit halber aufgezählt. Die Einrichtung von X-Terminals wird in diesem Abschnitt *nicht* beschrieben.

27.3.2. Konfiguration

Im Folgenden wird beschrieben, wie Sie Ihr FreeBSD-System konfigurieren müssen, um sich an einem Terminal anzumelden. Dabei wird vorausgesetzt, dass der Kernel bereits die serielle Schnittstelle, die mit dem Terminal verbunden ist, unterstützt. Weiterhin sollte der Terminal schon angeschlossen sein.

Aus Kapitel 13 wissen Sie, dass `init` für das Initialisieren des Systems und den Start von Prozessen zum Zeitpunkt des Systemstarts verantwortlich ist. Unter anderem liest `init /etc/ttys` ein und startet für jeden verfügbaren Terminal einen `getty` Prozess. `getty` wiederum fragt beim Anmelden den Benutzernamen ab und startet `login`.

Um Terminals auf Ihrem FreeBSD-System einzurichten, führen Sie folgenden Schritte als `root` durch:

1. Wenn er noch nicht da ist, fügen Sie einen Eintrag in `/etc/ttys` für die serielle Schnittstelle aus `/dev` ein.
2. Geben Sie `/usr/libexec/getty` als auszuführendes Programm an. Als Parameter für `getty` geben Sie den passenden Verbindungstyp aus `/etc/gettytab` an.
3. Geben Sie den Terminaltyp an.
4. Aktivieren Sie den Anschluss.
5. Geben Sie die Sicherheit des Anschlusses an.
6. Veranlassen Sie `init /etc/ttys` erneut zu lesen.

Optional können Sie in `/etc/gettytab` auch einen auf Ihre Zwecke angepassten Terminaltyp erstellen. Die genaue Vorgehensweise wird in diesem Abschnitt nicht erklärt, aber die Manualpages von gettytab(5) und getty(8) enthalten dazu weitere Informationen.

27.3.2.1. Hinzufügen eines Eintrags in `/etc/ttys`

In `/etc/ttys` werden alle Terminals aufgeführt, an denen Sie sich auf dem FreeBSD-System anmelden können. Hier findet sich zum Beispiel ein Eintrag für die erste virtuelle Konsole `/dev/ttyv0`, der es Ihnen ermöglicht, sich dort anzumelden. Die Datei enthält des Weiteren Einträge für andere virtuelle Konsolen, serielle Schnittstellen und

Pseudoterminals. Wenn Sie einen Terminal konfigurieren wollen, fügen sie einen Eintrag für den Namen des Gerätes aus `/dev` ohne das Präfix `/dev` hinzu. Zum Beispiel wird `/dev/ttyv0` als `ttyv0` aufgeführt.

In der Voreinstellung enthält `/etc/ttys` Einträge für die ersten vier seriellen Schnittstellen: `ttyu0` bis `ttyu3`. Wenn Sie an eine von diesen einen Terminal anschließen, brauchen Sie keinen weiteren Eintrag hinzuzufügen.

Beispiel 27-1. Einträge in `/etc/ttys` hinzufügen

Angenommen, wir wollen an ein System zwei Terminals anschließen: Einen Wyse-50 und einen alten 286 IBM PC, der mit **Procomm** einen VT-100 Terminal emuliert. Den Wyse-Terminal verbinden wir mit der zweiten seriellen Schnittstelle und den 286 mit der sechsten seriellen Schnittstelle (einem Anschluss auf einer Multiportkarte). Die entsprechenden Einträge in `/etc/ttys` würden dann wie folgt aussehen:

```
ttyu1❶  "/usr/libexec/getty std.38400"❷  wy50❸  on❹  insecure❺
ttyu5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

❶ Das erste Feld gibt normalerweise den Namen der Gerätedatei aus `/dev` an.

❷ Im zweiten Feld wird das auszuführende Kommando, normal ist das getty(8), angegeben. `getty` initialisiert und öffnet die Verbindung, setzt die Geschwindigkeit und fragt den Benutzernamen ab. Danach führt es login(1) aus.

`getty` akzeptiert einen optionalen Parameter auf der Kommandozeile, den Verbindungstyp, der die Eigenschaften der Verbindung, wie die Geschwindigkeit und Parität, festlegt. Die Typen und die damit verbundenen Eigenschaften liest `getty` aus `/etc/gettytab`.

`/etc/gettytab` enthält viele Einträge sowohl für neue wie auch alte Terminalverbindungen. Die meisten Einträge, die mit `std` beginnen, sollten mit einem festverdrahteten Terminal funktionieren. Für jede Geschwindigkeit zwischen 110 bps und 115200 bps gibt es einen `std` Eintrag. Natürlich können Sie auch eigene Einträge erstellen, Informationen dazu finden Sie in gettytab(5).

Wenn Sie den Verbindungstyp in `/etc/ttys` eintragen, stellen Sie bitte sicher, dass die Kommunikationseinstellungen auch mit denen des Terminals übereinstimmen.

In unserem Beispiel verwendet der Wyse-50 keine Parität und 38400 bps, der 286 PC benutzt ebenfalls keine Parität und arbeitet mit 19200 bps.

❸ Das dritte Feld gibt den Terminaltyp an, der normalerweise mit diesem Anschluss verbunden ist. Für Einwählverbindungen wird oft `unknown` oder `dialup` benutzt, da sich die Benutzer praktisch mit beliebigen Terminals oder Emulatoren anmelden können. Bei festverdrahteten Terminals ändert sich der Typ nicht, so dass Sie in diesem Feld einen richtigen Typ aus der termcap(5) Datenbank angeben können.

In unserem Beispiel benutzen wir für den Wyse-50 den entsprechenden Typ aus termcap(5), der 286 PC wird als VT-100, den er ja emuliert, angegeben.

❹ Das vierte Feld gibt an, ob der Anschluss aktiviert werden soll. Wenn Sie hier `on` angeben, startet `init` das Programm, das im zweiten Feld angegeben wurde (normal `getty`). Wenn Sie `off` angeben, wird das Kommando aus dem zweiten Feld nicht ausgeführt und folglich können Sie sich dann an dem betreffenden Terminal nicht anmelden.

❺ Im letzten Feld geben Sie die Sicherheit des Anschlusses an. Wenn Sie hier `secure` angeben, darf sich `root` (oder jeder Account mit der UID `0`) über diese Verbindung anmelden. Wenn Sie `insecure` angeben, dürfen sich nur unprivilegierte Benutzer anmelden. Diese können später mit su(1) oder einem ähnlichen Mechanismus zu `root` wechseln.

Es wird dringend empfohlen, `insecure` nur für Terminals hinter verschlossenen Türen zu verwenden, da Sie mit `su` leicht zum Superuser werden können.

27.3.2.2. `init` zwingen, `/etc/ttys` erneut zu lesen

Nachdem Sie die nötigen Änderungen in `/etc/ttys` vorgenommen haben, schicken Sie `init` ein SIGHUP-Signal (hangup), um es zu veranlassen, seine Konfigurationsdatei neu zu lesen:

```
# kill -HUP 1
```

Anmerkung: Da `init` immer der erste Prozess auf einem System ist, besitzt es immer die PID `1`.

Wenn alles richtig eingerichtet ist, alle Kabel angeschlossen und die Terminals eingeschaltet sind, sollte für jeden Terminal ein `getty` Prozess laufen und auf jedem Terminal sollten Sie eine Anmeldeaufforderung sehen.

27.3.3. Fehlersuche

Selbst wenn Sie den Anweisungen akribisch gefolgt sind, kann es immer noch zu Fehlern beim Einrichten eines Terminals kommen. Die folgende Aufzählung von Symptomen beschreibt mögliche Lösungen:

27.3.3.1. Es erscheint kein Anmeldeprompt

Stellen Sie sicher, dass der Terminal verbunden und eingeschaltet ist. Wenn ein PC als Terminal fungiert, überprüfen Sie, dass die Terminalemulation auf den richtigen Schnittstellen läuft.

Stellen Sie sicher, dass Sie das richtige Kabel verwenden und dass das Kabel fest mit dem Terminal und dem FreeBSD-Rechner verbunden ist.

Stellen Sie sicher, dass die Einstellungen für die Geschwindigkeit (bps) und Parität auf dem FreeBSD System und dem Terminal gleich sind. Wenn Ihr Terminal einen Bildschirm besitzt, überprüfen Sie die richtige Einstellung von Helligkeit und Kontrast. Wenn Ihr Terminal druckt, stellen Sie die ausreichende Versorgung mit Papier und Tinte sicher.

Überprüfen Sie mit `ps`, dass der `getty` Prozess für den Terminal läuft:

```
# ps -axww|grep getty
```

Für jeden Terminal sollten Sie einen Eintrag sehen. Aus dem folgenden Beispiel erkennen Sie, dass `getty` auf der zweiten seriellen Schnittstelle `tyyd1` läuft und den Verbindungstyp `std.38400` aus `/etc/gettytab` benutzt:

```
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyu1
```

Wenn `getty` nicht läuft, überprüfen Sie, ob der Anschluss in `/etc/ttys` aktiviert ist. Haben Sie `kill -HUP 1` abgesetzt, nachdem Sie `/etc/ttys` geändert hatten?

Wenn `getty` läuft, aber der Terminal immer noch kein Anmeldeprompt ausgibt, oder Sie am Anmeldeprompt nichts eingeben können, kann es sein, dass Ihr Terminal oder Kabel keinen Hardware-Handshake unterstützt. Ändern Sie

dann den Eintrag `std.38400` in `/etc/ttys` zu `3wire.38400`. Nachdem Sie `/etc/ttys` geändert haben, setzen Sie das Kommando `kill -HUP 1` ab. Der Eintrag `3wire` besitzt ähnliche Eigenschaften wie der Eintrag `std`, ignoriert aber den Hardware-Handshake. Wenn Sie den Eintrag `3wire` verwenden, müssen Sie vielleicht die Geschwindigkeit verkleinern oder die Software-Flusssteuerung aktivieren, um Pufferüberläufe zu vermeiden.

27.3.3.2. Es erscheinen nur unverständliche Zeichen

Stellen Sie sicher, dass die Einstellungen für die Geschwindigkeit (bps) und Parität auf dem FreeBSD System und dem Terminal gleich sind. Kontrollieren Sie den `getty` Prozess und stellen Sie sicher, dass der richtige Verbindungstyp aus `/etc/gettytab` benutzt wird. Wenn das nicht der Fall ist, editieren Sie `/etc/ttys` und setzen das Kommando `kill-HUP 1` ab.

27.3.3.3. Zeichen erscheinen doppelt und eingegebene Passwörter erscheinen im Klartext

Stellen Sie den Terminal oder die Terminalemulation von “half duplex” oder “local echo” auf “full duplex.” um.

27.4. Einwählverbindungen

Beigetragen von Guy Helmer. Mit Anmerkungen von Sean Kelly.

Warnung: Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

Das Einrichten von Einwählverbindungen ähnelt dem Anschließen von Terminals, nur dass Sie anstelle eines Terminals ein Modem verwenden.

27.4.1. Externe und interne Modems

Externe Modems sind für Einwählverbindungen besser geeignet, da sie die Konfiguration in nicht flüchtigem RAM speichern können. Zudem verfügen Sie über Leuchtanzeigen, die den Status wichtiger RS-232 Signale anzeigen und unter Umständen Besucher beeindrucken können.

Interne Modems verfügen normalerweise nicht über nicht flüchtiges RAM und lassen sich meist nur über DIP-Schalter konfigurieren. Selbst wenn ein internes Modem Leuchtanzeigen besitzt, sind diese meist schwer einzusehen, wenn das Modem eingebaut ist.

27.4.1.1. Modems und Kabel

Mit einem externen Modem müssen Sie das richtige Kabel benutzen: Ein Standard RS-232C Kabel, bei dem die folgenden Signale miteinander verbunden sind, sollte ausreichen:

Tabelle 27-4. Signalnamen

Abkürzung	Bedeutung
RD	Received Data
TD	Transmitted Data
DTR	Data Terminal Ready
DSR	Data Set Ready
DCD	Data Carrier Detect (dadurch erkennt RS-232 das Signal *Received Line*)
SG	Signal Ground
RTS	Request to Send
CTS	Clear to Send

Ab Geschwindigkeiten von 2400 bps benötigt FreeBSD die Signale RTS und CTS für die Flusssteuerung. Das Signal CD zeigt an, ob ein Träger vorliegt, das heißt ob die Verbindung aufgebaut ist oder beendet wurde. DTR zeigt an, dass das Gerät betriebsbereit ist. Es gibt einige Kabel, bei denen nicht alle nötigen Signale verbunden sind. Wenn Sie Probleme der Art haben, dass zum Beispiel die Sitzung nicht beendet wird, obwohl die Verbindung beendet wurde, kann das an einem solchen Kabel liegen.

Wie andere UNIX Betriebssysteme auch, benutzt FreeBSD Hardwaresignale, um festzustellen, ob ein Anruf beantwortet wurde, eine Verbindung beendet wurde, oder um die Verbindung zu schließen und das Modem zurückzusetzen. FreeBSD vermeidet es, dem Modem Kommandos zu senden, oder den Statusreport des Modems abzufragen. Falls Sie ein Benutzer von PC-basierenden Bulletin Board Systemen sind, mag Sie das verwundern.

27.4.2. Schnittstellenbausteine

FreeBSD unterstützt EIA RS-232C (CCITT V.24) serielle Schnittstellen, die auf den NS8250, NS16450, NS16550 oder NS16550A Bausteinen basieren. Die Bausteine der Serie 16550 verfügen über einen 16 Byte großen Puffer, der als FIFO angelegt ist. Wegen Fehler in der FIFO-Logik kann der Puffer in einem 16550 Baustein allerdings nicht genutzt werden, das heißt der Baustein muss als 16450 betrieben werden. Bei allen Bausteinen ohne Puffer und dem 16550 Baustein muss jedes Byte einzeln von dem Betriebssystem verarbeitet werden, was Fehler bei hohen Geschwindigkeiten oder großer Systemlast erzeugt. Es sollten daher nach Möglichkeit serielle Schnittstellen, die auf 16550A Bausteinen basieren, eingesetzt werden.

27.4.3. Überblick

Wie bei Terminals auch, startet `init` für jede serielle Schnittstelle, die eine Einwählverbindung zur Verfügung stellt, einen `getty` Prozess. Wenn das Modem beispielsweise an `/dev/ttyu0` angeschlossen ist, sollte in der Ausgabe von `ps ax` eine Zeile wie die folgende erscheinen:

```
 4850 ??  I      0:00.09 /usr/libexec/getty V19200 ttyu0
```

Wenn sich ein Benutzer einwählt und die Verbindung aufgebaut ist, zeigt das Modem dies durch das CD Signal (Carrier Detect) an. Der Kernel merkt, dass ein Signal anliegt und vollendet das Öffnen der Schnittstelle durch `getty`. Dann sendet `getty` das Anmeldeprompt mit der ersten für die Verbindung vereinbarten Geschwindigkeit und wartet auf eine Antwort. Wenn die Antwort unverständlich ist, weil zum Beispiel die Geschwindigkeit des Modems von `getty`s Geschwindigkeit abweicht, versucht `getty` die Geschwindigkeit solange anzupassen, bis es eine verständliche Antwort erhält.

`getty` führt, nachdem der Benutzer seinen Namen eingegeben hat, `/usr/bin/login` aus, welches das Passwort abfragt und danach die Shell des Benutzers startet.

27.4.4. Konfigurationsdateien

Drei Konfigurationsdateien in `/etc` steuern, ob eine Einwahl in Ihr FreeBSD-System möglich ist. Die erste, `/etc/gettytab`, konfiguriert den `/usr/libexec/getty` Dæmon. In `/etc/ttys` wird festgelegt, auf welchen Schnittstellen `/sbin/init` einen `getty` Prozess startet. Schließlich haben Sie in `/etc/rc.d/serial` die Möglichkeit, Schnittstellen zu initialisieren.

Es gibt zwei Ansichten darüber, wie Modems für Einwählverbindungen unter UNIX zu konfigurieren sind. Zum einen kann die Geschwindigkeit zwischen dem Modem und dem Computer fest eingestellt werden. Sie ist damit unabhängig von der Geschwindigkeit, mit der sich der entfernte Benutzer einwählt. Dies hat den Vorteil, dass der entfernte Benutzer das Anmeldeprompt sofort bekommt. Der Nachteil bei diesem Verfahren ist, dass das System die tatsächliche Geschwindigkeit der Verbindung nicht kennt. Damit können bildschirmorientierte Programme wie **Emacs** ihren Bildschirmaufbau nicht an langsame Verbindungen anpassen, um die Antwortzeiten zu verbessern.

Die andere Möglichkeit besteht darin, die Geschwindigkeit der RS-232 Schnittstelle des lokalen Modems an die Geschwindigkeit des entfernten Modems anzupassen. Bei einer V.32bis (14400 bps) Verbindung kann das lokale Modem die RS-232 Schnittstelle mit 19200 bps betreiben, während bei einer Verbindung mit 2400 bps die RS-232 Schnittstelle mit 2400 bps betrieben wird. Da `getty` die Verbindungsgeschwindigkeit des Modems nicht kennt, startet es den Anmeldevorgang mit der Ausgabe von `login:` und wartet auf eine Antwort. Wenn der Benutzer der Gegenstelle nun nur unverständliche Zeichen erhält, muss er solange **Enter** drücken, bis das Anmeldeprompt erscheint. Solange die Geschwindigkeiten nicht übereinstimmen, sind die Antworten der Gegenstelle für `getty` ebenfalls unverständlich. In diesem Fall wechselt `getty` zur nächsten Geschwindigkeit und gibt wieder `login:` aus. In aller Regel erhält der Benutzer der Gegenstelle nach ein bis zwei Tastendrücken eine erkennbare Anmeldeaufforderung. Diese Anmeldeprozedur sieht nicht so sauber wie die Methode mit einer festen Geschwindigkeit aus, bietet dem Benutzer einer langsamen Verbindung allerdings den Vorteil, dass sich bildschirmorientierte Programme an die Geschwindigkeit anpassen können.

Im Folgenden wird die Konfiguration für beide Methoden besprochen, doch die Methode der angepassten Geschwindigkeit wird bei der Diskussion bevorzugt.

27.4.4.1. `/etc/gettytab`

Mit `/etc/gettytab` wird getty(8) im Stil von termcap(5) konfiguriert. Das Format dieser Datei und die Bedeutung der Einträge wird in gettytab(5) beschrieben.

27.4.4.1.1. Konfiguration für feste Geschwindigkeit

Wenn Sie die Modemgeschwindigkeit vorgeben, werden Sie in `/etc/gettytab` nichts ändern müssen.

27.4.4.1.2. Konfiguration für angepasste Geschwindigkeit

In `/etc/gettytab` müssen Einträge für die Geschwindigkeiten, die Sie benutzen wollen, sein. Wenn Sie ein 2400 bps Modem besitzen, können Sie wahrscheinlich den schon vorhandenen `D2400` Eintrag benutzen.

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
```

```
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

Wenn Sie ein Modem mit einer höheren Geschwindigkeit besitzen, müssen Sie wahrscheinlich in `/etc/gettytab` weitere Einträge erstellen. Hier ist ein Beispiel, das Sie mit einem 14400 bps Modem benutzen können:

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

Die damit erzeugten Verbindungen verwenden 8 Bit und keine Parität.

Im obigen Beispiel startet die Geschwindigkeit bei 19200 bps (eine V.32bis Verbindung) und geht dann über 9600 bps (V.32), 400 bps, 1200 bps und 300 bps wieder zurück zu 19200 bps. Das Schlüsselwort `nx=` (*next table*) sorgt für das zyklische Durchlaufen der Geschwindigkeiten. Jede Zeile zieht zudem noch mit `tc=` (*table continuation*) die Vorgabewerte für die jeweilige Geschwindigkeit an.

Wenn Sie ein 28800 bps Modem besitzen und/oder Kompression mit einem 14400 bps Modem benutzen wollen, brauchen Sie höhere Geschwindigkeiten als 19200 bps. Das folgende Beispiel startet mit 57600 bps:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57600 bps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

> **Anmerkung:** Wenn Sie eine langsame CPU oder ein stark ausgelastetes System besitzen und sich kein 16550A in Ihrem System befindet, erhalten Sie bei 57600 bps vielleicht `sio` Fehlermeldungen der Form “silo overflow”.

27.4.4.2. `/etc/ttys`

`/etc/ttys` wurde bereits in Beispiel 27-1 besprochen. Die Konfiguration für Modems ist ähnlich, allerdings braucht `getty` ein anderes Argument und es muss ein anderer Terminaltyp angegeben werden. Der Eintrag für beide Methoden (feste und angepasste Geschwindigkeit) hat die folgende Form:

```
ttyu0   "/usr/libexec/getty xxx"   dialup on
```

Das erste Feld der obigen Zeile gibt die Gerätedatei für diesen Eintrag an – `ttyu0` bedeutet, dass `getty` mit `/dev/ttyu0` arbeitet. Das zweite Feld `"/usr/libexec/getty xxx"` gibt das Kommando an, das `init` für dieses Gerät startet (*xxx* wird durch einen passenden Eintrag aus `/etc/gettytab` ersetzt). Die Vorgabe für den Terminaltyp, hier `dialup`, wird im dritten Feld angegeben. Das vierte Feld, `on`, zeigt `init` an, dass die Schnittstelle aktiviert ist. Im fünften Feld könnte noch `secure` angegeben werden, um Anmeldungen von `root` zu erlauben, doch sollte das wirklich nur für physikalisch sichere Terminals, wie die Systemkonsole, aktiviert werden.

Die Vorgabe für den Terminaltyp, `dialup` im obigen Beispiel, hängt von lokalen Gegebenheiten ab. Traditionell wird `dialup` für Einwählverbindungen verwendet, so dass die Benutzer in ihren Anmeldeskripten den Terminaltyp auf ihren Terminal abstimmen können, wenn der Typ auf `dialup` gesetzt ist. Wenn Sie aber beispielsweise nur VT102 Terminals oder Emulatoren einsetzen, können Sie den Terminaltyp hier auch fest auf `vt102` setzen.

Nachdem Sie `/etc/ttys` geändert haben, müssen Sie `init` ein HUP Signal schicken, damit es die Datei wieder einliest. Sie können dazu das folgende Kommando verwenden:

```
# kill -HUP 1
```

Wenn Sie das System zum ersten Mal konfigurieren, sollten Sie dieses Kommando erst ausführen, wenn Sie Ihr Modem richtig konfiguriert und angeschlossen haben.

27.4.4.2.1. Konfiguration für feste Geschwindigkeit

Das Argument von `getty` muss in diesem Fall eine feste Geschwindigkeit vorgeben. Der Eintrag für ein Modem, das fest auf 19200 bps eingestellt ist, könnte wie folgt aussehen:

```
ttyu0   "/usr/libexec/getty std.19200"   dialup on
```

Wenn Ihr Modem auf eine andere Geschwindigkeit eingestellt ist, setzen Sie anstelle von `std.19200` einen passenden Eintrag der Form `std.`*speed* ein. Stellen Sie sicher, dass dies auch ein gültiger Verbindungstyp aus `/etc/gettytab` ist.

27.4.4.2.2. Konfiguration für angepasste Geschwindigkeit

Das Argument von `getty` muss hier auf einen der Einträge aus `/etc/gettytab` zeigen, der zu einer Kette von Einträgen gehört, die die zu probierenden Geschwindigkeiten beschreiben. Wenn Sie dem obigen Beispiel gefolgt sind und zusätzliche Einträge in `/etc/gettytab` erzeugt haben, können Sie die folgende Zeile verwenden:

```
ttyu0   "/usr/libexec/getty V19200"   dialup on
```

27.4.4.3. `/etc/rc.d/serial`

Modems, die höhere Geschwindigkeiten unterstützen, zum Beispiel V.32, V.32bis und V.34 Modems, benutzen Hardware-Flusssteuerung (`RTS/CTS`). Für die entsprechenden Schnittstellen können Sie die Flusssteuerung mit `stty` in `/etc/rc.d/serial` einstellen.

Um beispielsweise die Hardware-Flusssteuerung für die Geräte zur Ein- und Auswahl der zweiten seriellen Schnittstelle (`COM2`) zu aktivieren, benutzen Sie die Dateien zur Initialisierung der entsprechenden Geräte und fügen die folgenden Zeilen in `/etc/rc.d/serial` hinzu:

```
# Serial port initial configuration
stty -f /dev/ttyu1.init crtscts
stty -f /dev/cuad1.init crtscts
```

27.4.5. Modemkonfiguration

Wenn Sie ein Modem besitzen, das seine Konfiguration in nicht flüchtigem RAM speichert, werden Sie ein Terminalprogramm wie **Telix** unter MS-DOS oder `tip` unter FreeBSD benötigen, um die Parameter einzustellen. Verbinden Sie sich mit derselben Geschwindigkeit, die `getty` zuerst benutzen würde, mit dem Modem und treffen Sie folgende Einstellungen:

- DCD ist eingeschaltet, wenn das Trägersignal des entfernten Modems erkannt wird.
- Im Betrieb liegt DTR an. Bei einem Verlust von DTR legt das Modem auf und setzt sich zurück.
- CTS Flusssteuerung ist für ausgehende Daten aktiviert.
- XON/XOFF Flusssteuerung ist ausgeschaltet.
- RTS Flusssteuerung ist für eingehende Daten aktiviert.
- Keine Rückmeldungen ausgeben.
- Die Echo-Funktion ist deaktiviert.

In der Dokumentation Ihres Modems finden Sie die nötigen Befehle, die Sie absetzen müssen, und/oder nötigen DIP-Schalterstellungen, um die obigen Einstellungen zu treffen.

Für ein externes 14400 U.S. Robotics® Sportster® gelten zum Beispiel die folgenden Befehle:

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

Bei dieser Gelegenheit können Sie auch gleich andere Einstellungen, zum Beispiel ob Sie V42.bis und/oder MNP5 Kompression benutzen wollen, an Ihrem Modem vornehmen.

Bei einem externen 14400 U.S. Robotics Sportster müssen Sie auch noch einige DIP-Schalter einstellen. Die folgenden Einstellungen können Sie vielleicht als Beispiel für andere Modems verwenden:

- Schalter 1: OBEN – DTR normal
- Schalter 2: N/A (Rückmeldungen als Text/numerische Rückmeldungen)
- Schalter 3: OBEN – Keine Rückmeldungen ausgeben

- Schalter 4: UNTEN – Echo-Funktion aus
- Schalter 5: OBEN – Rufannahme aktiviert
- Schalter 6: OBEN – Carrier Detect normal
- Schalter 7: OBEN – Einstellungen aus dem NVRAM laden
- Schalter 8: N/A (Smart Mode/Dumb Mode)

Für Einwählverbindungen sollten die Rückmeldungen deaktiviert sein, da sonst `getty` dem Modem das Anmeldeprompt `login:` schickt und das Modem im Kommandomodus das Prompt wieder ausgibt (Echo-Funktion) oder eine Rückmeldung gibt. Das führt dann zu einer länglichen und fruchtlosen Kommunikation zwischen dem Modem und `getty`.

27.4.5.1. Konfiguration für feste Geschwindigkeit

Die Geschwindigkeit zwischen Modem und Computer muss auf einen festen Wert eingestellt werden. Mit einem externen 14400 U.S. Robotics Sportster Modem setzen die folgenden Kommandos die Geschwindigkeit auf den Wert der Datenendeinrichtung fest:

```
ATZ
AT&B1&W
```

27.4.5.2. Konfiguration für angepasste Geschwindigkeit

In diesem Fall muss die Geschwindigkeit der seriellen Schnittstelle des Modems der eingehenden Geschwindigkeit angepasst werden. Für ein externes 14400 U.S. Robotics Sportster Modem erlauben die folgenden Befehle eine Anpassung der Geschwindigkeit der seriellen Schnittstelle für Verbindungen, die keine Fehlerkorrektur verwenden:

```
ATZ
AT&B2&W
```

Verbindungen mit Fehlerkorrektur (V.42, MNP) verwenden die Geschwindigkeit der Datenendeinrichtung.

27.4.5.3. Überprüfen der Modemkonfiguration

Die meisten Modems verfügen über Kommandos, die die Konfiguration des Modems in lesbarer Form ausgeben. Auf einem externen 14400 U.S. Robotics Sportster zeigt `ATI5` die Einstellungen im nicht flüchtigen RAM an. Um die wirklichen Einstellungen unter Berücksichtigung der DIP-Schalter zu sehen, benutzen Sie `ATZ` gefolgt von `ATI4`.

Wenn Sie ein anderes Modem benutzen, schauen Sie bitte in der Dokumentation Ihres Modems nach, wie Sie die Konfiguration des Modems überprüfen können.

27.4.6. Fehlersuche

Bei Problemen können Sie die Einwählverbindung anhand der folgenden Punkte überprüfen:

27.4.6.1. Überprüfen des FreeBSD-Systems

Schließen Sie das Modem an das FreeBSD-System an und booten Sie das System. Wenn Ihr Modem über Statusindikatoren verfügt, überprüfen Sie, ob der DTR Indikator leuchtet, wenn das Anmeldeprompt erscheint. Dies zeigt an, dass das FreeBSD-System einen `getty` Prozess auf der entsprechenden Schnittstelle gestartet hat und das Modem auf einkommende Verbindungen wartet.

Wenn der DTR-Indikator nicht leuchtet, melden Sie sich an dem FreeBSD-System an und überprüfen mit `ps ax`, ob FreeBSD einen `getty`-Prozess auf der entsprechenden Schnittstelle gestartet hat. Unter den angezeigten Prozessen sollten Sie ähnliche wie die folgenden finden:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu0
115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu1
```

Wenn das Modem noch keinen Anruf entgegengenommen hat und Sie stattdessen die folgende Zeile sehen

```
114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyu0
```

bedeutet dies, dass `getty` die Schnittstelle schon geöffnet hat und zeigt Kabelprobleme oder eine falsche Modemkonfiguration an, da `getty` die Schnittstelle erst dann öffnen kann, wenn das CD Signal (Carrier Detect) vom Modem anliegt.

Wenn Sie keine `getty`-Prozesse auf den gewünschten `ttyu`*N* Ports finden, untersuchen Sie bitte `/etc/ttys` auf Fehler. Suchen Sie auch in `/var/log/messages` nach Meldungen von `init` oder `getty`. Wenn Sie dort Meldungen finden, sollten Sie noch einmal die beiden Konfigurationsdateien `/etc/ttys` und `/etc/gettytab` nach Fehlern durchsehen. Überprüfen Sie auch, ob die Gerätedateien `/dev/ttyu`*N* vorhanden sind.

27.4.6.2. Einwählversuch

Versuchen Sie, sich in Ihr System einzuwählen. Auf dem entfernten System stellen Sie bitte die folgenden Kommunikationsparameter ein: 8 Bit, keine Parität, ein Stop-Bit. Wenn Sie kein Anmeldeprompt erhalten oder nur unleserliche Zeichen sehen, drücken Sie mehrmals, in Abständen von ungefähr einer Sekunde, **Enter**. Wenn Sie immer noch nicht die `login:` Meldung sehen, schicken Sie ein `BREAK` Kommando. Wenn Sie zur Einwahl ein Highspeed-Modem benutzen, verwenden Sie eine feste Geschwindigkeit auf der seriellen Schnittstelle des Modems (`AT&B1` für ein U.S. Robotics Sportster).

Wenn Sie jetzt immer noch kein Anmeldeprompt erhalten, überprüfen Sie nochmals `/etc/gettytab` und stellen sicher, dass

- der Verbindungstyp in `/etc/ttys` zu einem gültigen Eintrag in `/etc/gettytab` gehört,
- jeder der `nx=` Einträge in `gettytab` gültig ist und
- jeder `tc=` Eintrag auf einen gültigen Eintrag in `gettytab` verweist.

Wenn das Modem an Ihrem FreeBSD-System auf einen eingehenden Anruf nicht antwortet, stellen Sie sicher, dass das Modem so konfiguriert ist, dass es einen Anruf beantwortet, wenn DTR anliegt. Wenn Ihr Modem Statusindikatoren besitzt, können Sie das Anliegen von DTR anhand der Leuchten überprüfen.

Wenn Sie alles schon mehrfach überprüft haben und es immer noch noch nicht funktioniert, machen Sie erst einmal eine Pause, bevor Sie weitermachen. Wenn es immer noch nicht funktioniert, können Sie eine Mail an die Mailingliste 'Fragen und Antworten zu FreeBSD' <`de-bsd-questions@de.FreeBSD.org`> schicken, in der Sie Ihr Modem und Ihr Problem beschreiben und Ihnen sollte geholfen werden.

27.5. Verbindungen nach Außen

> **Warnung:** Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

Die folgenden Ratschläge beschreiben, wie Sie mit einem Modem eine Verbindung zu einem anderen Computer herstellen. Dies können Sie nutzen, um sich auf einem entfernten Computer anzumelden, oder um eine Verbindung zu einem BBS (Bulletin Board System) herzustellen.

Weiterhin ist diese Art von Verbindungen nützlich, wenn mal Ihr PPP nicht funktioniert. Wenn Sie zum Beispiel eine Datei mit FTP übertragen wollen und das über PPP gerade nicht möglich ist, melden Sie sich auf dem entfernten Rechner an und führen dort die FTP-Sitzung durch. Die Dateien können Sie danach mit zmodem auf den lokalen Rechner übertragen.

27.5.1. Mein Hayes Modem wird nicht unterstützt – was kann ich tun?

Eigentlich ist die Onlinehilfe für `tip` nicht mehr aktuell. Es gibt einen eingebauten, allgemeinen Hayes Wähler. Verwenden Sie einfach `at=hayes` in `/etc/remote`.

Der Hayes-Treiber ist nicht schlau genug, um ein paar der erweiterten Funktionen von neueren Modems zu erkennen – Nachrichten wie `BUSY`, `NO DIALTONE` oder `CONNECT 115200` verwirren ihn nur. Sie sollten diese Nachrichten mit Hilfe von `ATX0&W` abschalten, wenn Sie `tip` benutzen.

Der Anwahl-Timeout von `tip` beträgt 60 Sekunden. Ihr Modem sollte weniger verwenden, oder `tip` denkt, dass ein Kommunikationsfehler vorliegt. Versuchen Sie es mit `ATS7=45&W`.

27.5.2. Wie soll ich die AT-Befehle eingeben?

Erstellen Sie einen so genannten `direct` Eintrag in `/etc/remote`. Wenn Ihr Modem zum Beispiel an der ersten seriellen Schnittstelle, `/dev/cuad0`, angeschlossen ist, dann fügen Sie die folgende Zeile hinzu:

```
cuad0:dv=/dev/cuad0:br#19200:pa=none
```

Verwenden Sie die höchste bps-Rate, die Ihr Modem in der br Fähigkeit unterstützt. Geben Sie dann `tip cuad0` ein und Sie sind mit Ihrem Modem verbunden.

Oder benutzen Sie `cu` als `root` mit dem folgenden Befehl:

```
# cu -llline -sspeed
```

`line` steht für die serielle Schnittstelle (`/dev/cuad0`) und *`speed`* für die Geschwindigkeit (`57600`). Wenn Sie mit dem Eingeben der AT Befehle fertig sind, beenden Sie mit `~.`.

27.5.3. Wieso funktioniert das @ Zeichen für die pn Fähigkeit nicht?

Das `@` Zeichen in der Telefonnummerfähigkeit sagt `tip`, dass es in der Datei `/etc/phones` nach einer Nummer suchen soll. Aber `@` ist auch ein spezielles Zeichen in den Dateien, in denen Fähigkeiten beschrieben werden, wie `/etc/remote`. Schreiben Sie es mit einem Backslash:

```
pn=\@
```

27.5.4. Wie kann ich von der Kommandozeile eine Telefonnummer wählen?

Stellen Sie einen allgemeinen Eintrag in `/etc/remote`. Zum Beispiel:

```
tip115200|Dial any phone number at 115200 bps:\
        :dv=/dev/cuad0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
        :dv=/dev/cuad0:br#57600:at=hayes:pa=none:du:
```

Mit dem folgenden Befehl können Sie dann wählen:

```
# tip -115200 5551234
```

Sollten Sie `cu` gegenüber `tip` bevorzugen, verwenden Sie einen allgemeinen cu-Eintrag:

```
cu115200|Use cu to dial any number at 115200bps:\
        :dv=/dev/cuad1:br#57600:at=hayes:pa=none:du:
```

und benutzen zum Wählen das Kommando:

```
# cu 5551234 -s 115200
```

27.5.5. Muss ich dabei jedes Mal die bps-Rate angeben?

Schreiben Sie einen `tip1200`- oder einen `cu1200`-Eintrag, aber geben Sie auch die bps-Rate an, die Ihr Modem wirklich unterstützt. Leider denkt tip(1), dass 1200 bps ein guter Standardwert ist und deswegen sucht es nach einem `tip1200`-Eintrag. Natürlich müssen Sie nicht 1200 bps benutzen.

27.5.6. Wie kann ich möglichst komfortabel über einen Terminal-Server auf verschiedene Rechner zugreifen?

Sie müssen nicht warten bis Sie verbunden sind, und jedes Mal `CONNECT` *`Rechner`* eingeben, benutzen Sie `tips` `cm`-Fähigkeit. Sie können diese Einträge in `/etc/remote` verwenden:

```
pain|pain.deep13.com|Forrester's machine:\
        :cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        :dv=/dev/cuad2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

Mit den Befehlen `tip pain` oder `tip muffin` können Sie eine Verbindungen zu den Rechnern `pain` oder `muffin` herstellen; mit `tip deep13` verbinden Sie sich mit dem Terminalserver.

27.5.7. Kann `tip` mehr als eine Verbindung für jede Seite ausprobieren?

Das ist oft ein Problem, wenn eine Universität mehrere Telefonleitungen hat und viele tausend Studenten diese benutzen wollen.

Erstellen Sie einen Eintrag für Ihre Universität in `/etc/remote` und benutzen Sie `@` für die `pn`-Fähigkeit:

```
big-university:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuad3:br#9600:at=courier:du:pa=none:
```

Listen Sie die Telefonnummern der Universität in `/etc/phones` auf:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` probiert jede der Nummern in der aufgelisteten Reihenfolge und gibt dann auf. Möchten Sie, dass `tip` beim Versuchen eine Verbindung herzustellen nicht aufgibt, lassen Sie es in einer while-Schleife laufen.

27.5.8. Warum muss ich zweimal Ctrl+P tippen, um ein Ctrl+P zu senden?

Ctrl+**P** ist das voreingestellte Zeichen, mit dem eine Übertragung erzwungen werden kann und wird benutzt, um `tip` zu sagen, dass das nächste Zeichen direkt gesendet werden soll und nicht als Fluchtzeichen interpretiert werden soll. Mit Hilfe der Fluchtsequenz `~s`, mit der man Variablen setzen kann, können Sie jedes andere Zeichen als "force"-Zeichen definieren.

Geben Sie `~sforce=`*`Zeichen`* gefolgt von **Enter** ein. Für *`Zeichen`* können Sie ein beliebiges einzelnes Zeichen einsetzen. Wenn Sie *`Zeichen`* weglassen, ist das "force"-Zeichen "nul", das Sie mit **Ctrl**+**2** oder **Ctrl**+**Leertaste** eingeben können. Ein guter Wert für *`Zeichen`* ist **Shift**+**Ctrl**+**6**, welches nur auf wenigen Terminal Servern benutzt wird.

Sie können das "force"-Zeichen auch bestimmen, indem Sie in `$HOME/.tiprc` das Folgende einstellen:

```
force=single-char
```

27.5.9. Warum ist auf einmal alles was ich schreibe in GROSSBUCHSTABEN??

Sie müssen **Ctrl**+**A**, eingegeben haben, das "raise"-Zeichen von `tip`, das speziell für Leute mit defekten caps-lock Tasten eingerichtet wurde. Benutzen Sie `~s` wie oben und setzen Sie die Variable `raisechar` auf etwas, das Ihnen angemessen erscheint. Tatsächlich kann die Variable auf das gleiche Zeichen wie das "force"-Zeichen gesetzt werden, wenn Sie diese Fähigkeiten niemals benutzen wollen.

Hier ist ein Muster der `.tiprc` Datei, perfekt für **Emacs** Benutzer, die oft **Ctrl**+**2** und **Ctrl**+**A** tippen müssen:

```
force=^^
raisechar=^^
```

Geben Sie für `^^` **Shift**+**Ctrl**+**6** ein.

27.5.10. Wie kann ich Dateien mit `tip` übertragen?

Wenn Sie mit einem anderen UNIX System kommunizieren, können Sie mit `~p` (put) und `~t` (take) Dateien senden und empfangen. Diese Befehle lassen `cat` und `echo` auf dem entfernten System laufen, um Dateien zu empfangen und zu senden. Die Syntax ist:

`~p` local-file [remote-file]

`~t` remote-file [local-file]

Es gibt keine Fehlerkontrolle, deshalb sollten Sie besser ein anderes Protokoll, wie zmodem, benutzen.

27.5.11. Wie kann ich zmodem mit `tip` laufen lassen?

Um Dateien zu empfangen, starten Sie das Programm zum Senden auf dem entfernten Computer. Geben Sie dann `~C rz` ein, um die Dateien lokal zu empfangen.

Um Dateien zu senden, starten Sie das Programm zum Empfangen auf dem entfernten Computer. Geben Sie dann `~C sz` *`Dateien`* ein, um Dateien auf das entfernte System zu senden.

27.6. Einrichten der seriellen Konsole

Beigesteuert von Kazutaka YOKOTA. Auf Grundlage eines Dokuments von Bill Paul.

Warnung: Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports von `/dev/ttydN` in `/dev/ttyuN` umbenannt. Setzen Sie noch FreeBSD 7.X ein, müssen Sie die Befehle in den folgenden Abschnitten entsprechend anpassen.

27.6.1. Einführung

FreeBSD kann ein System mit einem Dumb-Terminal (unintelligente Datenstation) an einer seriellen Schnittstelle als Konsole booten. Diese Konfiguration ist besonders nützlich für Systemadministratoren, die FreeBSD auf Systemen ohne Tastatur oder Monitor installieren wollen, und Entwickler, die den Kernel oder Gerätetreiber debuggen.

Wie in Kapitel 13 beschrieben, besitzt FreeBSD drei Bootphasen. Der Code für die ersten beiden Bootphasen befindet sich im Bootsektor am Anfang der FreeBSD-Slice der Bootplatte. Dieser Bootblock lädt den Bootloader (`/boot/loader`) in Phase drei.

Um eine serielle Konsole einzurichten, müssen Sie den Bootblock, den Bootloader und den Kernel konfigurieren.

27.6.2. Serielle Konsole einrichten, Kurzfassung

Dieser Abschnitt fasst zusammen, wie Sie eine serielle Konsole einrichten. Es wird vorausgesetzt, dass Sie die Voreinstellungen verwenden und wissen, wie serielle Schnittstellen verbunden werden.

1. Verbinden Sie die serielle Konsole mit `COM1` sowie dem Kontrollterminal.
2. Um die Startmeldungen der seriellen Konsole zu sehen, geben Sie als `root` Folgendes ein:

   ```
   # echo 'console="comconsole"' >> /boot/loader.conf
   ```

3. Ändern Sie in `/etc/ttys` den Eintrag für `ttyu0` von `off` auf `on`. Zusätzlich sollten Sie den Wert `dialup` auf `vt100` ändern. Nur so wird auf der seriellen Konsole eine Eingabeaufforderung mit einer Passwortabfrage aktiviert.
4. Starten Sie nun das System neu, damit die serielle Konsole aktiviert wird.

Wenn Sie eine unterschiedliche Konfiguration benötigen, sollten Sie Abschnitt 27.6.3 lesen.

27.6.3. Konfiguration der Konsole

1. Bereiten Sie ein serielles Kabel vor.

   Sie benötigen entweder ein Nullmodemkabel oder ein serielles Standard Kabel mit einem Nullmodemkabel-Adapter. In Abschnitt 27.2.2 wurden serielle Kabel beschrieben.

2. Trennen Sie die Tastatur vom Computer.

   Die meisten PC Systeme suchen beim Power On Self Test (POST) nach einer Tastatur und geben eine Fehlermeldung aus, wenn sie keine finden. Einige Maschinen werden sich sogar weigern, ohne Tastatur zu booten.

   Wenn Ihr Rechner trotz einer Fehlermeldung normal weiterbootet, brauchen Sie weiter nichts zu tun. Beispielsweise geben einige Maschinen mit einem Phoenix BIOS nur `Keyboard failed` aus und booten dann normal weiter.

   Wenn Ihr System ohne Tastatur nicht booten will, müssen Sie das BIOS so konfigurieren, das es diesen Fehler ignoriert (wenn das möglich ist). Das Handbuch zu Ihrem Motherboard sollte beschreiben, wie das zu bewerkstelligen ist.

   > **Tipp:** Selbst wenn Sie im BIOS “Not installed” für die Tastatur einstellen, können Sie eine Tastatur angeschlossen haben und diese auch weiterhin benutzen, da sie mit dieser Anweisung das BIOS lediglich anweisen, nach dem Einschalten des Rechners nicht nach einer Tastatur zu suchen und den Rechner ohne entsprechende Fehlermeldung zu starten. Wenn die oben beschriebene Option nicht im BIOS vorhanden ist, halten Sie stattdessen Ausschau nach einer “Halt on Error” Option. Sie können den gleichen Effekt wie oben erzielen, wenn Sie diese Option auf “All but Keyboard” oder sogar “No Errors” setzen.

   > **Anmerkung:** Wenn Ihr System über eine PS/2® Maus verfügt, müssen Sie diese wahrscheinlich auch abziehen. Da sich die PS/2 Maus und die Tastatur einige Hardwarekomponenten teilen, kann das dazu führen, dass die Hardwareerkennung fälschlicherweise eine Tastatur findet, wenn eine PS/2 Maus angeschlossen ist. Gateway 2000 Pentium 90 MHz Systemen wird dieses Verhalten nachgesagt. Normalerweise ist das kein Problem, da eine Maus ohne Tastatur sowieso nicht sinnvoll einsetzbar ist.

3. Schließen Sie einen Dumb-Terminal an `COM1` (`sio0`) an.

Wenn Sie keinen Dumb-Terminal besitzen, können Sie einen alten PC/XT mit einem Terminalemulator oder die serielle Schnittstelle eines anderen UNIX Rechners benutzen. Sie benötigen auf jeden Fall eine freie erste serielle Schnittstelle (`COM1`). Zurzeit ist es nicht möglich, in den Bootblöcken eine andere Schnittstelle zu konfigurieren, ohne diese neu zu kompilieren. Wenn Sie `COM1` bereits für ein anderes Gerät benutzen, müssen Sie dieses Gerät temporär entfernen und einen neuen Bootblock sowie Kernel installieren, wenn Ihr FreeBSD erst einmal installiert ist. Auf einem Server sollte `COM1` ohnehin verfügbar sein. Wenn Sie die Schnittstelle für ein anderes Gerät benutzen und Sie dieses nicht auf `COM2` (`sio1`) legen können, sollten Sie sich nicht an erster Stelle mit dem Aufsetzen einer seriellen Konsole beschäftigen.

4. Stellen Sie sicher, dass Ihre Kernelkonfiguration die richtigen Optionen für `COM1` (`sio0`) enthält.

   Relevante Optionen sind:

   `0x10`

   > Aktiviert die Konsolenunterstützung für dieses Gerät. Zurzeit kann nur ein Gerät die Konsolenunterstützung aktiviert haben. Das erste, in der Konfigurationsdatei aufgeführte Gerät, mit dieser Option, verfügt über eine aktivierte Konsolenunterstützung. Beachten Sie, dass diese Option alleine nicht ausreicht, um die serielle Konsole zu aktivieren. Setzen Sie entweder noch die nachfolgend diskutierte Option oder verwenden Sie beim Booten, wie unten beschrieben, den Schalter `-h`.

   `0x20`

   > Das erste Gerät in der Kernelkonfigurationsdatei mit dieser Option wird, unabhängig von dem unten diskutierten Schalter `-h`, zur Konsole. Die Option `0x20` muss zusammen mit `0x10` verwendet werden.

   `0x40`

   > Reserviert dieses Gerät und sperrt es für normale Zugriffe. Sie sollten diese Option nicht auf dem Gerät setzen, das Sie als serielle Konsole verwenden wollen. Der Zweck dieser Option ist es, dieses Gerät für das Remote-Debuggen zu reservieren. Das FreeBSD Developers' Handbook (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/developers-handbook/index.html) enthält dazu weitere Informationen.

   Beispiel:

   ```
   device sio0 at isa? port IO_COM1 tty flags 0x10 irq 4
   ```

   Weitere Einzelheiten entnehmen Sie bitte sio(4).

   Wenn diese Optionen nicht gesetzt sind, müssen Sie auf einer anderen Konsole beim Booten UserConfig starten oder den Kernel neu kompilieren.

5. Erstellen Sie `boot.config` im Rootverzeichnis der `a`-Partition des Bootlaufwerks.

   Der Code des Bootblocks entnimmt dieser Datei, wie Sie Ihr System booten möchten. Um die serielle Konsole zu aktivieren, müssen Sie hier eine oder mehrere Optionen (alle in derselben Zeile) angeben. Die folgenden Optionen stehen zur Auswahl der Konsole zur Verfügung:

   `-h`

   > Schaltet zwischen der internen und der seriellen Konsole um. Wenn Sie beispielsweise von der internen Konsole (Bildschirm) booten, weist `-h` den Bootloader und den Kernel an, die serielle Schnittstelle als Konsole zu nehmen. Wenn die Konsole normal auf der seriellen Schnittstelle liegt, wählen Sie mit `-h` den Bildschirm aus.

`-D`

Schaltet zwischen Einzelkonsole und Dual-Konsole um. Die Einzelkonsole ist entweder die interne Konsole (der Bildschirm) oder die serielle Schnittstelle, je nach dem Stand von `-h`. Im Dual-Konsolen Betrieb ist die Konsole, unabhängig von `-h`, gleichzeitig der Bildschirm und die serielle Schnittstelle. Dies trifft aber nur zu, wenn der Bootblock ausgeführt wird. Sobald der Bootloader ausgeführt wird, wird die durch `-h` gegebene Konsole die alleinige Konsole.

`-P`

Veranlasst den Bootblock nach einer Tastatur zu suchen. Wenn keine Tastatur gefunden wird, werden `-D` und `-h` automatisch gesetzt.

> **Anmerkung:** Wegen Platzbeschränkungen in den Bootblöcken kann `-P` nur erweiterte Tastaturen erkennen. Tastaturen mit weniger als 101 Tasten (und ohne F11 und F12 Tasten) werden wahrscheinlich, wie vielleicht auch die Tastaturen einiger Laptops, nicht erkannt. Wenn dies bei Ihrem System der Fall ist, können Sie `-P` nicht verwenden, da es leider noch keine Umgehung für dieses Problem gibt.

Benutzen Sie also entweder `-P`, um die Konsole automatisch zu setzen, oder `-h`, um die serielle Konsole zu verwenden.

In `boot.config` können Sie auch andere, in boot(8) beschriebene Optionen, aufnehmen.

Mit Ausnahme von `-P` werden die Optionen an den Bootloader (`/boot/loader`) weitergegeben. Der Bootloader untersucht dann einzig `-h` um festzustellen, welches Gerät die Konsole wird. Wenn Sie also nur `-D` angegeben haben, können Sie die serielle Schnittstelle nur als Konsole verwenden während der Bootblock ausgeführt wird. Danach wird der Bootloader, da ja `-h` fehlt, den Bildschirm zur Konsole machen.

6. Booten Sie die Maschine.

   Wenn Sie das FreeBSD-System starten, werden die Bootblöcke den Inhalt von `/boot.config` auf der Konsole ausgeben:

   ```
   /boot.config: -P
   Keyboard: no
   ```

   Die zweite Zeile sehen Sie nur, wenn Sie in `/boot.config` `-P` angegeben haben. Sie zeigt an, ob eine Tastatur angeschlossen ist oder nicht. Die Meldungen gehen je nach den Einstellungen in `/boot.config` auf die interne Konsole, die serielle Konsole, oder beide Konsolen.

Optionen	Meldungen erscheinen auf
keine	der internen Konsole
`-h`	der seriellen Konsole
`-D`	der seriellen und der internen Konsole
`-Dh`	der seriellen und der internen Konsole
`-P`, mit Tastatur	der internen Konsole
`-P`, ohne Tastatur	der seriellen Konsole

Nach den oben gezeigten Meldungen gibt es eine kleine Verzögerung bevor die Bootblöcke den Bootloader

laden und weitere Meldungen auf der Konsole erscheinen. Sie können die Ausführung der Bootblöcke unterbrechen, um zu überprüfen, ob auch alles richtig aufgesetzt ist, brauchen das aber unter normalen Umständen nicht zu tun.

Drücken Sie eine Taste außer **Enter** um den Bootvorgang zu unterbrechen. Sie erhalten dann ein Prompt, an dem Sie weitere Eingaben tätigen können:

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Je nach Inhalt von `/boot.config` erscheint das Prompt auf der seriellen Konsole, der internen Konsole oder beiden Konsolen. Wenn die Meldung auf der richtigen Konsole erscheint, drücken Sie **Enter** um fortzufahren.

Wenn Sie das Prompt auf der seriellen Konsole erwartet haben, dort aber nichts sehen, liegt ein Fehler in Ihren Einstellungen vor. Als Umgehung geben Sie an der momentanen Konsole `-h` ein, um den Bootblock und den Bootloader auf die serielle Konsole umzustellen. Führen Sie dann den Bootvorgang mit **Enter** weiter und wenn das System gebootet hat, können Sie die fehlerhaften Einstellungen korrigieren.

Nachdem der Bootloader geladen wurde und Sie sich in der dritten Bootphase befinden, können Sie immer noch zwischen der internen und der seriellen Konsole auswählen. Setzen Sie dazu, wie in Abschnitt 27.6.6 beschrieben, die entsprechenden Variablen des Bootloaders.

27.6.4. Zusammenfassung

Die folgende Übersicht zeigt, welche Konsole, abhängig von den getroffenen Einstellungen, ausgewählt wird.

27.6.4.1. Fall 1: Option 0x10 für `sio0`

```
device sio0 at isa? port IO_COM1 tty flags 0x10 irq 4
```

Optionen in `/boot.config`	Konsole in den Bootblöcken	Konsole im Bootloader	Konsole im Kernel
keine	interne	interne	interne
`-h`	serielle	serielle	serielle
`-D`	serielle und interne	interne	interne
`-Dh`	serielle und interne	serielle	serielle
`-P`, mit Tastatur	interne	interne	interne
`-P`, ohne Tastatur	serielle und interne	serielle	serielle

27.6.4.2. Fall 2: Option 0x30 für `sio0`

```
device sio0 at isa? port IO_COM1 tty flags 0x30 irq 4
```

Optionen in `/boot.config`	Konsole in den Bootblöcken	Konsole im Bootloader	Konsole im Kernel
keine	interne	interne	serielle

Optionen in `/boot.config`	Konsole in den Bootblöcken	Konsole im Bootloader	Konsole im Kernel
`-h`	serielle	serielle	serielle
`-D`	serielle und interne	interne	serielle
`-Dh`	serielle und interne	serielle	serielle
`-P`, mit Tastatur	interne	interne	serielle
`-P`, ohne Tastatur	serielle und interne	serielle	serielle

27.6.5. Hinweise zur seriellen Konsole

27.6.5.1. Verwenden einer höheren Geschwindigkeit

Die Vorgabewerte für die Kommunikationsparameter der seriellen Schnittstelle sind: 9600 baud, 8 Bit, keine Parität und ein Stopp-Bit. Wenn Sie die Standardgeschwindigkeit ändern wollen, haben Sie folgende Möglichkeiten:

- Geben Sie die neue Konsolengeschwindigkeit mit `BOOT_COMCONSOLE_SPEED` an und kompilieren Sie die Bootblöcke neu. Ausführliche Informationen zum Bau und zur Installation von neuen Bootblöcken finden Sie im Abschnitt 27.6.5.2 des Handbuchs.

  Wenn die serielle Konsole nicht mit der Option `-h` gestartet wird, oder wenn die verwendete serielle Konsole sich von der von den Bootblöcken verwendeten unterscheidet, müsssen Sie zusätzlich die folgende Option in Ihre Kernelkonfigurationsdatei aufnehmen und den Kernel neu bauen:

  ```
  options CONSPEED=19200
  ```

- Verwenden Sie die Option `-S`, um den Kernel zu booten. Die Option `-S` kann auch in die Datei `/boot.config` aufgenommen werden. Eine Beschreibung dieses Vorgangs sowie eine Auflistung der von `/boot.config` unterstützten Optionen finden Sie in der Manualpage boot(8).
- Aktivieren Sie die Option `comconsole_speed` in der Datei `/boot/loader.conf`.

  Diese Option setzt voraus, dass auch die Optionen `console`, `boot_serial`, sowie `boot_multicons` in der Datei `/boot/loader.conf` gesetzt sind. Im Folgenden finden Sie ein Beispiel, in dem `comconsole_speed` verwendet wird, um die Geschwindigkeit der seriellen Konsole zu ändern:

  ```
  boot_multicons="YES"
  boot_serial="YES"
  comconsole_speed="115200"
  console="comconsole,vidconsole"
  ```

27.6.5.2. Eine andere Schnittstelle als `sio0` benutzen

Wenn Sie, warum auch immer, ein anderes Gerät als `sio0` für die serielle Konsole einsetzen wollen, kompilieren Sie bitte die Bootblöcke, den Bootloader und den Kernel nach dem folgenden Verfahren neu.

1. Installieren Sie die Kernelquellen (siehe Kapitel 25).

2. Setzen Sie in `/etc/make.conf` `BOOT_COMCONSOLE_PORT` auf die Adresse der Schnittstelle (0x3F8, 0x2F8, 0x3E8 oder 0x2E8), die Sie benutzen möchten. Sie können nur `sio0` bis `sio3` (`COM1` bis `COM4`) benutzen, Multiportkarten können Sie nicht als Konsole benutzen. Interrupts müssen Sie hier nicht angeben.
3. Erstellen Sie eine angepasste Kernelkonfiguration und geben Sie dort die richtigen Optionen für die Schnittstelle, die Sie benutzen möchten, an. Wenn Sie zum Beispiel `sio1` (`COM2`) zur Konsole machen wollen, geben Sie dort entweder

   ```
   device sio1 at isa? port IO_COM2 tty flags 0x10 irq 3
   ```

   oder

   ```
   device sio1 at isa? port IO_COM2 tty flags 0x30 irq 3
   ```

   an. Keine andere serielle Schnittstelle sollte als Konsole definiert werden.
4. Übersetzen und installieren Sie die Bootblöcke und den Bootloader:

   ```
   # cd /sys/boot
   # make clean
   # make
   # make install
   ```
5. Bauen und installieren Sie einen neuen Kernel.
6. Schreiben Sie die Bootblöcke mit bsdlabel(8) auf die Bootplatte und booten Sie den neuen Kernel.

27.6.5.3. DDB Debugger über die serielle Schnittstelle

Wenn Sie den Kerneldebugger über eine serielle Verbindung bedienen möchten (nützlich, kann aber gefährlich sein, wenn auf der Leitung falsche BREAK-Signale generiert werden), sollten Sie einen Kernel mit den folgenden Optionen erstellen:

```
options BREAK_TO_DEBUGGER
options DDB
```

27.6.5.4. Benutzung der seriellen Konsole zum Anmelden

Da Sie schon die Bootmeldungen auf der Konsole verfolgen können und den Kerneldebugger über die Konsole bedienen können, wollen Sie sich vielleicht auch an der Konsole anmelden.

Öffnen Sie `/etc/ttys` in einem Editor und suchen Sie nach den folgenden Zeilen:

```
ttyu0 "/usr/libexec/getty std.9600" unknown off secure
ttyu1 "/usr/libexec/getty std.9600" unknown off secure
ttyu2 "/usr/libexec/getty std.9600" unknown off secure
ttyu3 "/usr/libexec/getty std.9600" unknown off secure
```

`ttyu0` bis `ttyu3` entsprechen `COM1` bis `COM4`. Ändern Sie für die entsprechende Schnittstelle `off` zu `on`. Wenn Sie auch die Geschwindigkeit der seriellen Schnittstelle geändert haben, müssen Sie `std.9600` auf die momentane Geschwindigkeit, zum Beispiel `std.19200`, anpassen.

Sie sollten auch den Terminaltyp von `unknown` auf den tatsächlich verwendeten Terminal setzen.

Damit die Änderungen an der Datei wirksam werden, müssen Sie noch `kill -HUP 1` absetzen.

27.6.6. Die Konsole im Bootloader ändern

In den vorigen Abschnitten wurde beschrieben, wie Sie die serielle Konsole durch Änderungen im Bootblock aktivieren. Dieser Abschnitt zeigt Ihnen, wie Sie mit Kommandos und Umgebungsvariablen die Konsole im Bootloader definieren. Da der Bootloader die dritte Phase im Bootvorgang ist und nach den Bootblöcken ausgeführt wird, überschreiben seine Einstellungen die des Bootblocks.

27.6.6.1. Festlegen der Konsole

Mit einer einzigen Zeile in `/boot/loader.conf` können Sie den Bootloader und den Kernel anweisen, die serielle Schnittstelle zur Konsole zu machen:

```
console="comconsole"
```

Unabhängig von den Einstellungen im Bootblock legt dies die Konsole fest.

Die obige Zeile sollte die erste Zeile in `/boot/loader.conf` sein, so dass Sie die Bootmeldungen so früh wie möglich auf der Konsole sehen.

Analog können Sie die interne Konsole verwenden:

```
console="vidconsole"
```

Wenn Sie `console` nicht setzen, bestimmt der Bootloader (und damit auch der Kernel) die Konsole über die `-h` Option des Bootblocks.

Sie können die Bootkonsole in `/boot/loader.conf.local` oder `/boot/loader.conf` angeben.

Weitere Informationen erhalten Sie in loader.conf(5).

> **Anmerkung:** Momentan gibt es im Bootloader nichts vergleichbares zu `-P` im Bootblock. Damit kann die Konsole nicht automatisch über das Vorhandensein einer Tastatur festgelegt werden.

27.6.6.2. Eine andere Schnittstelle als `sio0` benutzen

Sie müssen den Bootloader neu kompilieren, wenn Sie eine andere Schnittstelle als `sio0` benutzen wollen. Folgen Sie der Anleitung aus Abschnitt 27.6.5.2.

27.6.7. Vorbehalte

Hinter dem ganzen steckt die Idee, Server ohne Hardware für Grafik und ohne Tastatur zu betreiben. Obwohl es die meisten Systeme erlauben, ohne Tastatur zu booten, gibt es leider nur wenige Systeme, die ohne eine Grafikkarte booten. Maschinen mit einem AMI BIOS können ohne Grafik booten, indem Sie den Grafikadapter im CMOS-Setup auf `Not installed` setzen.

Viele Maschinen unterstützen diese Option allerdings nicht. Damit diese Maschinen booten, müssen sie über eine Grafikkarte, auch wenn es nur eine alte Monochromkarte ist, verfügen. Allerdings brauchen Sie keinen Monitor an die Karte anzuschließen. Sie können natürlich auch versuchen, auf diesen Maschinen ein AMI BIOS zu installieren.

Kapitel 28. PPP und SLIP

Restrukturiert, neu organisiert und aktualisiert von Jim Mock. Übersetzt von Thomas Schwarzkopf.

28.1. Übersicht

Unter FreeBSD stehen verschiedene Möglichkeiten zur Verfügung, um Computer miteinander zu verbinden. Der Aufbau einer Netzwerk- oder Internetverbindung mit Hilfe eines Einwahlmodems – für den eigenen oder für andere Rechner – erfordert den Einsatz von PPP oder SLIP.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- Wie Sie User-PPP einrichten.
- Wie Sie Kernel-PPP einrichten (nur für FreeBSD 7.X relevant).
- Was zu tun ist, um PPPoE (PPP over Ethernet) einzurichten.
- Wie Sie PPPoA (PPP over ATM) einrichten.
- Wie Sie einen SLIP-Client und -Server einrichten und konfigurieren ((nur für FreeBSD 7.X relevant)

Bevor Sie dieses Kapitel lesen, sollten Sie:

- mit den grundlegenden Begriffen der Netzwerktechnik vertraut sein.
- die Grundlagen und den Zweck einer Einwahlverbindung sowie PPP und/oder SLIP kennen.

Sie fragen sich vielleicht, worin denn der Hauptunterschied zwischen User-PPP und Kernel-PPP liegt. Die Antwort ist einfach: User-PPP verarbeitet die ein- und ausgehenden Daten im Userland, statt im Kernel. Dies ist zwar aufwändig, im Hinblick auf die Daten, die dadurch zwischen Kernel und Userland hin und her kopiert werden müssen, doch es ermöglicht auch eine PPP-Implementierung mit weitaus mehr Funktionen. User-PPP verwendet das Gerät `tun`, um mit anderen Rechnern zu kommunizieren, während Kernel-PPP hierfür das Gerät `ppp` benutzt.

> **Anmerkung:** In diesem Kapitel wird durchgängig vom Programm **ppp** gesprochen, wenn damit User-PPP gemeint ist. Ausnahmen werden gemacht, wenn eine Unterscheidung gegenüber anderer PPP-Software, wie **pppd** (nur unter FreeBSD 7.X), notwendig wird. Soweit nichts anderes angegeben ist, sollten alle Befehle, die in diesem Kapitel erklärt werden, als `root` ausgeführt werden.

28.2. User-PPP

Aktualisiert und erweitert von Tom Rhodes. Ursprünglich geschrieben von Brian Somers. Mit Beiträgen von Nik Clayton, Dirk Frömberg und Peter Childs.

> **Warnung:** Mit FreeBSD 8.0 wurden die Gerätedateien für serielle Ports umbenannt: `/dev/cuad`*`N`* wurde zu `/dev/cuau`*`N`*, `/dev/ttyd`*`N`* zu `/dev/ttyu`*`N`*. Verwenden Sie noch FreeBSD 7.X, müssen Sie dies beim Lesen der folgenden Abschnitte berücksichtigen.

28.2.1. User-PPP

28.2.1.1. Voraussetzungen

Dieses Dokument geht davon aus, dass Sie Folgendes zur Verfügung haben:

- Einen Account bei einem Internet Service Provider (ISP), zu dem Sie mit PPP eine Verbindung aufbauen können.
- Ein Modem oder ein anderes Gerät, das, richtig konfiguriert und mit Ihrem Rechner verbunden, Ihnen die Herstellung einer Verbindung zu Ihrem ISP erlaubt.
- Die Einwahlnummer(n) Ihres ISP.
- Ihren Login-Namen und Ihr Passwort (entweder ein reguläres Login/Passwort-Paar im UNIX-Stil oder ein PAP bzw. CHAP Login/Passwort-Paar).
- Die IP-Adresse von einem oder mehreren Nameservern. Üblicherweise werden Ihnen von Ihrem ISP zwei IP-Adressen für diesen Zweck zur Verfügung gestellt. Wenn Sie keine solche IP-Adresse von Ihrem Provider bekommen haben, können Sie das Kommando `enable dns` in der Datei `ppp.conf` verwenden, um **ppp** anzuweisen, den Nameserver für Sie einzutragen. Diese Funktion setzt allerdings voraus, dass Ihr ISP eine PPP-Implementierung verwendet, die das Aushandeln eines Nameservers unterstützt.

Die folgenden Informationen werden Ihnen möglicherweise von Ihrem ISP zur Verfügung gestellt, sie sind aber nicht zwingend erforderlich:

- Die Gateway IP-Adresse Ihres ISP. Als Gateway wird der Computer bezeichnet, zu dem Sie eine Verbindung aufbauen. Die IP-Adresse dieses Rechners wird als *default route* eingetragen. Wenn Sie diese Information nicht zur Verfügung haben, kann PPP so konfiguriert werden, dass der PPP-Server Ihres ISP während des Verbindungsaufbaus eine gültige Adresse übermittelt.

  ppp bezieht sich mit `HISADDR` auf diese IP-Adresse.
- Die Netzmaske, die Sie verwenden sollten. Falls Ihnen Ihr ISP keine Netzmaske vorgegeben hat, können Sie `255.255.255.255` verwenden.
- Wenn Ihnen Ihr ISP eine statische IP-Adresse zur Verfügung stellt, können Sie diese eintragen. Andernfalls lassen wir uns einfach von der Gegenstelle eine IP-Adresse zuweisen.

Falls Ihnen die erforderlichen Informationen fehlen sollten, nehmen Sie bitte Kontakt mit Ihrem ISP auf.

> **Anmerkung:** Die Beispieldateien, die in diesem Kapitel dargestellt werden, enthalten Zeilennummern. Die Nummerierung dient lediglich einer leichteren Orientierung und sollte von Ihnen nicht in Ihre Dateien übernommen werden. Richtiges Einrücken, durch Tabulatoren und Leerzeichen, ist ebenfalls wichtig.

28.2.1.2. Automatische Konfiguration von PPP

Sowohl `ppp` als auch `pppd` (die PPP-Implementierung auf Kernelebene, nur unter FreeBSD 7.X) verwenden die Konfigurationsdateien im Verzeichnis `/etc/ppp`. Beispiele für User-PPP sind in `/usr/share/examples/ppp/` zu finden.

Die Konfiguration von `ppp` erfordert, je nach Ihren besonderen Bedingungen, die Bearbeitung einiger Dateien. Was Sie in diese Dateien eintragen, hängt unter anderem davon ab, ob Ihnen Ihr ISP eine statische IP-Adresse (Sie verwenden immer dieselbe IP-Adresse, die Ihnen einmal zugeteilt wurde) oder eine dynamische IP-Adresse (Ihre IP-Adresse ändert sich bei jeder Verbindung mit dem ISP) zugewiesen hat.

28.2.1.2.1. PPP und statische IP-Adressen

Sie müssen die Konfigurationsdatei `/etc/ppp/ppp.conf` bearbeiten. Sie sollte so aussehen, wie in dem unten angegebenen Beispiel.

> **Anmerkung:** Zeilen die mit einem : enden, beginnen in der ersten Spalte (am Beginn der Zeile). Alle anderen Zeilen sollten wie dargestellt durch Leerzeichen oder Tabulatoren eingerückt werden.

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION (built COMPILATIONDATE)
4       set device /dev/cuau0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 \"\" \"\" gin:--gin: \\U word: \\P col: ppp"
16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR
```

Zeile 1:

Gibt den Standardeintrag an. Befehle dieses Eintrags werden automatisch ausgeführt, wenn ppp läuft.

Zeile 2:

Schaltet die Loggingparameter ein. Wenn die Verbindung zufriedenstellend funktioniert, können Sie diese Zeile verkürzen:

```
set log phase tun
```

Dies verhindert ein übermäßiges Anwachsen der Logdateien.

Zeile 3:

Gibt PPP an, wie es sich gegenüber der Gegenstelle identifizieren soll. PPP identifiziert sich gegenüber der Gegenstelle, wenn es Schwierigkeiten bei der Aushandlung und beim Aufbau der Verbindung gibt. Dabei werden Informationen bereitgestellt, die dem Administrator der Gegenstelle helfen können, die Ursache der Probleme zu finden.

Zeile 4:

Gibt das Device an, an dem das Modem angeschlossen ist. `COM1` entspricht `/dev/cuad0` und `COM2` entspricht `/dev/cuad1`.

Zeile 5:

Legt die Geschwindigkeit fest, mit der Sie die Verbindung betreiben möchten. Falls ein Wert von 115200 nicht funktioniert (was aber bei jedem einigermaßen neuen Modem funktionieren sollte), versuchen Sie es stattdessen mit 38400.

Zeilen 6 & 7:

Die Zeichenfolge für die Einwahl. User-PPP verwendet eine expect-send Syntax, ähnlich dem chat(8)-Programm. Weitere Informationen zu den Eigenschaften dieser Sprache bietet die Manual-Seite.

Beachten Sie, dass dieser Befehl aufgrund der besseren Lesbarkeit auf der nächsten Zeile weitergeht. Das kann für jeden Befehl in `ppp.conf` gelten, wenn `\` das letzte Zeichen in einer Zeile ist.

Zeile 8:

Legt den Zeitrahmen fest, innerhalb dessen eine Reaktion erfolgen muss. Der Standardwert liegt bei 180 Sekunden, so dass diese Zeile lediglich einen kosmetischen Charakter hat.

Zeile 9:

Weist PPP an, bei der Gegenstelle eine Bestätigung der lokalen Resolvereinstellungen anzufordern. Wenn Sie einen lokalen Nameserver betreiben, sollte diese Zeile auskommentiert oder gelöscht werden.

Zeile 10:

Eine leere Zeile zur besseren Lesbarkeit. Leere Zeilen werden von PPP ignoriert.

Zeile 11:

Bestimmt einen Provider, namens “provider”. Wenn Sie hier den Namen Ihres ISP einsetzen, können Sie später die Verbindung mit `load` *`ISP`* aufbauen.

Zeile 12:

Gibt die Telefonnummer des Providers an. Mehrere Telefonnummern können angegeben werden, indem Doppelpunkte (`:`) oder Pipe-Zeichen (`|`) als Trennzeichen verwendet werden. Der Unterschied zwischen diesen beiden Trennzeichen ist in ppp(8) beschrieben. Zusammenfassend: Wenn Sie die verschiedenen Nummern abwechselnd verwenden möchten, sollten Sie die Nummern durch einen Doppelpunkt trennen. Wenn Sie immer die erste Nummer verwenden möchten und die anderen nur zum Einsatz kommen sollen, wenn eine Einwahl mit der ersten Telefonnummer nicht möglich ist, sollten Sie das Pipe-Zeichen zur Trennung verwenden. Wie im Beispiel, sollten Sie die gesamte Reihe der Telefonnummern in Anführungszeichen setzen.

Sie müssen die Telefonnummer in Anführungszeichen (") setzen, wenn Sie Leerzeichen in der Telefonnummer verwenden, ansonsten rufen Sie einen Fehler hervor, der vielleicht schwer zu finden ist.

Zeilen 13 & 14:

Gibt den Benutzernamen und das Passwort an. Wenn Sie zur Verbindung einen Login-Prompt im UNIX-Stil verwenden, bezieht sich der Befehl `set login` mit den \U und \P Variablen auf diese Werte. Wenn Sie zum Verbindungsaufbau PAP oder CHAP verwenden, werden diese Werte zum Zeitpunkt der Authentifizierung verwendet.

Zeile 15:

Wenn Sie PAP oder CHAP einsetzen, gibt es an dieser Stelle keinen Login-Prompt, weshalb Sie diese Zeile auskommentieren oder löschen sollten. Der Abschnitt Authentifizierung mit PAP und CHAP enthält hierzu weitere Einzelheiten.

Der Login-String hat die gleiche chat-ähnliche Syntax, wie der Einwahlstring. Der String in diesem Beispiel funktioniert mit einem ISP, dessen Login-Session folgendermaßen aussieht:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

Sie müssen dieses Skript noch an Ihre eigenen Erfordernisse anpassen. Wenn Sie dieses Skript zum ersten Mal schreiben, sollten Sie sicherstellen, dass Sie “chat”-logging aktiviert haben, damit Sie überprüfen zu können, ob die Konversation zwischen Ihrem Rechner und dem Rechner des Providers wie erwartet abläuft.

Zeile 16:

Setzt einen Zeitrahmen (in Sekunden), innerhalb dessen eine Reaktion erfolgen muss. In diesem Fall, wird die Verbindung nach 300 Sekunden automatisch geschlossen, wenn keine Aktivität zu verzeichnen ist. Wenn Sie keinen Zeitrahmen festlegen wollen, nach dessen Überschreiten die Verbindung geschlossen wird, können Sie diesen Wert auf 0 setzen oder die Kommandozeilen-Option `-ddial` verwenden.

Zeile 17:

Gibt die IP-Adresse für das Interface an. Der String *x.x.x.x* sollte durch die IP-Adresse ersetzt werden, die Ihnen Ihr Provider zugeteilt hat. Der String *y.y.y.y* sollte durch die IP-Adresse ersetzt werden, die Ihr ISP als Gateway angegeben hat (das ist der Rechner, mit dem Ihr Rechner eine Verbindung aufbaut). Wenn Ihnen Ihr ISP keine Gateway Adresse zur Verfügung gestellt hat, verwenden Sie hier einfach `10.0.0.2/0`. Wenn Sie eine “erratene” IP-Adresse verwenden müssen, sollten Sie in der Datei `/etc/ppp/ppp.linkup` einen entsprechenden Eintrag machen. Folgen Sie dazu den Anweisungen im Abschnitt
PPP und dynamische IP-Adressen. Wenn diese Zeile ausgelassen wird, kann `ppp` nicht im `-auto` Modus betrieben werden.

Zeile 18:

Fügt eine Defaultroute für das Gateway Ihres Providers hinzu. Das Wort `HISADDR` wird dabei durch die in Zeile 17 angegebene Gateway Adresse ersetzt. Wichtig ist, dass diese Zeile nach Zeile 17 erscheint, da

andernfalls `HISADDR` noch nicht initialisiert ist.

Wenn Sie ppp nicht im `-auto` Modus betreiben, sollte diese Zeile in die Datei `ppp.linkup` verschoben werden.

Wenn Sie eine statische IP-Adresse verwenden und ppp im `-auto` Modus läuft, ist es nicht notwendig, einen Eintrag in die Datei `ppp.linkup` hinzuzufügen. In diesem Fall hat ihre Routingtabelle bereits die richtigen Einträge, bevor Sie die Verbindung aufbauen. Sie möchten aber vielleicht einen Eintrag hinzufügen, um ein Programm aufzurufen, nachdem die Verbindung aufgebaut ist. Dies wird weiter unten am Beispiel von Sendmail erklärt.

Beispiele für Konfigurationsdateien finden Sie im Verzeichnis `/usr/share/examples/ppp/`.

28.2.1.2.2. PPP und dynamische IP-Adressen

Wenn Ihnen Ihr ISP keine statische IP-Adresse zuteilt, kann `ppp` so konfiguriert werden, dass die lokale und die entfernte IP-Adresse beim Verbindungsaufbau ausgehandelt werden. Dies geschieht, indem zunächst eine IP-Adresse “erraten” wird, die von `ppp`, unter Verwendung des IP Configuration Protocol (IPCP) durch eine richtige ersetzt wird, wenn die Verbindung aufgebaut ist. Die Konfiguration der Datei `ppp.conf` entspricht derjenigen, die im Abschnitt PPP und statische IP- Adressen dargestellt wurde, jedoch mit folgender Änderung:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255 0.0.0.0
```

Auch hier dient die Zeilennummerierung lediglich der besseren Übersichtlichkeit. Einrückungen, von mindestens einem Leerzeichen, sind allerdings erforderlich.

Zeile 17:

Die Zahl nach dem `/` Zeichen, gibt die Anzahl der Bits der Adresse an, auf die ppp besteht. Sie möchten vielleicht andere IP-Adressen verwenden, die oben angegebenen werden aber immer funktionieren.

Das letzte Argument (`0.0.0.0`) weist PPP an, den Verbindungsaufbau mit der Adresse `0.0.0.0` zu beginnen, statt `10.0.0.1` zu verwenden. Dies ist bei einigen ISPs notwendig. Verwenden Sie nicht `0.0.0.0` als erstes Argument für `set ifaddr`, da so verhindert wird, dass PPP im `-auto` Modus eine initiale Route setzt.

Wenn PPP nicht im `-auto` Modus läuft, müssen Sie die Datei`/etc/ppp/ppp.linkup` editieren. `ppp.linkup` kommt zum Einsatz, wenn eine Verbindung aufgebaut worden ist. Zu diesem Zeitpunkt hat `ppp` die Interface Adressen vergeben und es ist möglich, die Einträge in der Routingtabelle hinzuzufügen:

```
1     provider:
2      add default HISADDR
```

Zeile 1:

Beim Aufbau einer Verbindung sucht `ppp` in der Datei `ppp.linkup` nach einem Eintrag. PPP geht dabei nach folgenden Regeln vor: Suche zunächst nach der gleichen Bezeichnung, die wir auch in der Datei `ppp.conf` verwendet haben. Falls das nicht funktioniert, suche nach einem Eintrag der IP-Adresse unseres Gateways. Dieser Eintrag ist eine Bezeichnung im Stil von IP-Adressen, die sich aus vier Oktetts zusammensetzt. Falls immer noch kein passender Eintrag gefunden wurde, suche nach dem Eintrag `MYADDR`.

Zeile 2:

Diese Zeile weist `ppp` an, eine Defaultroute zu verwenden, die auf `HISADDR` zeigt. `HISADDR` wird nach der Aushandlung mit IPCP durch die IP-Adresse des Gateways ersetzt.

Die Dateien `/usr/share/examples/ppp/ppp.conf.sample` und `/usr/share/examples/ppp/ppp.linkup.sample` bieten detaillierte Beispiele für `pmdemand` Einträge.

28.2.1.2.3. Annahme eingehender Anrufe

Wenn Sie **ppp** auf einem Rechner, der in ein LAN eingebunden ist, so konfigurieren, dass eingehende Anrufe angenommen werden, müssen Sie entscheiden, ob Pakete an das LAN weitergeleitet werden sollen. Wenn Sie das möchten, sollten Sie an die Gegenstelle eine IP-Adresse aus Ihrem lokalen Subnetz vergeben und den Befehl `enable proxy` in die Datei `/etc/ppp/ppp.conf` einfügen. Außerdem sollte die Datei `/etc/rc.conf` Folgendes enthalten:

```
gateway_enable="YES"
```

28.2.1.2.4. Welches getty?

Der Abschnitt Einwählverbindungen bietet eine gute Beschreibung, wie Einwählverbindungen unter Verwendung von getty(8) genutzt werden können.

Eine Alternative zu `getty` ist mgetty (http://mgetty.greenie.net/), eine raffiniertere Version von `getty`, die mit Blick auf Einwählverbindungen entworfen wurde. Sie können dieses Paket über den Port `comms/mgetty+sendfax` installieren.

Der Vorteil von `mgetty` ist, dass es auf aktive Weise mit Modems *spricht*, das heißt wenn ein Port in `/etc/ttys` ausgeschaltet ist, wird Ihr Modem nicht auf Anrufe reagieren.

Spätere Versionen von `mgetty` (von 0.99beta aufwärts) unterstützen auch die automatische Erkennung von PPP-Streams, was Ihren Clients den skriptlosen Zugang zu Ihren Servern erlaubt.

Der Abschnitt Mgetty und AutoPPP bietet weitere Informationen zu `mgetty`.

28.2.1.2.5. ***PPP*** *und Rechte*

Der Befehl `ppp` muss normalerweise als `root` ausgeführt werden. Wenn Sie jedoch möchten, dass `ppp` im Server-Modus auch von einem normalen Benutzer, wie unten beschrieben, durch Aufruf von `ppp` ausgeführt werden kann, müssen Sie diesem Benutzer die Rechte erteilen, `ppp` auszuführen, indem Sie ihn in der Datei `/etc/group` der Gruppe `network` hinzufügen.

Sie werden ihm ebenfalls den Zugriff auf einen oder mehrere Abschnitte der Konfigurationsdatei geben müssen, indem Sie den `allow` Befehl verwenden:

```
allow users fred mary
```

Wenn dieser Befehl im `default` Abschnitt verwendet wird, erhalten die angegebenen Benutzer vollständigen Zugriff.

28.2.1.2.6. PPP-Shells für dynamische IP-Adressen

Erzeugen Sie eine Datei mit dem Namen `/etc/ppp/ppp-shell`, die Folgendes enthält:

```
#!/bin/sh
IDENT=‘echo $0 | sed -e ’s/^.*-\(.*\)$/\1/’‘
```

```
CALLEDAS="$IDENT"
TTY=‘tty‘

if [ x$IDENT = xdialup ]; then
        IDENT=‘basename $TTY‘
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

Dieses Skript sollte ausführbar sein. Nun erzeugen Sie einen symbolischen Link `ppp-dialup` auf dieses Skript mit folgendem Befehl:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

Sie sollten dieses Skript als *Shell* für alle Benutzer von Einwählverbindungen verwenden. Dies ist ein Beispiel aus der Datei `/etc/passwd` für einen Benutzer namens `pchilds`, der PPP für Einwählverbindungen verwenden kann (Denken Sie daran, die Passwortdatei nicht direkt zu editieren, sondern dafür vipw(8) zu verwenden).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Erstellen Sie ein Verzeichnis `/home/ppp`, das von allen Benutzern gelesen werden kann und die folgenden leeren Dateien enthält:

```
-r--r--r--   1 root     wheel           0 May 27 02:23 .hushlogin
-r--r--r--   1 root     wheel           0 May 27 02:22 .rhosts
```

Dies verhindert, dass `/etc/motd` angezeigt wird.

28.2.1.2.7. PPP-Shells für statische IP-Adressen

Erstellen Sie die Datei `ppp-shell` wie oben oben dargestellt. Erzeugen Sie nun für jeden Account mit statischer IP-Adresse einen symbolischen Link auf `ppp-shell`.

Wenn Sie beispielsweise die drei Kunden, `fred`, `sam` und `mary` haben, für die Sie CIDR-/24-Netzwerke routen, schreiben Sie Folgendes:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

Jeder Einwählzugang dieser Kunden sollte den oben erzeugten symbolischen Link als Shell haben (`mary`’s Shell sollte also `/etc/ppp/ppp-mary` sein).

28.2.1.2.8. Einrichten von `ppp.conf` für dynamische IP-Adressen

Die Datei `/etc/ppp/ppp.conf` sollte in etwa wie folgt aussehen:

```
default:
  set debug phase lcp chat
```

```
  set timeout 0

ttyu0:
  set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
  enable proxy

ttyu1:
  set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
  enable proxy
```

Anmerkung: Die Einrückungen sind wichtig.

Der Abschnitt `default:` wird für jede Sitzung geladen. Erstellen Sie für jede Einwählverbindung, die Sie in der Datei `/etc/ttys` ermöglicht haben, einen Eintrag, wie oben für `ttyu0:` gezeigt. Jede Verbindung sollte eine eigene IP-Adresse aus dem Pool der Adressen bekommen, die sie für diese Benutzergruppe reserviert haben.

28.2.1.2.9. Einrichten von `ppp.conf` für statische IP-Adressen

Zu dem bisher dargestellten Inhalt der Beispieldatei `/usr/share/examples/ppp/ppp.conf` sollten Sie einen Abschnitt für jeden Benutzer mit statisch zugewiesener IP-Adresse hinzufügen. Wir werden nun unser Beispiel mit den Accounts `fred`, `sam` und `mary` weiterführen.

```
fred:
  set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
  set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
  set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

Die Datei `/etc/ppp/ppp.linkup` sollte, falls erforderlich, ebenfalls Routinginformationen für jeden Benutzer mit statischer IP-Adresse enthalten. Die unten dargestellte Zeile würde dem Netzwerk `203.14.101.0/24` eine Route über die PPP-Verbindung des Client hinzufügen.

```
fred:
  add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
  add 203.14.102.0 netmask 255.255.255.0 HISADDR

mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

28.2.1.2.10. `mgetty` und AutoPPP

In der Voreinstellung wird `mgetty` mit der Option `AUTO_PPP` konfiguriert und kompiliert. Dadurch kann `mgetty` die LCP Phase von PPP-Verbindungen erkennen und automatisch eine ppp-Shell starten. Da hierbei jedoch die Login/Passwort-Sequenz nicht durchlaufen wird, ist es notwendig, Benutzer durch PAP oder CHAP zu authentifizieren.

In diesem Abschnitt wird davon ausgegangen, dass der Benutzer den Port `comms/mgetty+sendfax` auf seinem System kompiliert und installiert hat.

Stellen Sie sicher, dass die Datei `/usr/local/etc/mgetty+sendfax/login.config` Folgendes enthält:

```
/AutoPPP/ -     -                       /etc/ppp/ppp-pap-dialup
```

Hierdurch wird `mgetty` angewiesen, das Skript `ppp-pap-dialup` für die erkannten PPP-Verbindungen auszuführen.

Erstellen Sie nun die Datei `/etc/ppp/ppp-pap-dialup` mit folgendem Inhalt (die Datei sollte ausführbar sein):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Erstellen Sie bitte für jede Einwählverbindung, die Sie in `/etc/ttys` ermöglicht haben, einen korrespondierenden Eintrag in der Datei `/etc/ppp/ppp.conf`. Diese Einträge können problemlos, mit den Definitionen die wir weiter oben gemacht haben, koexistieren.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Jeder Benutzer, der sich auf diese Weise anmeldet, benötigt einen Benutzernamen und ein Passwort in der Datei `/etc/ppp/ppp.secret`. Sie haben auch die Möglichkeit, Benutzer mit Hilfe von PAP zu authentifizieren, indem Sie der Datei `/etc/passwd` folgende Option hinzufügen:

```
enable passwdauth
```

Wenn Sie bestimmten Benutzern eine statische IP-Adresse zuweisen möchten, können Sie diese als drittes Argument in der Datei `/etc/ppp/ppp.secret` angeben. In `/usr/share/examples/ppp/ppp.secret.sample` finden Sie hierfür Beispiele.

28.2.1.2.11. MS-Erweiterungen

Es ist möglich PPP so zu konfigurieren, dass bei Bedarf DNS und NetBIOS Nameserveradressen bereitgestellt werden.

Um diese Erweiterungen für die PPP Version 1.x zu aktivieren, sollte der entsprechende Abschnitt der Datei `/etc/ppp/ppp.conf` um folgende Zeilen ergänzt werden:

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Für PPP Version 2 und höher:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Damit werden den Clients die primären und sekundären Nameserveradressen sowie ein NetBIOS Nameserver-Host mitgeteilt.

In Version 2 und höher verwendet PPP die Werte, die in `/etc/resolv.conf` zu finden sind, wenn die Zeile `set dns` weggelassen wird.

28.2.1.2.12. Authentifizierung durch PAP und CHAP

Einige ISPs haben ihr System so eingerichtet, dass der Authentifizierungsteil eines Verbindungsaufbaus mit Hilfe von PAP oder CHAP-Mechanismen durchgeführt wird. Wenn dies bei Ihnen der Fall sein sollte, wird Ihnen Ihr ISP bei der Verbindung keinen `login:`-Prompt präsentieren, sondern sofort mit der Aushandlung der PPP-Verbindung beginnen.

PAP ist nicht so sicher wie CHAP, doch die Sicherheit ist hierbei normalerweise kein Problem, da Passwörter, obgleich von PAP im Klartext versandt, lediglich über die serielle Verbindung verschickt werden. Es gibt für Cracker wenig Möglichkeiten zu “lauschen”.

Zurückkommend auf die Abschnitte PPP und statische IP-Adressen oder PPP und dynamische IP-Adressen müssen folgende Veränderungen vorgenommen werden:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

Zeile 13:

> Diese Zeile legt Ihren PAP/CHAP Benutzernamen fest. Sie müssen den richtigen Wert für *`MyUserName`* eingeben.

Zeile 14:

> Diese Zeile legt Ihr PAP/CHAP Passwort fest. Sie müssen den richtigen Wert für *`MyPassword`* eingeben. Sie können eine zusätzliche Zeile, wie etwa:
>
> ```
> 16      accept PAP
> ```
>
> oder
>
> ```
> 16      accept CHAP
> ```
>
> verwenden, um deutlich zu machen, dass dies beabsichtigt ist, aber sowohl PAP wie auch CHAP als standardmäßig akzeptiert werden.

Zeile 15:

> Ihr ISP wird normalerweise nicht von Ihnen verlangen, dass Sie sich am Server einloggen, wenn Sie PAP oder CHAP verwenden. Sie müssen deshalb den String “set login” deaktivieren.

28.2.1.2.13. Veränderung Ihrer ppp Konfiguration im laufenden Betrieb

Es ist möglich, dem Programm ppp Befehle zu erteilen, während es im Hintergrund läuft. Dazu ist jedoch die Einrichtung eines passenden Diagnose-Ports erforderlich. Ergänzen Sie hierzu Ihre Konfigurationsdatei um folgende Zeile:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

Damit wird PPP angewiesen, auf den angegebenen UNIX-Domainsocket zu hören und Clients nach dem angegebenen Passwort zu fragen, bevor der Zugang Gewährt wird. Das %d wird durch die Nummer des benutzten tun-Devices ersetzt.

Wenn ein Socket eingerichtet ist, kann das Programm pppctl(8) in Skripten verwendet werden, mit denen in das laufende Programm eingegriffen wird.

28.2.1.3. Interne NAT von PPP benutzen

PPP kann Network Address Translation (NAT) ohne Hilfe des Kernels durchführen. Wenn Sie diese Funktion benutzen wollen, fügen Sie die folgende Zeile in /etc/ppp/ppp.conf ein:

```
nat enable yes
```

Sie können NAT mit der Option -nat auf der Kommandozeile von PPP aktivieren. Weiterhin kann NAT in /etc/rc.conf mit der Variablen ppp_nat aktiviert werden. Dies ist auch die Voreinstellung.

Die nachstehende /etc/ppp/ppp.conf benutzt NAT für bestimmte eingehende Verbindungen:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

Wenn Sie Verbindungen von außen überhaupt nicht trauen, benutzen Sie die folgende Zeile:

```
nat deny_incoming yes
```

28.2.1.4. Abschließende Systemkonfiguration

Sie haben ppp nun konfiguriert, aber bevor PPP eingesetzt werden kann, gibt noch einige weitere Dinge zu erledigen, die alle die Bearbeitung der Datei /etc/rc.conf erfordern.

Gehen Sie diese Datei von oben nach unten durch, und stellen Sie als Erstes sicher, dass die Zeile hostname= vorhanden ist:

```
hostname="foo.example.com"
```

Wenn Ihnen Ihr ISP eine statische IP-Adresse und einen Namen zugewiesen hat, ist es wahrscheinlich am besten, wenn Sie diesen Namen als Hostnamen verwenden.

Schauen Sie nach der Variable network_interfaces. Wenn Sie Ihr System so konfigurieren möchten, dass bei Bedarf eine Verbindung zu Ihrem ISP aufgebaut wird, sollten Sie das Device tun0 zu der Liste hinzufügen oder es andernfalls entfernen.

```
network_interfaces="lo0 tun0"
```

```
ifconfig_tun0=
```

> **Anmerkung:** Die Variable `ifconfig_tun0` sollte leer sein und eine Datei namens `/etc/start_if.tun0` sollte erstellt werden. Diese Datei sollte die nachfolgende Zeile enthalten:
>
> ```
> ppp -auto mysystem
> ```
>
> Dieses Skript startet Ihren ppp-Dæmon im Automatik-Modus. Es wird bei der Netzwerkkonfiguration ausgeführt. Wenn Ihr Rechner als Gateway für ein LAN fungiert, möchten Sie vielleicht auch die Option `-alias` verwenden. In der Manual-Seite sind weitere Einzelheiten hierzu zu finden.

Stellen Sie sicher, dass der Start eines Routerprogramms in `/etc/rc.conf` wie folgt deaktiviert ist:

```
router_enable="NO"
```

Es ist wichtig, dass der `routed` Dæmon nicht gestartet wird da `routed` dazu tendiert, die von `ppp` erstellten Einträge der Standardroute zu überschreiben.

Es ist außerdem sinnvoll, darauf zu achten, dass die Zeile `sendmail_flags` nicht die Option `-q` enthält, da `sendmail` sonst ab und zu die Netzwerkverbindung prüfen wird, was möglicherweise dazu führt, dass sich Ihr Rechner einwählt. Sie können hier Folgendes angeben:

```
sendmail_flags="-bd"
```

Der Nachteil dieser Lösung ist, dass Sie `sendmail` nach jedem Aufbau einer ppp-Verbindung auffordern müssen, die Mailwarteschlange zu überprüfen, indem Sie Folgendes eingeben:

```
# /usr/sbin/sendmail -q
```

Vielleicht möchten Sie den Befehl `!bg` in der Datei `ppp.linkup` verwenden, um dies zu automatisieren:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

Wenn Sie dies nicht möchten, ist es möglich, einen “dfilter” einzusetzen, um SMTP-Verkehr zu blockieren. Weitere Einzelheiten hierzu finden Sie in den Beispieldateien.

Das Einzige, was nun noch zu tun bleibt, ist Ihren Rechner neu zu starten. Nach dem Neustart können Sie entweder:

```
# ppp
```

und danach `dial provider` eingeben, um eine PPP-Sitzung zu starten, oder Sie geben:

```
# ppp -auto provider
```

ein, um `ppp` bei Datenverkehr aus Ihrem Netzwerk heraus, automatisch eine Verbindung herstellen zu lassen (vorausgesetzt Sie haben kein `start_if.tun0` Skript erstellt).

28.2.1.5. Zusammenfassung

Die folgenden Schritte sind nötig, wenn ppp zum ersten Mal eingerichtet werden soll:

Clientseite:

1. Stellen Sie sicher, dass das `tun` Device in den Kernel eingebaut ist.
2. Vergewissern Sie sich, dass die Gerätedatei `tun`*`N`* im Verzeichnis `/dev` vorhanden ist.
3. Bearbeiten Sie die Datei `/etc/ppp/ppp.conf`. Das Beispiel `pmdemand` sollte für die meisten ISP ausreichen.
4. Wenn Sie eine dynamische IP-Adresse haben, erstellen Sie einen Eintrag in der Datei `/etc/ppp/ppp.linkup`.
5. Aktualisieren Sie die Datei `/etc/rc.conf`.
6. Erstellen Sie das Skript `start_if.tun0`, wenn Sie einen bedarfgesteuerten Einwahlprozess (*demand dialing*) benötigen.

Serverseite:

1. Stellen Sie sicher, dass das `tun` Device in den Kernel eingebaut ist.
2. Vergewissern Sie sich, dass die Gerätedatei `tun`*`N`* im Verzeichnis `/dev` vorhanden ist
3. Erstellen Sie einen Eintrag in der Datei `/etc/passwd` (verwenden Sie dazu das Programm vipw(8)).
4. Erstellen Sie ein Profil im Heimatverzeichnis des Benutzers, das `ppp -direct direct-server` o.Ä. ausführt.
5. Bearbeiten Sie die Datei `/etc/ppp/ppp.conf`. Das Beispiel `direct-server` sollte ausreichen.
6. Erzeugen Sie einen Eintrag in `/etc/ppp/ppp.linkup`.
7. Aktualisieren Sie die Datei `/etc/rc.conf`.

28.3. Kernel-PPP

Teile wurden ursprünglich beigetragen von Gennady B. Sorokopud und Robert Huff.

Warnung: Der folgende Abschnitt ist ausschließlich für FreeBSD 7.X relevant und gültig.

28.3.1. Einrichtung von Kernel-PPP

Bevor Sie PPP auf Ihrem Computer einrichten, sollten Sie dafür sorgen, dass `pppd` im Verzeichnis `/usr/sbin` vorhanden ist und `/etc/ppp` existiert.

`pppd` kann auf zweierlei Weise arbeiten:

1. Als “Client” – Sie möchten Ihren Rechner mit einem Netz verbinden, indem Sie eine serielle PPP-Verbindung aufbauen.
2.

Als “Server” – Ihr Rechner ist in ein Netzwerk eingebunden und stellt die PPP-Verbindung für andere Rechner im Netzwerk her.

In beiden Fällen werden Sie eine Datei mit den benötigten Optionen erstellen müssen (`/etc/ppp/options` oder, wenn mehr als ein Benutzer PPP verwendet, `~/.ppprc`).

Sie benötigen außerdem eine Software (vorzugsweise `comms/kermit`), mit der Sie seriell per Modem wählen und eine Verbindung zu dem entfernten Host aufbauen können.

28.3.2. Verwendung von `pppd` als Client

Basierend auf Informationen von Trev Roydhouse.

Die folgende Datei `/etc/ppp/options` kann für einen Verbindungsaufbau mit PPP zu einem Cisco Terminalserver verwendet werden.

```
crtscts         # enable hardware flow control
modem           # modem control line
noipdefault     # remote PPP server must supply your IP address
                # if the remote host does not send your IP during IPCP
                # negotiation, remove this option
passive         # wait for LCP packets
domain ppp.foo.com      # put your domain name here

:remote_ip      # put the IP of remote PPP host here
                # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to local_ip:remote_ip

defaultroute    # put this if you want that PPP server will be your
                # default router
```

Um eine Verbindung herzustellen, sollten Sie:

1. Mit **Kermit** (oder einem anderen Modemprogramm) den entfernten Host anwählen und Ihren Benutzernamen sowie Ihr Passwort (oder was sonst nötig ist, um PPP auf dem entfernten Host zu aktivieren) eingeben.
2. **Kermit** beenden (ohne die Verbindung abzubrechen).
3. Folgendes eingeben:

   ```
   # /usr/sbin/pppd /dev/tty01 19200
   ```

   Achten Sie darauf, dass sie eine geeignete Geschwindigkeit wählen und das richtige Device verwenden.

Nun ist Ihr Computer mit Hilfe von PPP verbunden. Wenn die Verbindung nicht funktionieren sollte, können Sie die Option `debug` in die Datei `/etc/ppp/options` eintragen und die Ausgaben auf der Konsole beobachten, um die Fehler zu finden.

Das folgende Skript `/etc/ppp/pppup` führt alle 3 Schritte automatisch aus:

```
#!/bin/sh
pgrep -l pppd
pid=‘pgrep pppd‘
if [ "X${pid}" != "X" ] ; then
```

```
        echo 'killing pppd, PID=' ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

`/etc/ppp/kermit.dial` ist ein **Kermit**-Skript das den Einwählvorgang und alle notwendigen Autorisationen auf dem entfernten Host durchführt (ein Beispiel für ein solches Skript ist im Anhang zu diesem Dokument zu finden).

Verwenden Sie das folgende Skript `/etc/ppp/pppdown`, um die PPP-Verbindung abzubrechen:

```
#!/bin/sh
pid=`pgrep pppd`
if [ X${pid} != "X" ] ; then
        echo 'killing pppd, PID=' ${pid}
        kill -TERM ${pid}
fi

pgrep -l kermit
pid=`pgrep kermit`
if [ "X${pid}" != "X" ] ; then
        echo 'killing kermit, PID=' ${pid}
        kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Prüfen Sie, ob **pppd** immer noch läuft, indem Sie `/usr/etc/ppp/ppptest` ausführen. Dieses Skript sollte folgendermaßen aussehen:

```
#!/bin/sh
pid=`pgrep pppd`
if [ X${pid} != "X" ] ; then
        echo 'pppd running: PID=' ${pid-NONE}
else
        echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

Um die Modemverbindung abzubrechen, können Sie das Skript `/etc/ppp/kermit.hup` verwenden, das Folgendes enthalten sollte:

```
set line /dev/tty01     ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
exit
```

Hier ist eine alternative Methode, bei der `chat` an Stelle von **Kermit** eingesetzt wird:

Die folgenden beiden Dateien reichen aus, um eine Verbindung über `pppd` herzustellen.

`/etc/ppp/options`:

```
/dev/cuad1 115200

crtscts                 # enable hardware flow control
modem          # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault    # remote PPP serve must supply your IP address
               # if the remote host doesn't send your IP during
                # IPCP negotiation, remove this option
passive         # wait for LCP packets
domain <your.domain>   # put your domain name here

:              # put the IP of remote PPP host here
               # it will be used to route packets via PPP link
                # if you didn't specified the noipdefault option
                # change this line to local_ip:remote_ip;

defaultroute   # put this if you want that PPP server will be
               # your default router
```

`/etc/ppp/login.chat.script`:

Anmerkung: Die folgenden Angaben sollten in einer Zeile stehen.

```
ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDTphone.number
```

```
  CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: login-id
  TIMEOUT 5 sword: password
```

Wenn diese Dateien richtig installiert und modifiziert sind, müssen Sie `pppd`, nur noch wie folgt starten:

```
# pppd
```

28.3.3. Verwendung von `pppd` als Server

`/etc/ppp/options` sollte etwa Folgendes enthalten:

```
crtscts                         # Hardware flow control
netmask 255.255.255.0           # netmask (not required)
192.114.208.20:192.114.208.165  # IP's of local and remote hosts
                                # local ip must be different from one
                                # you assigned to the Ethernet (or other)
                                # interface on your machine.
                                # remote IP is IP address that will be
                                # assigned to the remote machine
domain ppp.foo.com              # your domain
passive                         # wait for LCP
modem                           # modem line
```

Das folgende Skript `/etc/ppp/pppserv` lässt **pppd** als Server zu arbeiten:

```
#!/bin/sh
pgrep -l pppd
pid=‘pgrep pppd‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=‘pgrep kermit‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermit -y /etc/ppp/kermit.ans

# run ppp
pppd /dev/tty01 19200
```

Verwenden Sie das Skript`/etc/ppp/pppservdown`, um den Server zu beenden:

```
#!/bin/sh
```

```
pgrep -l pppd
pid=‘pgrep pppd‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing pppd, PID=’ ${pid}
        kill ${pid}
fi
pgrep -l kermit
pid=‘pgrep kermit‘
if [ "X${pid}" != "X" ] ; then
        echo ’killing kermit, PID=’ ${pid}
        kill -9 ${pid}
fi
ifconfig ppp0 down
ifconfig ppp0 delete

kermit -y /etc/ppp/kermit.noans
```

Mit dem **Kermit**-Skript (`/etc/ppp/kermit.ans`) lässt sich die Funktion Ihres Modems, automatisch zu antworten, ein- bzw. ausschalten. Es sollte folgendermaßen aussehen:

```
set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
inp 5 OK
echo \13
out ATS0=1\13   ; change this to out ATS0=0\13 if you want to disable
                ; autoanswer mode
inp 5 OK
echo \13
exit
```

Ein Skript namens `/etc/ppp/kermit.dial` wird für die Einwahl und Authentifizierung am entfernten Host verwendet. Sie müssen es noch an Ihre lokalen Gegebenheiten anpassen. Geben Sie in diesem Skript Ihren Benutzernamen und Ihr Passwort ein. In Abhängigkeit von der Reaktion Ihres Modems und des entfernten Hosts, werden Sie auch noch die `input` Anweisungen verändern müssen.

```
;
; put the com line attached to the modem here:
;
set line /dev/tty01
```

```
;
; put the modem speed here:
;
set speed 19200
set file type binary            ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto                ; Then SET CARRIER if necessary,
set dial display on             ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                       ; login prompt counter
goto slhup

:slcmd                          ; put the modem in command mode
echo Put the modem in command mode.
clear                           ; Clear unread characters from input buffer
pause 1
output +++                      ; hayes escape sequence
input 1 OK\13\10                ; wait for OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd              ; if modem doesn't answer OK, try again

:slhup                          ; hang up the phone
clear                           ; Clear unread characters from input buffer
pause 1
echo Hanging up the phone.
output ath0\13                  ; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd              ; if no OK answer, put modem in command mode

:sldial                         ; dial the number
pause 1
echo Dialing.
output atdt9,550311\13\10               ; put phone number here
assign \%x 0                    ; zero the time counter

:look
clear                           ; Clear unread characters from input buffer
increment \%x                   ; Count the seconds
```

```
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                        ; login
assign \%x 0                    ; zero the time counter
pause 1
echo Looking for login prompt.

:slloop
increment \%x                   ; Count the seconds
clear                           ; Clear unread characters from input buffer
output \13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop         ; try 10 times to get a login prompt
else goto slhup                 ; hang up and start again if 10 failures

:sluid
;
; put your userid here:
;
output ppp-login\13
input 1 {Password: }
;
; put your password here:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7No dialtone.  Check the telephone line!\7
exit 1

; local variables:
```

```
; mode: csh
; comment-start: "; "
; comment-start-skip: "; "
; end:
```

28.4. Probleme bei PPP-Verbindungen

Beigetragen von Tom Rhodes.

Warnung: Mit FreeBSD 8.0 wurde der sio(4)-Treiber durch den Treiber uart(4) ersetzt. Parallel dazu wurden auch die entsprechenden Gerätedateien für die seriellen Ports umbenannt: `/dev/cuadN` wurde zu `/dev/cuauN`, `/dev/ttydN` zu `/dev/ttyuN`. Verwenden Sie noch FreeBSD 7.X, müssen Sie dies beim Lesen der folgenden Abschnitte berücksichtigen.

Dieser Abschnitt behandelt Probleme, die auftauchen können, wenn PPP über ein Modem verwendet wird. Sie müssen beispielsweise genau die Eingabeaufforderung des Systems kennen, in das Sie sich einwählen. Einige ISPs verwenden `ssword` andere verwenden `password`; wenn das Einwahlskript falsch ist, scheitert die Anmeldung. Üblicherweise suchen Sie nach Fehlern der PPP-Verbindung indem Sie sich manuell verbinden. Wie das genau geht, wird im Folgenden gezeigt.

28.4.1. Gerätedateien überprüfen

Wenn Sie einen eigenen Kernel verwenden, stellen Sie sicher, dass die folgende Zeile in der Kernelkonfigurationsdatei vorhanden ist:

```
device   uart
```

Das `uart`-Gerät ist bereits im `GENERIC`-Kernel vorhanden, deshalb sind in diesem Fall keine zusätzlichen Schritte vonnöten. Kontrollieren Sie die Ausgabe von `dmesg`:

```
# dmesg | grep uart
```

In der Ausgabe sollten die entsprechenden `uart`-Geräte, beispielsweise `uart1` (`COM2`), angezeigt werden. Wird ein passendes Gerät angezeigt, brauchen Sie keinen neuen Kernel zu erstellen. Wenn Ihr Modem an `uart1` angeschlossen ist (in DOS ist dieser Anschluss als `COM2` bekannt), ist `/dev/cuau1` die dazugehörende Gerätedatei.

28.4.2. Manuelle Verbindungen

Ein Verbindungsaufbau zum Internet durch manuelle Steuerung von `ppp` geht schnell, ist einfach und stellt einen guten Weg dar, eine Verbindung auf Fehler hin zu überprüfen oder einfach Informationen darüber zu sammeln, wie Ihr ISP Verbindungen handhabt. Lassen Sie uns **PPP** von der Kommandozeile aus starten. Beachten Sie, dass in allen Beispielen *example* der Hostname der Maschine ist, auf der **PPP** läuft. Sie starten `ppp`, indem Sie einfach `ppp` eingeben:

```
# ppp
```

Wir haben ppp nun gestartet.

```
ppp ON example> set device /dev/cuau1
```

Wir geben das Device an, an das unser Modem angeschlossen ist. In diesem Fall ist es cuau1.

```
ppp ON example> set speed 115200
```

Wir geben die Verbindungsgeschwindigkeit an. Im Beispiel verwenden wir 115200 kbps

```
ppp ON example> enable dns
```

Wir weisen ppp an, unseren Resolver zu konfigurieren und in der Datei /etc/resolv.conf Einträge für den Nameserver hinzuzufügen. Falls ppp unseren Hostnamen nicht bestimmen kann, geben wir diesen später manuell an.

```
ppp ON example> term
```

Wir wechseln in den “Terminal”-Modus, um das Modem manuell kontrollieren zu können.

```
deflink: Entering terminal mode on /dev/cuau1
type '~h' for help

at
OK
atdt123456789
```

Sie verwenden at zur Initialisierung Ihres Modems und dann atdt sowie die Nummer Ihres ISP, um den Einwählprozess zu starten.

```
CONNECT
```

Dies ist die Bestätigung, dass eine Verbindung aufgebaut wurde. Falls wir Verbindungsprobleme bekommen, die nicht mit der Hardware zusammenhängen, werden wir an dieser Stelle ansetzen müssen, um eine Lösung zu finden.

```
ISP Login:myusername
```

Hier werden Sie nach einem Benutzernamen gefragt. Geben Sie am Prompt den Namen ein, den Ihnen Ihr ISP zur Verfügung gestellt hat.

```
ISP Pass:mypassword
```

An dieser Stelle müssen Sie das Passwort angeben, das Ihnen von Ihrem ISP vorgegeben wurde. Das Passwort wird, analog dem normalen Anmeldevorgang, nicht angezeigt.

```
Shell or PPP:ppp
```

Abhängig von Ihrem ISP, kann es sein, dass dieser Prompt bei Ihnen gar nicht erscheint. Wir werden hier gefragt, ob wir eine Shell beim Provider verwenden oder ppp starten wollen. Weil wir eine Internetverbindung aufbauen wollen, haben wir uns in diesem Beispiel für ppp entschieden.

```
Ppp ON example>
```

Beachten Sie, dass sich in diesem Beispiel das erste p in einen Großbuchstaben verwandelt hat. Dies zeigt, dass wir erfolgreich eine Verbindung zu unserem ISP hergestellt haben.

```
PPp ON example>
```

An dieser Stelle haben wir uns erfolgreich bei unserem ISP authentifiziert und warten darauf, dass uns eine IP-Adresse zugewiesen wird.

```
PPP ON example>
```

Wir haben uns mit der Gegenstelle auf eine IP-Adresse geeinigt und den Verbindungsaufbau erfolgreich abgeschlossen

```
PPP ON example> add default HISADDR
```

Hier geben wir unsere Standardroute an. Weil zu diesem Zeitpunkt unsere einzige Verbindung zu unserer Gegenstelle besteht, müssen wir dies tun, bevor wir Kontakt zu unserer Umwelt aufnehmen können. Falls dies aufgrund bestehender Routen nicht funktionieren sollte, können Sie ein Ausrufungszeichen `!` vor `add` setzen. Sie können diese Standardroute aber auch vor dem eigentlichen Verbindungsaufbau angeben und **PPP** wird entsprechend eine neue Route aushandeln.

Wenn alles gut ging, sollten wir nun eine aktive Internetverbindung haben, die wir mit **Ctrl**+**z** in den Hintergrund schicken können Wenn sie feststellen, dass `PPP` wieder zu `ppp` wird, ist die Verbindung abgebrochen. Es ist gut dies zu wissen, weil dadurch der Verbindungsstatus angezeigt wird. Große `Ps` zeigen an, dass wir eine Verbindung zum ISP haben und kleine `ps` zeigen an, dass wir aus irgendeinem Grund die Verbindung verloren haben. `ppp` hat nur diese beiden Zustände.

28.4.2.1. Fehlersuche

Wenn sie einen Direktanschluss haben und keine Verbindung aufbauen können, schalten Sie die Hardware-Flusssteuerung CTS/RTS aus, indem Sie die Option `set ctsrts off` verwenden. Dies ist zumeist dann der Fall, wenn Sie mit einem **PPP**-fähigen Terminalserver verbunden sind. Hier bleibt **PPP** bei dem Versuch hängen, Daten über Ihre Nachrichtenverbindung zu schicken, weil auf ein CTS-Signal (Clear-to-Send) gewartet wird, das nie kommt. Wenn Sie diese Option jedoch gebrauchen, sollten Sie auch die Option `set accmap` verwenden, die erforderlich sein kann, um bestimmte Hardware zu kontrollieren, die auf die Übertragung bestimmter Zeichen zwischen den Kommunikations-Endpunkten (zumeist XON/XOFF) angewiesen ist. Die Manual-Seite ppp(8) bietet mehr Informationen zu dieser Option und ihrer Verwendung.

Wenn Sie ein älteres Modem haben, benötigen Sie vielleicht die Option `set parity even`. Standardmäßig wird keine Parität vorausgesetzt, sie ist aber für die Fehlerprüfung bei älteren Modems und bei bestimmten ISPs erforderlich. Sie könnten diese Option für den ISP Compuserve benötigen.

PPP kehrt möglicherweise nicht in den Befehlsmodus zurück, was normalerweise auf einen Fehler bei der Aushandlung hinweist, wobei der ISP wartet, dass Ihre Seite den Aushandlungsprozess beginnt. Die Option `~p` erzwingt in diesem Fall den Beginn des Aushandlungsprozesses.

Wenn Sie nie einen Login-Prompt erhalten, müssen Sie statt des im Beispiel gezeigten UNIX-Stils höchst wahrscheinlich PAP oder CHAP für die Authentifizierung verwenden. Um PAP oder CHAP zu verwenden, ergänzen Sie **PPP** einfach um folgende Optionen, bevor Sie in den Terminalmodus wechseln:

```
ppp ON example> set authname myusername
```

Hierbei sollte *`myusername`* durch den Benutzernamen ersetzt werden, den Sie von Ihrem ISP bekommen haben.

```
ppp ON example> set authkey mypassword
```

mypassword sollten Sie durch das Passwort ersetzen, das Ihnen Ihr ISP gegeben hat.

Wenn die Verbindung aufgebaut wird, Sie aber keine Rechner unter ihrem Domänen-Namen erreichen können, versuchen Sie, einen Rechner mit ping(8) und seiner IP-Adresse zu erreichen. Wenn 100% der Pakete verloren gehen, ist es sehr wahrscheinlich, dass Ihnen keine Standardroute zugewiesen wurde. Überprüfen Sie, ob während des Verbindungsaufbaus die Option `add default HISADDR` gesetzt war. Wenn Sie zu einer entfernten IP-Adresse eine Verbindung aufbauen können, ist es möglich, dass die Adresse eines Nameservers nicht in die Datei `/etc/resolv.conf` eingetragen wurde. Diese Datei sollte folgendermaßen aussehen:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

Dabei sollten *x.x.x.x* und *y.y.y.y* durch die IP-Adressen der DNS-Server Ihres ISPs ersetzt werden. Diese Information ist Ihnen bei Vertragsabschluss mitgeteilt worden. Wenn nicht, sollte ein Anruf bei Ihrem ISP Abhilfe schaffen.

Mit syslog(3) können Sie Ihre **PPP**-Verbindung protokollieren. Fügen Sie einfach die folgende Zeile in `/etc/syslog.conf` ein:

```
!ppp
*.*     /var/log/ppp.log
```

In den meisten Fällen existiert diese Funktionalität bereits.

28.5. PPP over Ethernet (PPPoE)

Beigetragen (durch http://node.to/freebsd/how-tos/how-to-freebsd-pppoe.html) von Jim Mock.

Dieser Abschnitt beschreibt, wie Sie PPP over Ethernet (PPPoE) einrichten.

28.5.1. Konfiguration des Kernels

Eine besondere Kernelkonfiguration ist für PPPoE nicht mehr erforderlich. Sofern die notwendige NetGraph-Unterstützung nicht in den Kernel eingebaut wurde, wird diese von **ppp** dynamisch geladen.

28.5.2. Einrichtung von `ppp.conf`

Dies hier ist ein Beispiel einer funktionierenden `ppp.conf`:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your Ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
```

```
set login
add default HISADDR
```

28.5.3. ppp ausführen

Als `root`, geben Sie ein:

```
# ppp -ddial name_of_service_provider
```

28.5.4. ppp beim Systemstart ausführen

Fügen Sie Folgendes in Ihre Datei `/etc/rc.conf` ein:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"  # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

28.5.5. Verwendung einer PPPoE-Dienstbezeichnung (service tag)

Manchmal kann es notwendig sein, eine Dienstbezeichnung (*service tag*) zu verwenden, um eine Verbindung aufzubauen. Dienstbezeichnungen werden eingesetzt, um zwischen verschiedenen PPPoE-Servern unterscheiden zu können, die einem bestehenden Netzwerk zugeteilt sind.

Die erforderlichen Dienstbezeichnungen sollten in der Dokumentation, zu finden sein, die Ihnen Ihr ISP zur Verfügung gestellt hat. Wenn Sie diese Informationen dort nicht finden, fragen Sie beim technischen Kundendienst Ihres ISP danach.

Als letzte Möglichkeit, bleibt die Methode, die von dem Programm Roaring Penguin PPPoE (http://www.roaringpenguin.com/pppoe/) vorgeschlagen wird, das in der Ports-Sammlung zu finden ist. Bedenken Sie aber, dass dadurch Daten Ihres Modems gelöscht werden können, so dass es nicht mehr benutzt werden kann. Überlegen Sie also genau, ob Sie dies machen wollen. Installieren Sie einfach das Programm, das Ihnen Ihr Provider zusammen mit dem Modem geliefert hat. Gehen Sie dann in das Menü System dieses Programms. Der Name Ihres Profils, sollte in der Liste aufgeführt sein. Normalerweise ist dies *ISP*.

Der Name des Profils (*service tag*) wird im Eintrag für die PPPoE-Konfiguration in der Datei `ppp.conf` verwendet, als der Teil des Befehls `set device` (die manpage ppp(8) enthält Einzelheiten hierzu), der den Provider angibt. Dieser Eintrag sollte folgendermaßen aussehen:

```
set device PPPoE:xl1:ISP
```

Vergessen Sie nicht, statt *`xl1`* das richtige Device Ihrer Netzwerkkarte anzugeben.

Denken sie auch daran, *`ISP`* durch das Profil, das Sie oben gefunden haben zu ersetzen.

Weitere Informationen bieten:

- Cheaper Broadband with FreeBSD on DSL (http://renaud.waldura.com/doc/freebsd/pppoe/) von Renaud Waldura.

28.5.6. PPPoE mit einem 3Com® HomeConnect® ADSL Modem Dual Link

Dieses Modem folgt nicht dem RFC 2516 (http://www.faqs.org/rfcs/rfc2516.html) (*A Method for transmitting PPP over Ethernet (PPPoE)*, verfasst von L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, und R. Wheeler). Stattdessen wurden andere Pakettyp-Codes für die Ethernet Frames verwendet. Bitte beschweren Sie sich unter 3Com (http://www.3com.com/), wenn Sie der Ansicht sind, dass dieses Modem die PPPoE-Spezifikation einhalten sollte.

Um FreeBSD in die Lage zu versetzen, mit diesem Gerät zu kommunizieren, muss ein sysctl Befehl angegeben werden. Dies kann beim Systemstart automatisch geschehen, indem die Datei `/etc/sysctl.conf` angepasst wird:

```
net.graph.nonstandard_pppoe=1
```

oder, wenn der Befehl unmittelbar wirksam werden soll, durch:

```
# sysctl net.graph.nonstandard_pppoe=1
```

Da hiermit eine systemweit gültige Einstellung vorgenommen wird, ist es nicht möglich, gleichzeitig mit einem normalen PPPoE-Client oder Server und einem 3Com HomeConnect® ADSL Modem zu kommunizieren.

28.6. PPP over ATM (PPPoA)

Nachfolgend wird beschrieben, wie PPP over ATM (PPPoA) eingerichtet wird. PPPoA ist vor allem unter europäischen DSL-Providern populär.

28.6.1. Der Einsatz von PPPoA mit dem Alcatel SpeedTouch™ USB

PPPoA-Unterstützung für dieses Gerät ist unter FreeBSD als Port verfügbar, da die Firmware unter Alcatels Lizenzvereinbarung (http://www.speedtouchdsl.com/disclaimer_lx.htm) vertrieben wird und deshalb nicht mit dem FreeBSD-Basissystem frei verteilt werden kann.

Um die Software zu installieren, verwenden Sie einfach die Ports-Sammlung. Installieren Sie den Port `net/pppoa` und folgen Sie den dabei angegebenen Instruktionen.

Für den ordnungsgemäßen Betrieb muss das Alcatel SpeedTouch™ USB, wie viele USB-Geräte, Firmware auf den Gastrechner laden. FreeBSD kann die Firmware automatisch laden, wenn das Gerät mit dem USB-Anschluss verbunden wird. Dazu fügen Sie als Benutzer `root` die nachstehenden Zeilen in `/etc/usbd.conf` ein:

```
device "Alcatel SpeedTouch USB"
    devname "ugen[0-9]+"
    vendor 0x06b9
    product 0x4061
    attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

Den USB-Dæmon aktivieren Sie mit der folgenden Zeile in `/etc/rc.conf`:

```
usbd_enable="YES"
```

Wenn die Verbindung beim Start von **ppp** aufgebaut werden soll, fügen Sie die nachstehenden Zeilen als Benutzer `root` in `/etc/rc.conf` ein:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

Verwenden Sie bitte diese Einstellungen zusammen mit der Beispielkonfiguration in `ppp.conf` des Ports `net/pppoa`.

28.6.2. Die Verwendung von mpd

Sie können **mpd** verwenden, um zu einer Reihe von Diensten, insbesondere PPTP-Diensten eine Verbindung herzustellen. Sie finden **mpd** in der Ports-Sammlung unter `net/mpd`. Viele ADSL Modems, wie das Alcatel SpeedTouch Home, sind auf einen PPTP-Tunnel zwischen dem Modem und dem Rechner angewiesen.

Zuerst müssen Sie den Port installieren, um danach **mpd** entsprechend Ihren Anforderungen und den Vorgaben Ihres Providers konfigurieren zu können. Der Port installiert auch einige gut dokumentierte Beispielkonfigurationsdateien in *`PREFIX`*`/etc/mpd/`. Beachten Sie, dass *`PREFIX`* hier das Verzeichnis angibt, in das Ihre Ports installiert werden. Standardmäßig ist dies das Verzeichnis `/usr/local/`. Ein kompletter Leitfaden zur Konfiguration von **mpd** ist im HTML-Format verfügbar, sobald der Port installiert ist. Dieser ist in *`PREFIX`*`/share/doc/mpd/` zu finden. Hier ist eine Beispielkonfiguration, um mit **mpd** eine Verbindung zu einem ADSL-Dienst aufzubauen. Die Konfiguration ist auf zwei Dateien verteilt. Zunächst die Datei `mpd.conf`:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ❶
    set bundle password password ❷
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

❶ Der Benutzername, den Sie zur Authentifizierung bei Ihrem ISP verwenden.

❷ Das Passwort, das Sie zur Authentifizierung bei Ihrem ISP verwenden.

Die Datei `mpd.links` enthält Informationen über die Verbindung(en), die Sie aufbauen möchten. Eine Beispieldatei `mpd.links`, die das vorige Beispiel ergänzt, wird unten angegeben:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ❶
    set pptp peer 10.0.0.138 ❷
```

❶ Die IP-Adresse des FreeBSD-Rechners von dem aus Sie **mpd** verwenden.

❷ Die IP-Adresse des ADSL-Modems. Das Alcatel SpeedTouch Home hat die Adresse `10.0.0.138` voreingestellt.

Ein Verbindungsaufbau kann einfach durch Eingabe des folgenden Befehls als `root` gestartet werden:

```
# mpd -b adsl
```

Sie können sich den Status der Verbindung durch folgenden Befehl anzeigen lassen:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Die Verwendung von **mpd** ist der empfehlenswerteste Weg, um mit FreeBSD eine Verbindung zu einem ADSL-Dienst aufzubauen.

28.6.3. Die Verwendung von pptpclient

Es ist außerdem möglich, mit FreeBSD eine Verbindung zu anderen PPPoA-Diensten aufzubauen. Dazu wird `net/pptpclient` verwendet.

Um mit `net/pptpclient` eine eine Verbindung zu einem DSL-Dienst aufbauen zu können, müssen Sie den entsprechenden Port bzw. das Paket installieren und die Datei `/etc/ppp/ppp.conf` bearbeiten. Sie müssen `root` sein, um diese Schritte durchführen zu können. Eine Beispieldatei für `ppp.conf` ist weiter unten angegeben. Weitere Informationen zu den Optionen von `ppp.conf` bietet die Manual-Seite **ppp** ppp(8):

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname username ❶
 set authkey password ❷
 set ifaddr 0 0
 add default HISADDR
```

❶ Der Benutzername für den Zugang zu den Diensten Ihres ISP.

❷ Das Passwort für Ihren Account.

> **Warnung:** Weil Sie Ihr Passwort in der Datei `ppp.conf` in Klartext angeben müssen, sollten Sie sicherstellen, dass niemand den Inhalt dieser Datei lesen kann. Die folgende Reihe von Befehlen stellt sicher, dass die Datei nur von `root` lesbar ist. Zusätzliche Informationen bieten die Manual-Seiten chmod(1) und chown(8):

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Dies wird einen Tunnel für eine PPP-Session zu Ihrem DSL-Router öffnen. Ethernet-DSL-Modems haben eine vorkonfigurierte LAN-IP-Adresse, mit der Sie eine Verbindung aufbauen. Im Falle des Alcatel SpeedTouch Home handelt es sich dabei um die Adresse `10.0.0.138`. In der Dokumentation Ihres Routers sollte angegeben sein, welche Adresse Ihr Gerät verwendet. Um den Tunnel zu öffnen und eine PPP-Session zu starten, führen Sie bitte folgenden Befehl aus:

```
# pptp address adsl
```

> **Tipp:** Vielleicht möchten Sie ein kaufmännisches Und (“&”) an das Ende oben angegebenen Kommandos anfügen, da **pptp** sonst den Prompt nicht zurückgibt.

Ein virtuelles Tunnel-Device `tun` wird für das Zusammenspiel der Prozesse **pptp** und **ppp** geschaffen. Wenn Sie den Prompt zurückerhalten haben oder der **pptp**-Prozess das Vorliegen einer Verbindung bestätigt, können Sie den Tunnel folgendermaßen überprüfen:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
        Opened by PID 918
```

Wenn Sie nicht in der Lage sein sollten, eine Verbindung aufzubauen, überprüfen Sie die Konfiguration Ihres Routers, den Sie normalerweise per **telnet** oder mit einem Web-Browser erreichen können. Falls dennoch keine Verbindung zustande kommt, sollten Sie die Ausgabe des Befehls `pptp` und die Logdatei `/var/log/ppp.log` von **ppp** nach Hinweisen auf die Ursache durchsuchen.

28.7. SLIP

Ursprünglich beigetragen von Satoshi Asami. Mit Beiträgen von Guy Helmer und Piero Serini.

> **Warnung:** Der folgende Abschnitt ist ausschließlich für FreeBSD 7.X relevant und gültig.

28.7.1. Einrichtung eines SLIP-Clients

Im Folgenden wird ein Weg beschrieben, SLIP auf einer FreeBSD-Maschine für ein Netzwerk mit festen Hostnamen einzurichten. Bei einer dynamischen Zuweisung des Hostnamens (das heißt wenn sich Ihre Adresse bei jeder Einwahl ändert) wird die Einrichtung wahrscheinlich etwas komplexer aussehen.

Bestimmen Sie zuerst, an welcher seriellen Schnittstelle Ihr Modem angeschlossen ist. Viele Leute erzeugen einen symbolischen Link, wie etwa `/dev/modem`, der auf den wirklichen Gerätenamen `/dev/cuadN` verweist. Damit ist es Ihnen möglich, vom eigentlichen Gerätenamen zu abstrahieren, sollten Sie das Modem einmal an eine andere

Schnittstelle anschließen müssen. Es kann ziemlich umständlich sein, wenn Sie eine viele Dateien in `/etc` und `.kermrc`-Dateien, die über das ganze System verstreut sind, anpassen müssen!

> **Anmerkung:** `/dev/cuau0` ist `COM1`, `/dev/cuau1` ist `COM2`, etc.

Stellen Sie sicher, dass Folgendes in Ihrer Kernelkonfigurationsdatei steht:

```
device   sl      1
```

Dieses pseudo-device ist im `GENERIC` Kernel enthalten. Falls es von Ihnen nicht gelöscht wurde, sollten Sie hier kein Problem haben.

28.7.1.1. Dinge, die Sie nur einmal erledigen müssen

1. Tragen Sie Ihren lokalen Rechner, das Gateway, sowie die Nameserver in Ihre Datei `/etc/hosts` ein. Diese Datei sieht bei mir so aus:

   ```
   127.0.0.1               localhost loghost
   136.152.64.181          water.CS.Example.EDU water.CS water
   136.152.64.1            inr-3.CS.Example.EDU inr-3 slip-gateway
   128.32.136.9            ns1.Example.EDU ns1
   128.32.136.12           ns2.Example.EDU ns2
   ```

2. Vergewissern Sie sich, dass in der Datei `/etc/host.conf` im Abschnitt `hosts: files` vor `dns` steht. Ohne diese Reihenfolge könnten lustige Dinge passieren.

3. Editieren Sie die Datei `/etc/rc.conf`.

   1. Ihren Hostnamen geben Sie an, indem Sie folgende Zeile bearbeiten:

      ```
      hostname="myname.my.domain"
      ```

      Hier sollte der vollständige Internethostname Ihres Rechners angegeben werden.

   2. Den Defaultrouter geben Sie durch die Modifikation folgender Zeile an:

      ```
      defaultrouter="NO"
      ```

      wird zu:

      ```
      defaultrouter="slip-gateway"
      ```

4. Erstellen Sie die Datei `/etc/resolv.conf`, die Folgendes enthält:

   ```
   domain CS.Example.EDU
   nameserver 128.32.136.9
   nameserver 128.32.136.12
   ```

   Wie Sie sehen, werden hiermit die Nameserver angegeben. Natürlich hängen die tatsächlichen Domainnamen und Adressen von Ihren Gegebenheiten ab.

5. Legen Sie ein Passwort für `root` und `toor` (sowie für alle anderen Accounts die kein Passwort haben) fest.

6. Starten Sie Ihren Rechner neu und überprüfen Sie, ob er mir dem richtigen Hostnamen startet.

28.7.1.2. Aufbau einer SLIP-Verbindung

1. Wählen Sie sich ein, geben Sie `slip` und am Prompt den Namen Ihres Rechners sowie Ihr Passwort ein. Was Sie eingeben müssen, hängt von Ihren Gegebenheiten ab. Wenn Sie **Kermit** verwenden, können Sie ein Skript ähnlich dem Folgenden verwenden:

   ```
   # kermit setup
   set modem hayes
   set line /dev/modem
   set speed 115200
   set parity none
   set flow rts/cts
   set terminal bytesize 8
   set file type binary
   # The next macro will dial up and login
   define slip dial 643-9600, input 10 =>, if failure stop, -
   output slip\x0d, input 10 Username:, if failure stop, -
   output silvia\x0d, input 10 Password:, if failure stop, -
   output ***\x0d, echo \x0aCONNECTED\x0a
   ```

   Natürlich müssen Sie hier Ihren Benutzernamen und Ihr Passwort eintragen. Wenn Sie das getan haben, können Sie am **Kermit**-Prompt einfach `slip` eingeben, um sich zu verbinden.

   > **Anmerkung:** Es ist generell eine *schlechte* Idee, Ihr Passwort in einer unverschlüsselten Textdatei irgendwo im Dateisystem zu speichern. Tun Sie dies auf Ihr eigenes Risiko.

2. Belassen Sie **Kermit** so (Sie können es mit **Ctrl**-**z** unterbrechen) und geben Sie als `root` ein:

   ```
   # slattach -h -c -s 115200 /dev/modem
   ```

   Wenn Sie mit `ping` Hosts auf der anderen Seite des Routers erreichen können, sind Sie verbunden! Wenn es nicht funktionieren sollte, können Sie versuchen `-a` statt `-c` als Argument für `slattach` zu verwenden.

28.7.1.3. Beenden der Verbindung

Um `slattach` zu beenden, geben Sie Folgendes ein:

```
# kill -INT ‘cat /var/run/slattach.modem.pid‘
```

Beachten Sie, dass Sie `root` sein müssen, um dies durchführen zu können. Kehren Sie zu `kermit` zurück (mit Hilfe von `fg`, wenn Sie es unterbrochen haben) und beenden Sie dieses Programm (**q**).

slattach(8) gibt an, dass `ifconfig sl0 down` verwendet werden soll, um das Interface zu deaktivieren, doch das scheint keinen Unterschied zu machen. (`ifconfig sl0` gibt dasselbe aus).

Es kann vorkommen, dass Ihr Modem sich weigert, das Trägersignal zu beenden. In diesem Fall starten Sie `kermit` einfach neu und beenden es wieder. Beim zweiten Versuch geht es meist aus.

28.7.1.4. Lösungen bei Problemen

Wenn es nicht funktionieren sollte, können Sie an die Mailingliste freebsd-net (http://lists.FreeBSD.org/mailman/listinfo/freebsd-net) schreiben. Über diese Dinge sind Benutzer bisher gestolpert:

- Nicht `-c` oder `-a` in `slattach` verwenden (Das sollte nicht entscheidend sein, aber einige Benutzer haben berichtet, dass dies ihre Probleme löst).
- Verwendung von `s10` statt `sl0` (bei einigen Schriftarten kann der Unterschied schwer zu erkennen sein).
- Probieren Sie `ifconfig sl0`, um den Status Ihrer Schnittstelle abzufragen. Das Ergebnis könnte beispielsweise so aussehen:

  ```
  # ifconfig sl0
  sl0: flags=10<POINTOPOINT>
          inet 136.152.64.181 --> 136.152.64.1 netmask ffffff00
  ```

- Wenn ping(8) die Fehlermeldung `no route to host` ausgibt, kann die Routingtabelle falsch sein. Die Routen können Sie sich mit dem Kommando `netstat -r` ansehen:

  ```
  # netstat -r
  Routing tables
  Destination      Gateway            Flags     Refs     Use  IfaceMTU    Rtt    Netmasks:

  (root node)
  (root node)

  Route Tree for Protocol Family inet:
  (root node) =>
  default          inr-3.Example.EDU  UG          8   224515  sl0 -      -
  localhost.Exampl localhost.Example. UH          5    42127  lo0 -       0.438
  inr-3.Example.ED water.CS.Example.E UH          1        0  sl0 -      -
  water.CS.Example localhost.Example. UGH        34 47641234  lo0 -       0.438
  (root node)
  ```

  Die Zahlen im Beispiel stammen von einer recht ausgelasteten Maschine. Die Zahlen auf Ihrem System werden, je nach Netzaktivität, von den gezeigten abweichen.

28.7.2. Einrichtung eines SLIP-Servers

Dieses Dokument bietet Empfehlungen, wie Sie Ihr FreeBSD-System als SLIP-Server einrichten. Typischerweise bedeutet dies, Ihr System so zu, konfigurieren, dass beim Login automatisch eine Verbindung für entfernte SLIP-Clients aufgebaut wird.

28.7.2.1. Voraussetzungen

Dieser Abschnitt ist ausgesprochen technischer Natur, weshalb Hintergrundwissen erforderlich ist. Wir gehen davon aus, dass Sie mit dem TCP/IP Protokoll, insbesondere mit Netzwerk- und Rechneradressierung, Netzwerkmasken, Subnetzen, Routing und Routingprotokollen, wie RIP, vertraut sind. Die Konfiguration von SLIP-Diensten auf einem Einwählserver erfordert die Kenntnis dieser Konzepte. Wenn Sie damit nicht vertraut sein sollten, lesen Sie bitte Craig Hunt's *TCP/IP Network Administration* publiziert von O'Reilly & Associates, Inc. (ISBN Nummer 0-937175-82-X) oder die Bücher von Douglas Comer über das TCP/IP Protokoll.

Wir gehen außerdem davon aus, dass Sie Ihr(e) Modem(s) eingerichtet haben und die entsprechenden Systemdateien so konfiguriert haben, dass Logins durch Ihr Modem zugelassen sind. Wenn Sie Ihr System dafür noch nicht vorbereitet haben, lesen Sie bitte Abschnitt 27.4, um Ihre Einwahlverbindung zu konfigurieren. Hilfreich sind auch die Manualpages sio(4) mit Informationen zum Gerätetreiber der seriellen Schnittstelle ttys(5), sowie gettytab(5), getty(8) und init(8) für Informationen zur Konfiguration von Logins über ein Modem. stty(1) bietet Informationen zur Einstellung der Parameter der seriellen Schnittstelle (etwa von `clocal` für direkt angeschlossene serielle Geräte).

28.7.2.2. Ein kurzer Überblick

Mit der normal verwendeten Konfiguration funktioniert der FreeBSD-SLIP-Server folgendermaßen: Ein SLIP-Benutzer wählt einen FreeBSD-SLIP-Server an und meldet sich mit einer speziellen SLIP-Login-ID ein, wobei `/usr/sbin/sliplogin` als Shell dieses besonderen Accounts dient. Das Programm `sliplogin` durchsucht die Datei `/etc/sliphome/slip.hosts` nach einer passenden Zeile für diesen Account. Falls ein Treffer erzielt wird, verbindet es den seriellen Anschluss mit einem verfügbaren SLIP-Interface und führt das Shellskript `/etc/sliphome/slip.login` aus, um das SLIP-Interface zu konfigurieren.

28.7.2.2.1. Ein Beispiel für ein Login eines SLIP-Servers

Wenn beispielsweise die Kennung eines SLIP-Benutzers, `Shelmerg` wäre, könnte der Eintrag des Benutzers `Shelmerg` in der Datei `/etc/master.passwd` etwa so aussehen:

```
Shelmerg:password:1964:89::0:0:Guy Helmer - SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

Wenn sich `Shelmerg` anmeldet, wird `sliplogin` die Datei `/etc/sliphome/slip.hosts` nach einer übereinstimmenden Benutzerkennung durchsuchen. So könnte etwa folgende Zeile in `/etc/sliphome/slip.hosts` stehen:

```
Shelmerg        dc-slip sl-helmer       0xffffc00               autocomp
```

`sliplogin` wird die passende Zeile finden, den seriellen Anschluss mit dem nächsten verfügbaren SLIP-Interface verbinden und dann `/etc/sliphome/slip.login` wie hier dargestellt ausführen:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xffffc00 autocomp
```

Wenn alles gut läuft, wird `/etc/sliphome/slip.login` ein `ifconfig` für das SLIP-Interface durchführen, mit dem sich `sliplogin` verbunden hat (in obigem Beispiel ist das slip 0, der als erster Parameter in der Liste an `slip.login` übergeben wurde), um die lokale IP-Adresse (`dc-slip`), die entfernte IP-Adresse (`sl-helmer`), die Netzmaske des SLIP-Interface (`0xffffc00`) und alle zusätzlichen Optionen (`autocomp`) festzulegen. Wenn etwas schief laufen sollte, bietet, `sliplogin` normalerweise informative Meldungen durch den **syslogd**-Daemon, der die Meldungen standardmäßig nach `/var/log/messages` schreibt (sehen Sie hierzu auch in den Manual-Seiten für syslogd(8) und syslog.conf(5) nach). Überprüfen Sie auch `/etc/syslog.conf`, um zu sehen, was `syslogd` aufzeichnet und wo es aufgezeichnet wird.

28.7.2.3. Kernelkonfiguration

Der Standardkernel von FreeBSD (`GENERIC`) bietet bereits SLIP-Unterstützung (sl(4)). Falls Sie einen angepassten Kernel verwenden, müssen Sie sicherstellen, dass Ihre Kernelkonfigurationsdatei folgende Zeile enthält:

```
device   sl
```

In der Voreinstellung leitet Ihr FreeBSD-Rechner keine Pakete weiter. Wenn Sie Ihren FreeBSD-SLIP-Server als Router einsetzen möchten, müssen Sie die Datei `/etc/rc.conf` bearbeiten und den Wert der Variable `gateway_enable` auf `YES` setzen. Dadurch ist sichergestellt, dass die Routingoptionen auch nach einem Neustart erhalten bleiben.

Um die Einstellungen sofort anzuwenden, führen Sie den folgenden Befehl als `root`-Benutzer aus:

```
# /etc/rc.d/routing start
```

Weitere Informationen zur Konfiguration Ihres Kernels, finden Sie in Kapitel 9 dieses Handbuches.

28.7.2.4. Konfiguration des Sliplogin

Wie bereits erwähnt, gibt es im Verzeichnis `/etc/sliphome` drei Dateien, die Teil der Konfiguration für `/usr/sbin/sliplogin` sind (`sliplogin` ist in sliplogin(8) beschrieben): `slip.hosts`, definiert die SLIP-Benutzer sowie deren IP-Adresse; `slip.login`, womit normalerweise nur das SLIP-Interface konfiguriert wird und (optional) `slip.logout`, womit die Auswirkungen von `slip.login` rückgängig gemacht werden, wenn die serielle Verbindung beendet wird.

28.7.2.4.1. Konfiguration der Datei `slip.hosts`

`/etc/sliphome/slip.hosts` enthält Zeilen, die mindestens vier durch Leerzeichen getrennte Elemente enthalten:

- Login-Kennung des SLIP-Benutzers
- Lokale Adresse (lokal für den SLIP-Server) der SLIP-Verbindung
- Entfernte Adresse der SLIP-Verbindung
- Netzwerkmaske

Die lokalen und entfernten Adressen können Hostnamen sein, deren zugehörige IP-Adresse durch die Datei `/etc/hosts` oder mithilfe des Domain Name Service aufgelöst wird. Wie die Adressen aufgelöst werden, hängt von den Einstellungen in `/etc/nsswitch.conf` ab. Die Netzwerkmaske kann ein Name sein, der durch eine Suche in `/etc/networks` aufgelöst werden kann. Auf einem Beispielsystem, würde die Datei `/etc/sliphome/slip.hosts` folgendermaßen aussehen:

```
#
# login local-addr      remote-addr     mask            opt1    opt2
#                                               (normal,compress,noicmp)
#
Shelmerg  dc-slip       sl-helmerg      0xfffffc00      autocomp
```

Am Ende der Zeile stehen eine oder mehrere der folgenden Optionen.

- `normal` – keine Header-Kompression
- `compress` – Header werden komprimiert
- `autocomp` – Header werden komprimiert, sofern die Gegenstelle es erlaubt
- `noicmp` – ICMP-Pakete werden deaktiviert (“ping” Pakete werden unterdrückt, statt die Ihnen zur Verfügung stehende Bandbreite aufzubrauchen)

Die Auswahl von lokalen und entfernten Adressen für Ihre SLIP-Verbindung, hängt davon ab, ob Sie ein TCP/IP-Subnetz reservieren oder ob Sie “proxy ARP” auf Ihrem SLIP-Server verwenden (es handelt sich nicht um “echtes” proxy ARP, aber dieser Begriff wird in diesem Abschnitt verwendet, um diesen Sachverhalt zu beschreiben). Wenn Sie nicht sicher sind, welche Methode Sie wählen sollen oder wie IP-Adressen zugewiesen werden, lesen Sie bitte in den Büchern zum Thema TCP/IP nach, die als Voraussetzungen für SLIP (Abschnitt 28.7.2.1) angegeben worden sind oder fragen Sie Ihren IP-Netzwerkadministrator.

Wenn Sie für Ihre SLIP-Clients ein eigenes Subnetz verwenden, werden Sie die Nummer des Subnetzes aus der Ihnen zugewiesenen IP-Netzwerknummer zuteilen und die IP-Adressen Ihrer SLIP-Clients aus diesem Subnetz verwenden müssen. Dann können Sie eine statische Route zu Ihrem SLIP-Subnetz über Ihren SLIP-Server auf Ihren nächsten IP-Router konfigurieren.

Wenn Sie aber andererseits die “proxy ARP” Methode verwenden möchten, werden Sie die IP-Adressen Ihrer SLIP-Clients aus dem Subnetz Ihres SLIP-Server nehmen und die Skripte `/etc/sliphome/slip.login` `/etc/sliphome/slip.logout` anpassen müssen, damit diese arp(8) zur Verwaltung der “proxy ARP”-Einträge in der ARP-Tabelle Ihres SLIP-Servers verwenden.

28.7.2.4.2. Konfiguration von `slip.login`

Eine typische Datei `/etc/sliphome/slip.login` sieht folgendermaßen aus:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

Diese `slip.login` Datei führt lediglich `ifconfig` für das entsprechende SLIP-Interface mit den lokalen und entfernten Adressen und der Netzwerkmaske des SLIP-Interface aus.

Wenn Sie sich dafür entschieden haben, die “proxy ARP” Methode zu verwenden (statt eines separaten Subnetzes für Ihre SLIP-Clients) sollte Ihre Datei `/etc/sliphome/slip.login` etwa folgendermaßen aussehen:

```
#!/bin/sh -
#
#       @(#)slip.login  5.1 (Berkeley) 7/1/90

#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

Die zusätzliche Zeile `arp -s $5 00:11:22:33:44:55 pub` in der Datei `slip.login` erzeugt einen ARP-Eintrag in der ARP-Tabelle des SLIP-Servers. Dieser ARP-Eintrag veranlasst den SLIP-Server mit seiner Ethernet MAC-Adresse zu antworten, sobald ein anderer IP-Knoten im Ethernet mit der IP-Adresse des SLIP-Clients Kontakt aufnehmen möchte.

Wenn Sie das Beispiel von oben verwenden, achten Sie darauf die Ethernet MAC-Adresse (`00:11:22:33:44:55`) durch die MAC-Adresse der Ethernetkarte Ihres Systems zu ersetzen. Sonst wird Ihr "proxy ARP" sicher nicht funktionieren! Sie können die MAC-Adresse Ihres SLIP-Servers herausfinden, indem Sie sich die Ausgabe von `netstat -i` ansehen. Die zweite Zeile der Ausgabe sollte ungefähr aussehen wie diese hier:

```
ed0   1500  <Link>0.2.c1.28.5f.4a         191923        0   129457     0   116
```

Dies zeigt an, dass die Ethernet MAC-Adresse dieses Systems `00:02:c1:28:5f:4a` lautet. Die Punkte in der Ethernet MAC-Adresse, die von `netstat -i` ausgegeben wird, müssen durch Doppelpunkte ersetzt werden. Bei jeder einstelligen Hexadezimalzahl sollten außerdem führende Nullen hinzugefügt werden, um die Adresse in die Form zu bringen, die von arp(8) verlangt wird. Die Manual-Seite von arp(8) bietet hierzu eine vollständige Übersicht.

> **Anmerkung:** Wenn Sie die Dateien `/etc/sliphome/slip.login` und `/etc/sliphome/slip.logout` erstellen, müssen diese ausführbar gemacht werden (`chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`), da `sliplogin` auf deren Ausführbarkeit angewiesen ist.

28.7.2.4.3. Konfiguration von `slip.logout`

Die Datei`/etc/sliphome/slip.logout` ist nicht zwingend erforderlich (außer Sie verwenden "proxy ARP"), aber falls Sie diese Datei erzeugen möchten, ist hier ein Beispiel für ein grundlegendes `slip.logout` Skript:

```
#!/bin/sh -
#
#       slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

Wenn Sie "proxy ARP" einsetzen, muss `/etc/sliphome/slip.logout` den ARP-Eintrag für den SLIP-Client löschen:

```
#!/bin/sh -
#
#       @(#)slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1        2         3        4          5         6     7-n
#   slipunit ttyspeed loginname local-addr remote-addr mask opt-args
```

```
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
/usr/sbin/arp -d $5
```

`arp -d $5` löscht den ARP-Eintrag, den die “proxy ARP” `slip.login` hinzufügte, als der SLIP-Client sich eingeloggt hatte.

Es soll nochmals darauf hingewiesen werden, dass für die Datei `/etc/sliphome/slip.logout` das Ausführungs-Bit gesetzt werden muss, nachdem die Datei erstellt worden ist (z.B. `chmod 755 /etc/sliphome/slip.logout`).

28.7.2.5. Überlegungen zum Routing

Wenn Sie nicht die “proxy ARP” Methode benutzen, um Datenpakete zwischen Ihren SLIP-Clients und dem Rest Ihres Netzwerkes (oder vielleicht dem Internet) zu routen, werden Sie wahrscheinlich statische Routen zu Ihrem nächsten Standardrouter hinzufügen müssen, um Pakete aus dem Subnetz Ihres SLIP-Clients über Ihren SLIP-Server weiterzuleiten.

28.7.2.5.1. Statische Routen

Das Hinzufügen von statischen Routen zu Ihrem nächsten Standardrouter kann problematisch sein (oder unmöglich, wenn Sie nicht die erforderliche Berechtigung haben...). Wenn Sie in Ihrer Organisation ein Netzwerk mit mehreren Routern haben, müssen einige Router, wie etwa die von Cisco und Proteon hergestellten, nicht nur mit der statischen Route zum SLIP-Subnetz konfiguriert werden, sondern es muss ihnen auch mitgeteilt werden, über welche statischen Routen sie andere Router informieren sollen. Daher ist einiges an Fachwissen und Problemlösungskompetenz erforderlich, um auf statischen Routen basierendes Routing erfolgreich einzurichten.

Kapitel 29. Elektronische Post (E-Mail)

Ursprüglicher Text von Bill Lloyd. Neugeschrieben von Jim Mock. Übersetzt von Robert Drehmel.

29.1. Terminologie

Das Akronym *MTA* steht für *Mail Transfer Agent* was übersetzt “Mailübertragungs-Agent” bedeutet.

Während die Bezeichnung *Server-Dämon* die Komponente eines MTA benennt, die für eingehende Verbindungen zuständig ist, wird mit dem Begriff *Mailer* öfters die Komponente des MTA bezeichnet, die E-Mails versendet.

29.2. Übersicht

“Elektronische Post”, besser bekannt als E-Mail, ist eine der am weit verbreitetsten Formen der Kommunikation heutzutage. Dieses Kapitel bietet eine grundlegende Einführung in das Betreiben eines E-Mail-Servers unter FreeBSD. Ebenfalls wird der Versand und Empfang von E-Mails unter FreeBSD behandelt. Das Kapitel ist jedoch keine komplette Referenz und es werden viele wichtige Überlegungen außer Acht gelassen. Wenn Sie das Thema detaillierter betrachten möchten, werden Sie bei einem der exzellenten Bücher fündig, die in Anhang B aufgelistet sind.

Dieses Kapitel behandelt die folgenden Punkte:

- Welche Software-Komponenten beim Senden und Empfangen von elektronischer Post involviert sind.
- Wo sich grundlegende **sendmail** Konfigurationsdateien in FreeBSD befinden.
- Den Unterschied zwischen entfernten und lokalen Postfächern.
- Wie man Versender von Massennachrichten daran hindern kann, Ihren E-Mail-Server illegalerweise als Weiterleitung zu verwenden.
- Wie man den Standard-Mailer des Systems, **sendmail**, ersetzt.
- Wie man oft auftretende E-Mail-Server Probleme behebt.
- Wie E-Mails mit UUCP verschickt werden.
- Wie E-Mails über einen Relay verschickt werden.
- Wie E-Mails über eine Einwahlverbindung gehandhabt werden.
- Wie Sie die SMTP-Authentifizierung einrichten.
- Den Empfang und den Versand von E-Mails mithilfe von Programmen wie **mutt**.
- Wie E-Mails von einem entfernten Server mit POP oder IMAP abgeholt werden.
- Wie eingehende E-Mail automatisch gefiltert wird.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Ihre Netzwerk-Verbindung richtig einrichten. (Kapitel 32).
- Die DNS-Information für Ihren E-Mail-Server einstellen (Kapitel 30).
- Wissen, wie man zusätzliche Dritthersteller-Software installiert (Kapitel 5).

29.3. Elektronische Post benutzen

Fünf größere Teile sind am E-Mail-Austausch beteiligt: Das Benutzerprogramm, der Server-Dämon, DNS, ein entferntes oder lokales Postfach und natürlich der E-Mail-Server selbst.

29.3.1. Das Benutzerprogramm

Das beinhaltet Kommandozeilenprogramme wie **mutt**, **alpine**, **elm**, **mail** und Programme mit grafischer Benutzeroberfläche, wie **balsa** und **xfmail** um einige zu nennen, und "aufwändigere", wie WWW-Browser. Diese Programme geben die E-Mail-Transaktionen an den lokalen "E-Mail-Server", weiter, entweder über einen der verfügbaren Server-Dämonen oder eine TCP-Verbindung.

29.3.2. E-Mail-Server Dämon

FreeBSD enthält standardmäßig **sendmail**; es lassen sich aber auch andere E-Mail-Server Dämonen betreiben, beispielsweise

- **exim**,
- **postfix** oder
- **qmail**.

Der Server-Dämon hat üblicherweise zwei Funktionen: Er kümmert sich um das Empfangen von eingehenden E-Mails und stellt ausgehende E-Mails zu. Es ist *nicht* Aufgabe des Dämons, E-Mails über POP oder IMAP bereit zu stellen, noch Zugriffe auf das lokale Postfach `mbox` oder Verzeichnisse mit Postfächern zu gewähren. Dafür benötigen Sie einen zusätzlichen Dämon.

> **Warnung:** Alte Versionen von **sendmail** enthalten schwerwiegende Sicherheitslöcher, die einem Angreifer Zugriff auf Ihren Rechner verschaffen können. Um Sicherheitsprobleme zu umgehen, sollten Sie eine aktuelle **sendmail**-Version benutzen. Sie können auch einen anderen MTA aus der FreeBSD Ports-Sammlung benutzen.

29.3.3. E-Mail und DNS

Das Domain Name System (DNS) und sein Dämon `named` spielen eine große Rolle in der Auslieferung von E-Mails. Um E-Mails auszuliefern, fragt der Mail-Server-Dämon im DNS den Rechner ab, der E-Mails für das Zielsystem entgegennimmt. Der gleiche Vorgang läuft ab, wenn eine E-Mail von einem entfernten Server auf Ihrem Mail-Server zugestellt wird.

Im DNS werden Rechnernamen auf IP-Adressen abgebildet. Daneben werden spezielle Informationen für das Mail-System gespeichert, die *MX-Einträge* (*MX record*) genannt werden. Der MX-Eintrag (von *Mail eXchanger*) gibt an, welcher Rechner oder welche Rechner E-Mails für eine Domain annehmen. Existiert kein MX-Record für einen Rechner oder dessen Domain, werden E-Mails direkt an den Rechner zugestellt, vorausgesetzt der Rechner besitzt einen A-Eintrag, der den Rechnernamen auf seine IP-Adresse abbildet.

Mit dem Kommando host(1) können Sie die MX-Einträge für eine Domain abfragen:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled (pri=10) by mx1.FreeBSD.org
```

29.3.4. E-Mails empfangen

Der E-Mail-Server empfängt alle E-Mails für Ihre Domäne. Er speichert die E-Mails entweder im `mbox`-Format (die Vorgabe) oder im Maildir-Format. Die E-Mails können lokal mit Programmen wie mail(1) oder **mutt** gelesen werden. Mithilfe von Protokollen wie POP oder IMAP können die E-Mails auch von entfernten Rechnern gelesen werden. Wenn Sie die E-Mails direkt auf dem E-Mail-Server lesen möchten, wird kein POP- oder IMAP-Server gebraucht.

29.3.4.1. Auf entfernte Postfächer mit POP und IMAP zugreifen

Wenn Sie auf entfernte Postfächer zugreifen wollen, benötigen Sie den Zugang zu einem POP- oder IMAP-Server. Beide Protokolle bieten einen einfachen Zugriff auf entfernte Postfächer. IMAP besitzt allerdings einige Vorteile, unter anderem:

- IMAP kann sowohl Nachrichten auf einem entfernten Server speichern als auch von dort abholen.
- IMAP unterstützt gleichzeitig ablaufende Aktualisierungen.
- Da es nicht gleich die komplette Nachricht herunterlädt, ist IMAP über langsame Verbindungen sehr nützlich. Weiterhin können E-Mails auf dem Server durchsucht werden, was den Datenverkehr zwischen Clients und dem Server minimiert.

Wenn Sie einen POP- oder IMAP-Server installieren wollen, gehen Sie nach den folgenden Schritten vor:

1. Wählen Sie einen IMAP- oder POP-Server aus, der Ihre Anforderungen erfüllt. Die nachstehenden Server sind sehr bekannt:
   - **qpopper**
   - **teapop**
   - **imap-uw**
   - **courier-imap**
   - **dovecot**
2. Installieren Sie den ausgewählten POP- oder IMAP-Daemon aus der Ports-Sammlung.
3. Wenn erforderlich, passen Sie die Datei `/etc/inetd.conf` an, um den POP- oder IMAP-Server zu starten.

> **Warnung:** Beachten Sie, dass sowohl POP als auch IMAP Daten, wie den Benutzernamen und das Passwort, im Klartext übertragen. Wenn Sie die mit diesen Protokollen übertragenen Daten schützen wollen, können Sie die Sitzung mittels ssh(1) tunneln oder SSL verwenden. Tunneln von Sitzungen wird in Abschnitt 15.10.8 beschrieben und SSL wird in Abschnitt 15.8 dargestellt.

29.3.4.2. Auf lokale Postfächer zugreifen

Auf Postfächer können Sie lokal mithilfe spezieller Benutzerprogramme, die *Mail-User-Agents* (MUAs) genannt werden, zugreifen. Beispiele für solche Programme sind **mutt** oder mail(1).

29.3.5. Der E-Mail-Server

“E-Mail-Server” wird der Rechner genannt, welcher für die Zustellung und das Empfangen von E-Mails auf Ihrem Rechner oder vielleicht Ihrem Netzwerk zuständig ist.

29.4. sendmail-Konfiguration

Beigesteuert von Christopher Shumway.

sendmail(8) ist das standardmäßig in FreeBSD installierte Mailübertragungsprogramm (MTA). Die Aufgabe von **sendmail** ist es, E-Mails von E-Mail-Benutzerprogrammen (MUA) anzunehmen und diese zu den entsprechenden Mailern zu liefern, die in der Konfigurationsdatei definiert sind. **sendmail** kann auch Netzwerkverbindungen annehmen und E-Mails zu lokalen *Mailboxen* [1] oder anderen Programmen liefern.

sendmail benutzt folgende Konfigurationsdateien:

Dateiname	Funktion
`/etc/mail/access`	Datenbank, in der Zugriffsrechte auf **sendmail** verwaltet werden
`/etc/mail/aliases`	Mailbox Aliase
`/etc/mail/local-host-names`	Liste der Rechner für die **sendmail** E-Mails akzeptiert
`/etc/mail/mailer.conf`	Mailer Programmkonfiguration
`/etc/mail/mailertable`	Mailer Versand-Zuordnungstabelle
`/etc/mail/sendmail.cf`	Hauptkonfigurationsdatei für **sendmail**
`/etc/mail/virtusertable`	Virtuelle Benutzer und Domänen-Tabellen

29.4.1. `/etc/mail/access`

Die Zugriffsdatenbank bestimmt, welche(r) Rechner oder IP-Adresse(n) Zugriff auf den lokalen E-Mail-Server haben und welche Art von Zugriff ihnen gestattet wird. Rechner können als `OK`, `REJECT` oder `RELAY` eingetragen oder einfach an **sendmail**s Fehlerbehandlungsroutine mit einem angegebenen Mailer-Fehler übergeben werden. Rechner, die als `OK` eingetragen sind, was die Grundeinstellung ist, sind berechtigt E-Mails zu diesem Rechner zu schicken, solange die endgültige Zieladresse der lokale Rechner ist. Verbindungen von Rechnern, die als `REJECT` aufgelistet sind, werden abgelehnt. Rechnern mit gesetzter `RELAY`-Option für ihren Rechnernamen wird erlaubt Post für jede Zieladresse durch diesen Mail-Server zu senden.

Beispiel 29-1. Konfigurieren der sendmail Zugriffsdatenbank

```
cyberspammer.com              550 We do not accept mail from spammers
FREE.STEALTH.MAILER@          550 We do not accept mail from spammers
another.source.of.spam        REJECT
okay.cyberspammer.com         OK
128.32                        RELAY
```

In diesem Beispiel haben wir fünf Einträge. E-Mail-Versender, die mit der linken Spalte der Tabelle übereinstimmen, sind betroffen von der Aktion in der rechten Spalte. Die ersten beiden Beispiele übergeben einen Fehlercode an **sendmail**s Fehlerbehandlungsroutine. Die Nachricht wird an den entfernten Rechner gesendet, wenn eine Nachricht

mit der linken Spalte der Tabelle übereinstimmt. Der nächste Eintrag lehnt Post von einem bestimmten Rechner des Internets ab (`another.source.of.spam`). Der nächste Eintrag akzeptiert E-Mail-Verbindungen des Rechners `okay.cyberspammer.com`, der exakter angegeben wurde als `cyberspammer.com` in der Zeile darüber. Genauere Übereinstimmungen haben den Vorrang vor weniger genauen. Der letzte Eintrag erlaubt die Weiterleitung von elektronischer Post von Rechnern mit einer IP-Adresse die mit `128.32` beginnt. Diese Rechner würden E-Mails durch diesen E-Mail-Server senden können, die für andere E-Mail-Server bestimmt sind.

Wenn diese Datei geändert wird, müssen Sie `make` in `/etc/mail` ausführen um die Datenbank zu aktualisieren.

29.4.2. `/etc/mail/aliases`

Die Alias-Datenbank enthält eine Liste der virtuellen Mailboxen, die in andere Benutzer, Dateien, Programme oder andere Aliase expandiert werden. Hier sind ein paar Beispiele, die in `/etc/mail/aliases` benutzt werden können:

Beispiel 29-2. E-Mail Aliases

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

Das Dateiformat ist simpel; Der Name der Mailbox auf der linken Seite des Doppelpunkts wird mit den Zielen auf der rechten Seite ersetzt. Das erste Beispiel ersetzt die Mailbox `root` mit der Mailbox `localuser`, die dann wieder in der Alias-Datenbank gesucht wird. Wird kein passender Eintrag gefunden, wird die Nachricht zum lokalen Benutzer `localuser` geliefert. Das nächste Beispiel zeigt eine E-Mail-Verteilerliste. E-Mails an die Mailbox `ftp-bugs` werden zu den drei lokalen Mailboxen `joe`, `eric` und `paul` gesendet. Eine lokale Mailbox kann auch als <`user@example.com`> angegeben werden. Das nächste Beispiel zeigt das Schreiben von E-Mails in eine Datei, in diesem Fall `/dev/null`. Das letzte Beispiel verdeutlicht das Senden von E-Mails an ein Programm, in diesem Fall wird die Nachricht in die Standardeingabe von `/usr/local/bin/procmail` mittels einer UNIX Pipe geschrieben.

Wenn diese Datei geändert wird, müssen Sie `make` in `/etc/mail` ausführen um die Änderungen in die Datenbank zu übernehmen.

29.4.3. `/etc/mail/local-host-names`

Das ist die Liste der Rechnernamen, die sendmail(8) als lokalen Rechnernamen akzeptiert. Setzen Sie alle Domänen oder Rechner, für die **sendmail** Mail empfangen soll, in diese Datei. Wenn dieser Mail-Server zum Beispiel E-Mails für die Domäne example.com und den Rechner `mail.example.com` annehmen soll, könnte seine `local-host-names` Datei so aussehen:

```
example.com
mail.example.com
```

Wird diese Datei geändert, muss sendmail(8) neu gestartet werden, damit es die Neuerungen einliest.

29.4.4. `/etc/mail/sendmail.cf`

Die Hauptkonfigurations-Datei von **sendmail** (`sendmail.cf`) kontrolliert das allgemeine Verhalten von **sendmail**, einschließlich allem vom Umschreiben von E-Mail Adressen bis hin zum Übertragen von Ablehnungsnachrichten an

entfernte E-Mail-Server. Mit solch einer mannigfaltigen Rolle ist die Konfigurationsdatei natürlich ziemlich komplex und ihre Einzelheiten können in diesem Kapitel nicht besprochen werden. Glücklicherweise muss diese Datei selten für Standard E-Mail-Server geändert werden.

Die **sendmail** Hauptkonfigurationsdatei kann mit m4(1) Makros erstellt werden, die Eigenschaften und Verhalten von **sendmail** definieren. Einige der Details finden Sie in `/usr/src/contrib/sendmail/cf/README`.

Wenn Änderungen an dieser Datei vorgenommen werden, muss **sendmail** neu gestartet werden, damit die Änderungen Wirkung zeigen.

29.4.5. `/etc/mail/virtusertable`

Die Datei `virtusertable` ordnet Adressen für virtuelle Domänen und Mailboxen reellen Mailboxen zu. Diese Mailboxen können lokal, auf entfernten Systemen, Aliase in `/etc/mail/aliases` oder eine Datei sein.

Beispiel 29-3. Beispiel einer virtuellen Domänen Zuordnung

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

In dem obigen Beispiel haben wir einen Eintrag für die Domäne `example.com`. Diese Datei wird nach dem ersten übereinstimmenden Eintrag durchsucht. Die erste Zeile ordnet <`root@example.com`> der lokalen Mailbox `root` zu. Der nächste Eintrag ordnet <`postmaster@example.com`> der Mailbox `postmaster` auf dem Rechner `noc.example.net` zu. Zuletzt, wenn keine Übereinstimmung von `example.com` gefunden wurde, wird der letzte Eintrag verglichen, der mit jeder Mail-Nachricht übereinstimmt, die an jemanden bei `example.com` adressiert wurde. Diese werden der lokalen Mailbox joe zugeordnet.

29.5. Wechseln des Mailübertragungs-Agenten

Geschrieben von Andrew Boothman. Informationen entnommen aus E-Mails geschrieben von Gregory Neil Shapiro.

Wie bereits erwähnt, ist bei FreeBSD **sendmail** schon als Ihr Mailübertragungs-Agent installiert. Deswegen ist es standardmäßig für Ihre aus- und eingehenden E-Mails verantwortlich.

Jedoch wollen einige Systemadministratoren den MTA ihres Systems wechseln, was eine Reihe von Gründen haben kann. Diese Gründe reichen von einfach einen anderen MTA ausprobieren wollen bis hin dazu eine bestimmte Besonderheit zu benötigen oder ein Paket, welches auf einen anderen Mailer angewiesen ist. Glücklicherweise macht FreeBSD das Wechseln einfach, egal aus welchem Grund.

29.5.1. Installieren eines neuen MTA

Sie haben eine große Auswahl an verfügbaren MTA-Programmen. Ein guter Startpunkt ist die FreeBSD-Ports-Sammlung, wo Sie viele finden werden. Selbstverständlich steht es Ihnen frei, jeden MTA von überall her zu verwenden, solange Sie ihn unter FreeBSD zum Laufen bekommen.

Fangen Sie an, indem Sie Ihren neuen MTA installieren. Sobald er installiert ist, gibt er Ihnen die Chance zu entscheiden ob er wirklich Ihren Bedürfnissen genügt. Zusätzlich gibt er Ihnen die Möglichkeit die neue Software zu

konfigurieren, bevor sie den Job von **sendmail** übernimmt. Dabei sollten Sie sicherstellen, dass beim Installieren der neuen Software keine Versuche unternommen werden, System-Programme wie `/usr/bin/sendmail` zu überschreiben. Ansonsten wurde Ihre neue E-Mail-Software in den Dienst gestellt, bevor Sie sie konfiguriert haben.

Für Informationen über die Konfiguration des von Ihnen gewählten MTAs sehen Sie bitte in der dazugehörigen Dokumentation nach.

29.5.2. Ausschalten von sendmail

> **Warnung:** Wenn Sie die Sendefunktion von **sendmail** deaktivieren, müssen Sie für den E-Mail-Versand ein alternatives System installieren. Tun Sie dies nicht, sind Systemfunktionen wie periodic(8) nicht mehr in der Lage, ihre Resulate und Meldungen als E-Mail zu versenden. Aber auch viele andere Teile Ihres Systems erwarten, dass Sie über ein **sendmail**-kompatibles System verfügen. Sind Programme auf (die von Ihnen deaktivierten) **sendmail**-Binärdateien angewiesen, landen deren E-Mails ansonsten in einer inaktiven **sendmail**-Warteschlange und können nicht ausgeliefert werden.

Um **sendmail** komplett zu deaktivieren (also inklusive der Funktion zum Versand von E-Mails), fügen Sie die Zeile

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

in `/etc/rc.conf` ein.

Wenn Sie lediglich die Funktion zum Empfang von E-Mails durch **sendmail** deaktivieren wollen, sollten Sie folgenden Eintrag in `/etc/rc.conf` einfügen:

```
sendmail_enable="NO"
```

Weitere Informationen zu den Startoptionen von **sendmail** finden Sie in der Manualpage rc.sendmail(8).

29.5.3. Starten Ihres neuen MTA beim Hochfahren des Systems

Der neue MTA kann beim Hochfahren durch das Hinzufügen einer Konfigurationszeile in der `/etc/rc.conf` gestartet werden, wie das folgende Beispiel für Postfix zeigt:

```
# echo 'postfix_enable="YES"' >> /etc/rc.conf
```

Der MTA wird jetzt automatisch beim Hochfahren des Systems gestartet.

29.5.4. Ersetzen von sendmail als Standard-Mailer des Systems

Das Programm **sendmail** ist so allgegenwärtig als Standard-Software auf UNIX Systemen, dass einige Programme einfach annehmen es sei bereits installiert und konfiguriert. Aus diesem Grund stellen viele alternative MTAs ihre eigenen kompatiblen Implementierung der **sendmail** Kommandozeilen-Schnittstelle zur Verfügung. Das vereinfacht ihre Verwendung als “drop-in” Ersatz für **sendmail**.

Folglich werden Sie, wenn Sie einen alternativen Mailer benutzen, sicherstellen müssen, dass ein Programm, das versucht **sendmail**s Standard-Dateien wie `/usr/bin/sendmail` auszuführen, stattdessen Ihr gewähltes Mailübertragungsprogramm ausführt. Zum Glück stellt FreeBSD das mailwrapper(8)-System zur Verfügung, das diese Arbeit für Sie erledigt.

Wenn **sendmail** arbeitet wie es installiert wurde, werden Sie in `/etc/mail/mailer.conf` etwas wie das Folgende vorfinden:

```
sendmail        /usr/libexec/sendmail/sendmail
send-mail      /usr/libexec/sendmail/sendmail
mailq          /usr/libexec/sendmail/sendmail
newaliases     /usr/libexec/sendmail/sendmail
hoststat       /usr/libexec/sendmail/sendmail
purgestat      /usr/libexec/sendmail/sendmail
```

Das bedeutet, dass wenn eines der gewöhnlichen Kommandos (wie zum Beispiel `/usr/bin/sendmail` selbst) ausgeführt wird, das System tatsächlich eine Kopie des mailwrapper mit dem Namen `sendmail` startet, die `mailer.conf` überprüft und `/usr/libexec/sendmail/sendmail` ausführt. Mit diesem System lassen sich die Programme, die für die **sendmail**-Funktionen gestartet werden, leicht ändern.

Daher könnten Sie, wenn Sie wollten, dass `/usr/local/supermailer/bin/sendmail-compat` anstelle von **sendmail** ausgeführt wird, `/etc/mailer.conf` wie folgt abändern:

```
sendmail        /usr/local/supermailer/bin/sendmail-compat
send-mail      /usr/local/supermailer/bin/sendmail-compat
mailq          /usr/local/supermailer/bin/mailq-compat
newaliases     /usr/local/supermailer/bin/newaliases-compat
hoststat       /usr/local/supermailer/bin/hoststat-compat
purgestat      /usr/local/supermailer/bin/purgestat-compat
```

29.5.5. Fertigstellen

Sobald Sie alles Ihren Wünschen entsprechend konfiguriert haben, sollten Sie entweder die **sendmail** Prozesse beenden, die Sie nicht mehr benötigen, und die zu Ihrer neuen Software zugehörigen Prozesse starten, oder einfach das System neustarten. Das Neustarten des Systems gibt Ihnen auch die Gelegenheit sicherzustellen, dass Sie Ihr System korrekt konfiguriert haben, um Ihren neuen MTA automatisch beim Hochfahren zu starten.

29.6. Fehlerbehebung

Hier finden sich ein paar häufig gestellte Fragen und ihre Antworten, die von der FAQ (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/faq/) übernommen wurden.

1. Warum muss ich einen FQDN (fully-qualified domain name/ voll ausgeschriebenen Domänennamen) für meine Rechner verwenden?

Vielleicht liegen die Rechner in einer unterschiedlichen Domäne; zum Beispiel, wenn Sie sich in `foo.bar.edu` befinden, und einen Rechner namens `mumble` in der `bar.edu` Domäne erreichen wollen, müssen Sie ihn mit dem voll ausgeschriebenen Domänennamen `mumble.bar.edu` kontaktieren, anstatt bloß mit `mumble`.

Traditionell wurde das von dem BSD BIND *Resolver* erlaubt. Wie auch immer, die aktuelle Version von **BIND**, die mit FreeBSD ausgeliefert wird, bietet keine Standardabkürzungen für nicht komplett angegebene Domänennamen außerhalb der Domäne, in der Sie sich befinden. Daher muss ein nicht-qualifizierter Rechner `mumble` entweder als `mumble.foo.bar.edu` gefunden werden, oder er wird in der root Domäne gesucht.

Damit unterscheidet es sich von vorherigem Verhalten, bei dem die Suche über `mumble.bar.edu` und `mumble.edu` lief. Schauen Sie sich RFC 1535 an, wenn Sie wissen möchten, warum das als schlecht und sogar als Sicherheitsloch angesehen wurde.

Um das zu umgehen, können Sie die Linie

```
search foo.bar.edu bar.edu
```

anstatt der vorherigen

```
domain foo.bar.edu
```

in Ihre `/etc/resolv.conf` einsetzen. Aber stellen Sie sicher, dass die Suchordnung nicht die Begrenzung von “lokaler und öffentlicher Administration”, wie RFC 1535 sie nennt, überschreitet.

2. Warum meldet Sendmail `mail loops back to myself`?

Dies wird in der Sendmail-FAQ wie folgt beantwortet:

Ich erhalte folgende Fehlermeldungen:

553 MX list for domain.net points back to relay.domain.net
554 <user@domain.net>... Local configuration error

Wie kann ich dieses Problem lösen?

> Sie haben durch die Benutzung eines MX-Eintrags eingestellt, dass
> Mail für die Domäne (z.B. domain.net) an einen speziellen
> Host (in diesem Fall relay.domain.net) weitergeleitet wird,
> aber der Relay-Host erkennt sich selbst nicht als
> domain.net. Fügen Sie domain.net in /etc/mail/local-host-names
> [die Datei hieß vor der Version 8.10 /etc/sendmail.cw]
> (falls Sie FEATURE(use_cw_file) benutzen) oder "Cw domain.net"
> in /etc/mail/sendmail.cf ein.

Die aktuelle Version der Sendmail-FAQ (ftp://rtfm.mit.edu/pub/usenet/news.answers/mail/sendmail-faq) wird nicht mehr mit dem Sendmail-Release verwaltet. Sie wird jedoch regelmäßig nach comp.mail.sendmail (news:comp.mail.sendmail), comp.mail.misc (news:comp.mail.misc), comp.mail.smail (news:comp.mail.smail), comp.answers (news:comp.answers) und news.answers (news:news.answers) gepostet. Sie können auch eine Kopie per E-Mail bekommen, indem Sie eine Mail mit dem Inhalt `send usenet/news.answers/mail/sendmail-faq` an `mail-server@rtfm.mit.edu` schicken.

3. Wie kann ich einen E-Mail-Server auf einem Anwahl-PPP Rechner betreiben?

Sie wollen einen FreeBSD-Rechner in einem LAN an das Internet anbinden. Der FreeBSD-Rechner wird ein E-Mail *Gateway* für das LAN. Die PPP-Verbindung ist keine Standleitung.

Es gibt mindestens zwei Wege um dies zu tun. Einer davon ist UUCP zu verwenden.

Ein anderer Weg ist, von einem immer mit dem Internet verbundenen Server einen sekundären MX-Dienst für Ihre Domäne zur Verfügung gestellt zu bekommen. Wenn die Domäne Ihrer Firma `example.com` ist, und Ihr Internet-Dienstanbieter `example.net` so eingestellt hat, dass er Ihrer Domäne einen sekundären MX-Dienst zur Verfügung stellt:

```
example.com.            MX        10      bigco.com.
                      MX        20      example.net.
```

Nur ein Rechner sollte als Endempfänger angegeben sein (fügen Sie `Cw example.com` zu `/etc/sendmail.cf` auf example.com).

Wenn das `sendmail` des Versenders versucht, die E-Mail zuzustellen, wird es versuchen, Sie über die Modem-Verbindung (`example.com`) zu erreichen. Wahrscheinlich wird es keine Verbindung zustande bringen können, da Sie nicht eingewählt sind. `sendmail` wird die E-Mail automatisch zu der sekundären MX-Stelle geliefert, zu Ihrem Internet-Provider (`example.net`). Die sekundäre MX-Stelle wird periodisch versuchen versuchen eine Verbindung zu Ihnen aufzubauen, um die E-Mail zu der primären MX-Stelle (`example.com` zu liefern.

Eventuell wollen Sie etwas wie dies als Login-Skript:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Wenn Sie ein separates Login-Skript für einen Benutzer erstellen wollen, könnten Sie stattdessen `sendmail -qRexample.com` in dem oben gezeigten Skript verwenden. Das erzwingt die sofortige Verarbeitung der E-Mails in Ihrer Warteschlange für `example.com`

Eine weitere Verfeinerung der Situation ist wie folgt:

Die Nachricht wurde der FreeBSD Internet service providers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp) entnommen.

```
> wir stellen einem Kunden den sekundären MX zur Verfügung.
> Der Kunde verbindet sich mit unseren Diensten mehrmals am Tag
> automatisch um die E-Mails zu seinem primären MX zu holen
> (wir wählen uns nicht bei ihm ein, wenn E-Mails für seine
> Domäne eintreffen).  Unser sendmail sendet den Inhalt der
> E-Mail-Warteschlange alle 30 Minuten.  Momentan muss er 30 Minuten
> eingewählt bleiben um sicher zu sein, dass alle seine E-Mails
> beim primären MX eingetroffen sind.
>
> Gibt es einen Befehl, der sendmail dazu bringt, alle E-Mails sofort
> zu senden? Der Benutzer hat natürlich keine root-Rechte auf
> unserer Maschine.

In der “privacy flags” Sektion von sendmail.cf befindet sich die
Definition Opgoaway,restrictqrun
```

```
Entferne restrictqrun um nicht-root Benutzern zu erlauben, die Verarbeitung
der Nachrichten-Warteschlangen zu starten.  Möglicherweise willst du
auch die MX neu sortieren.  Wir sind der primäre MX für unsere
Kunden mit diesen Wünschen und haben definiert:

# Wenn wir der beste MX für einen Rechner sind, versuche es direkt
# anstatt einen lokalen Konfigurationsfehler zu generieren.
OwTrue

Auf diesem Weg liefern Gegenstellen direkt zu dir, ohne die Kundenverbindung
zu versuchen.  Dann sendest du zu deinem Kunden.  Das funktioniert nur
für “Rechner”, du musst also deinen Kunden dazu bringen,
ihre E-Mail Maschine “customer.com” zu nennen, sowie
“hostname.customer.com” im DNS.  Setze einfach einen A-Eintrag
in den DNS für “customer.com”.
```

4. Warum bekomme ich die Fehlermeldung `Relaying Denied`, wenn ich E-Mails von anderen Rechnern verschicke?

In der standardmäßigen FreeBSD-Installation wird **sendmail** nur dazu konfiguriert, E-Mails von dem Rechner, auf dem es läuft, zu senden. Wenn zum Beispiel ein POP-Server installiert ist, können Benutzer ihre E-Mails von der Schule, von der Arbeit oder von anderen Orten überprüfen. Sie werden jedoch keine E-Mails von außen verschicken können. Typischerweise wird ein paar Sekunden nach dem Versuch eine E-Mail von **MAILER-DAEMON** mit einer `5.7 Relaying Denied` Fehlermeldung versendet werden.

Es sind mehrere Wege möglich, dies zu umgehen. Die geradlinigste Lösung ist die Adresse Ihres Internet-Dienstanbieters in die Datei für die Weiterleitungs-Domänen zu platzieren. Das lässt sich schnell erreichen mit:

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

Nach Erstellen oder Editieren dieser Datei müssen Sie **sendmail** neu starten. Das funktioniert großartig wenn Sie ein Server-Administrator sind und E-Mails nicht lokal versenden, oder gerne ein Client/System mit grafischer Oberfläche auf einer anderen Maschine oder sogar über einen anderen Internet-Dienstanbieter verwenden wollen. Es ist auch sehr nützlich, wenn Sie nur ein oder zwei E-Mail Accounts eingerichtet haben. Soll eine größere Anzahl Adressen hinzugefügt werden, können Sie die Datei einfach in Ihrem favorisierten Editor öffnen und die Domänen anfügen, je eine pro Zeile:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Jetzt wird jede E-Mail, die durch Ihr System von einem der in diese Liste eingetragenen Rechner geschickt wurde, ihr Ziel erreichen (vorausgesetzt, der Benutzer hat einen Account auf Ihrem System). Dies ist ein sehr schöner Weg, um Benutzern das entfernte E-Mail Versenden von Ihrem System zu erlauben, ohne dem Massenversand (SPAM) die Tür zu öffnen.

29.7. Weiterführende Themen

Die folgenden Abschnitte behandeln kompliziertere Themen wie E-Mail-Konfiguration und das Einrichten von E-Mail für Ihre ganze Domäne.

29.7.1. Grundlegende Konfiguration

Mit der Software im Auslieferungszustand sollten Sie fähig sein, E-Mails an externe Rechner zu senden, solange Sie `/etc/resolv.conf` eingerichtet haben oder Ihren eigenen Name Server laufen lassen. Wenn Sie die E-Mails für Ihren Rechner zu einem anderen Rechner geliefert haben wollen, gibt es zwei Methoden:

- Betreiben Sie Ihren eigenen Name Server und haben Sie Ihre eigene Domäne, zum Beispiel `FreeBSD.org`.
- Lassen Sie sich E-Mails direkt zu Ihrem Rechner liefern. Dies geschieht indem E-Mails direkt zu dem aktuellen DNS Namen Ihrer Maschine geliefert werden. Zum Beispiel `example.FreeBSD.org`.

Ungeachtet welche Methode Sie auswählen, um E-Mails direkt zu Ihrem Rechner geliefert zu bekommen, benötigen Sie eine permanente (statische) IP-Adresse (keine dynamische PPP-Anwahl). Wenn Sie sich hinter einer Firewall befinden, muss diese den SMTP-Verkehr an Sie weiterleiten. Wollen Sie E-Mails an Ihrem Rechner selbst empfangen, müssen Sie eines der folgenden Dinge sicherstellen:

- Vergewissern Sie sich, dass der MX-Eintrag in Ihrem DNS zu der IP-Adresse Ihres Rechners zeigt.
- Stellen Sie sicher, dass sich für Ihren Rechner kein MX-Eintrag im DNS befindet.

Jede der erwähnten Konfigurationsmöglichkeiten erlaubt Ihnen, E-Mails direkt auf Ihrem Rechner zu empfangen.

Versuchen Sie das:

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

Wenn Sie diese Ausgabe erhalten, sollten direkt an <`yourlogin@example.FreeBSD.org`> geschickte E-Mails ohne Probleme funktionieren.

Sehen Sie stattdessen etwas wie dies:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

So wird jede an Ihren Rechner (`example.FreeBSD.org`) gesandte E-Mail auf `hub` unter dem gleichen Benutzernamen gesammelt anstatt direkt zu Ihrem Rechner geschickt zu werden.

Die obige Information wird von Ihrem DNS-Server verwaltet. Der DNS-Eintrag, der die E-Mail Routen-Information enthält, ist der *Mail eXchange* Eintrag. Existiert kein MX-Eintrag, werden E-Mails direkt anhand der IP-Adresse geliefert.

Der MX-Eintrag für `freefall.FreeBSD.org` sah einmal so aus:

```
freefall                 MX      30      mail.crl.net
freefall                MX      40      agora.rdrop.com
freefall                MX      10      freefall.FreeBSD.org
```

```
freefall                MX      20      who.cdrom.com
```

Wie Sie sehen können, hatte `freefall` viele MX-Einträge. Die kleinste MX-Nummer ist der Rechner, der die E-Mails letztendlich bekommt, wobei die anderen temporär E-Mails in Warteschlangen einreihen während `freefall` beschäftigt oder unerreichbar ist.

Um besonders nützlich zu sein, sollten stellvertretende MX-Seiten nicht dieselben Internet-Verbindungen wie Ihre eigene verwenden. Für Ihren Internet-Dienstleister oder andere sollte es kein Problem darstellen, Ihnen diesen Dienst zur Verfügung zu stellen.

29.7.2. E-Mails für Ihre Domäne

Um einen “E-Mail-Server” (auch bekannt als Mail-Server) einzurichten, benötigen Sie eine Umlenkung jeglicher E-Mails zu Ihm, die an die verschiedenen Workstations gesendet werden. Im Grunde wollen Sie jede an Ihre Domäne gesendete E-Mail abfangen (in diesem Fall `*.FreeBSD.org`), damit Ihre Benutzer E-Mails mittels POP oder direkt auf dem Server überprüfen können.

Am einfachsten ist es, wenn Accounts mit gleichen *Benutzernamen* auf beiden Maschinen existieren. Verwenden Sie adduser(8), um dies zu erreichen.

Der E-Mail-Server, den Sie verwenden wollen, muss als für den E-Mail-Austausch zuständiger Rechner auf jeder Workstation im Netzwerk gekennzeichnet werden. Dies wird in der DNS-Konfiguration so ausgeführt:

```
example.FreeBSD.org     A       204.216.27.XX           ; Workstation
                        MX      10 hub.FreeBSD.org      ; Mailhost
```

Diese Einstellung wird E-Mail für die Workstations zu dem E-Mail-Server leiten, wo auch immer der A-Eintrag hinzeigt. Die E-Mails werden zum MX-Rechner gesandt.

Sofern Sie nicht einen DNS-Server laufen haben, können Sie diese Einstellung nicht selbst vornehmen. Ist es Ihnen nicht möglich einen eigenen DNS-Server laufen zu lassen, reden Sie mit Ihren Internet-Dienstleister oder wer auch immer Ihre DNS-Verwaltung übernimmt.

Wenn Sie ein virtuelles E-Mail System anbieten, werden die folgenden Informationen nützlich sein. Für ein Beispiel nehmen wir an, Sie haben einen Kunden mit einer eigenen Domäne, in diesem Fall `customer1.org` und Sie wollen jegliche E-Mails für `customer1.org` zu Ihrem E-Mail-Server gesendet haben, der `mail.myhost.com` heißt. Der Eintrag in Ihrem DNS sollte wie folgender aussehen:

```
customer1.org           MX      10      mail.myhost.com
```

Sie benötigen *keinen* A-Eintrag, wenn Sie für die Domain nur E-Mails verwalten wollen.

> **Anmerkung:** Bedenken Sie, dass das Pingen von `customer1.org` nicht möglich ist, solange kein A-Eintrag für diese Domäne existiert.

Jetzt müssen Sie nur noch **sendmail** auf Ihrem Mailrechner mitteilen, für welche Domänen und/oder Rechnernamen es Mails akzeptieren soll. Es gibt einige Wege wie dies geschehen kann. Die Folgenden funktionieren alle gleichermaßen:

- Fügen Sie die Rechnernamen zu Ihrer `/etc/sendmail.cw` Datei hinzu, wenn Sie `FEATURE(use_cw_file)` verwenden. Ab **sendmail** 8.10 heißt diese Datei `/etc/mail/local-host-names`.
- Tragen Sie eine Zeile mit dem Inhalt `Cwyour.host.com` in Ihre `/etc/sendmail.cf` Datei (beziehungsweise `/etc/mail/sendmail.cf` ab **sendmail** 8.10) ein.

29.8. SMTP über UUCP

Die **sendmail**-Konfigurationsdatei, die mit FreeBSD ausgeliefert wird, ist für Systeme geeignet, die direkt ans Internet angeschlossen sind. Systeme, die ihre E-Mails per UUCP austauschen wollen, müssen eine andere Konfigurationsdatei installieren.

Die manuelle Bearbeitung von `/etc/mail/sendmail.cf` ist nur etwas für Puristen. Sendmail Version 8 bietet die neue Möglichkeit der Generierung von Konfigurationsdateien über eine Vorverarbeitung mit m4(1), wobei die tatsächliche, händische Konfiguration auf einer höheren Abstraktionsstufe stattfindet. Sie sollten die Konfigurationsdateien unter `/usr/src/usr.sbin/sendmail/cf` benutzen. Die Datei `README` im Verzeichnis `cf` kann zur grundlegenden Einführung in die m4(1)-Konfiguration dienen.

Zur Zustellung über UUCP sind Sie am besten damit beraten, die `mailertable`-Datenbank zu benutzen. Mit dieser Datenbank ermittelt **sendmail** mit welchem Protokoll und wohin eine E-Mail zugestellt werden soll.

Zunächst müssen Sie Ihre `.mc`-Datei erstellen. Das Verzeichnis `/usr/share/sendmail/cf/cf` ist die Basis für diese Dateien. Sehen Sie sich um, es gibt bereits einige Beispiele. Wenn Sie Ihre Datei `foo.mc` genannt haben, müssen Sie die folgenden Befehle ausführen, um sie in eine gültige `sendmail.cf` umzuwandeln:

```
# cd /etc/mail
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

Eine typische `.mc`-Datei könnte so aussehen:

```
VERSIONID('Your version number')
OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, 'hash -o /etc/mail/mailertable')

define('UUCP_RELAY', your.uucp.relay)
define('UUCP_MAX_SIZE', 200000)
define('confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

Die Einstellungen `accept_unresolvable_domains`, `nocanonify` und `confDONT_PROBE_INTERFACES` werden die Benutzung von DNS bei der Zustellung von Mails verhindern. Die Klausel `UUCP_RELAY` wird aus seltsamen

Gründen benötigt – fragen Sie nicht, warum. Setzen Sie dort einfach den Namen eines Hosts ein, der in der Lage ist, Adressen mit der Pseudodomäne .UUCP zu behandeln; wahrscheinlich werden Sie dort den Relayhost Ihres ISP eintragen.

Wenn Sie soweit sind, müssen Sie die Datei `/etc/mail/mailertable` erzeugen. Hierzu wieder ein typisches Beispiel:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
.  uucp-dom:your.uucp.relay
```

Ein komplexeres Beispiel könnte wie folgt aussehen:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de   uucp-dom:horus
.interface-business.de        uucp-dom:if-bus
interface-business.de         uucp-dom:if-bus
.heep.sax.de                  smtp8:%1
horus.UUCP                    uucp-dom:horus
if-bus.UUCP                   uucp-dom:if-bus
.                             uucp-dom:
```

Die ersten drei Zeilen behandeln spezielle Fälle, in denen an Domänen adressierte E-Mails nicht über die Standard-Route versendet werden sollen, sondern zu einem UUCP-Nachbarn, um den Zustellweg "abzukürzen". Die nächsten Zeilen behandeln E-Mails an Rechner in der lokalen Domain. Diese Mails können direkt per SMTP zugestellt werden. Schließlich werden die UUCP-Nachbarn in der Notation mit der Pseudodomäne .UUCP aufgeführt, um die Standardregeln mit *`uucp-neighbour`*`!`*`recipient`* zu überschreiben. Die letzte Zeile besteht stets aus einem einzelnen Punkt, der als Ihr Universalgateway in die Welt dient. Alle Knoten hinter dem Schlüsselwort `uucp-dom:` müssen gültige UUCP-Nachbarn sein, was Sie mit dem Befehl `uuname` überprüfen können.

Als Erinnerung daran, dass diese Datei in eine DBM-Datenbankdatei konvertiert werden muss, bevor sie benutzt werden kann, sollte der Befehl hierzu als Kommentar am Anfang der `mailertable` plaziert werden. Sie müssen den Befehl jedes Mal ausführen, wenn Sie die `mailertable` geändert haben.

Abschließender Hinweis: Wenn Sie unsicher sind, ob bestimmte Zustellwege funktionieren, erinnern Sie sich an die Option `-bt` von **sendmail**. Sie startet **sendmail** im *Adress-Testmodus*; geben Sie einfach `3,0`, gefolgt von der Adresse, für die Sie den Zustellweg testen möchten, ein. Die letzte Zeile nennt Ihnen den benutzten Mailagenten, den Zielhost, mit dem dieser Agent aufgerufen wird und die (möglicherweise übersetzte) Adresse. Verlassen Sie diesen Modus, indem Sie **Ctrl**+**D** eingeben.

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify           input: foo @ example . com
...
parse            returns: $# uucp-dom $@ your.uucp.relay $: foo < @ example . com . >
> ^D
```

29.9. Ausgehende E-Mail über einen Relay versenden

Beigetragen von Bill Moran.

In vielen Fällen wollen Sie E-Mail nur über einen Relay verschicken. Zum Beispiel:

- Sie wollen von Ihrem Arbeitsplatz Programme wie send-pr(1) benutzen. Dazu soll der Relay Ihres ISPs verwendet werden.
- Ein Server, der E-Mails nicht selbst verarbeitet, soll alle E-Mails zu einem Relay schicken.

So ziemlich jeder MTA kann diese Aufgaben erfüllen. Leider ist es oft schwierig, einen vollwertigen MTA so zu konfigurieren, dass er lediglich ausgehende E-Mails weiterleitet. Es ist übertrieben, Programme wie **sendmail** und **postfix** nur für diesen Zweck einzusetzen.

Weiterhin kann es sein, dass die Bestimmungen Ihres Internetzugangs es verbieten, einen eigenen Mail-Server zu betreiben.

Um die hier beschriebenen Anforderungen zu erfüllen, installieren Sie einfach den Port `mail/ssmtp`. Führen Sie dazu als `root` die nachstehenden Befehle aus:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Nach der Installation konfigurieren Sie `mail/ssmtp` mit den folgenden vier Zeilen in `/usr/local/etc/ssmtp/ssmtp.conf`:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Stellen Sie sicher, dass Sie eine gültige E-Mail-Adresse für `root` verwenden. Geben Sie für `mail.example.com` den Mail-Relay Ihres ISPs an (einige ISPs nennen den Relay “Postausgangsserver” oder “SMTP-Server”).

Deaktivieren Sie **sendmail** indem Sie in `/etc/rc.conf` `sendmail_enable="NONE"` angeben.

`mail/ssmtp` verfügt über weitere Optionen. Die Musterkonfiguration in `/usr/local/etc/ssmtp` oder die Hilfeseite von **ssmtp** enthalten weitere Beispiele.

Wenn Sie **ssmtp** wie hier beschrieben eingerichtet haben, funktionieren Anwendungen, die E-Mails von Ihrem Rechner verschicken. Sie verstoßen damit auch nicht gegen Bestimmungen Ihres ISPs und laufen nicht in Gefahr, dass Ihr Rechner zum Versenden von Spams missbraucht wird.

29.10. E-Mail über Einwahl-Verbindungen

Wenn Sie eine feste IP-Adresse haben, müssen Sie die Standardeinstellungen wahrscheinlich gar nicht ändern. Stellen Sie Ihren Hostnamen entsprechend Ihrem zugeordneten Internetnamen ein und **sendmail** übernimmt das Übrige.

Wenn Sie eine dynamische IP-Adresse haben und eine **PPP**-Wählverbindung zum Internet benutzen, besitzen Sie wahrscheinlich eine Mailbox auf dem Mailserver Ihres ISPs. Lassen Sie uns annehmen, die Domäne ihres ISPs sei `example.net` und Ihr Benutzername `user`; außerdem nehmen wir an, dass Sie Ihre Maschine `bsd.home` genannt haben und, dass Ihr ISP ihnen gesagt hat, dass Sie `relay.example.net` als Mail-Relayhost benutzen können.

Um Mails aus Ihrer Mailbox abzuholen, müssen Sie ein gesondertes Programm installieren; **fetchmail** ist eine gute Wahl, weil es viele verschiedene Protokolle unterstützt. Das Programm können Sie als Paket oder von der Ports-Sammlung (`mail/fetchmail`) installieren. Für gewöhnlich wird von Ihrem ISP POP zur Verfügung gestellt. Falls Sie sich dafür entschieden haben, User-PPP zu benutzen, können Sie durch folgenden Eintrag in der Datei `/etc/ppp/ppp.linkup` Ihre Mails automatisch abholen lassen, wenn eine Verbindung zum Netz aufgebaut wird:

```
MYADDR:
!bg su user -c fetchmail
```

Falls Sie (wie unten gezeigt) **sendmail** benutzen, um Mails an nicht-lokale Benutzer zu versenden, fügen Sie den Befehl

```
!bg su user -c "sendmail -q"
```

nach dem oben gezeigten Eintrag ein. Das veranlasst **sendmail**, Ihre ausgehenden Mails zu verarbeiten, sobald eine Verbindung zum Internet aufgebaut wird.

Nehmen wir an, dass auf `bsd.home` ein Benutzer `user` existiert. Erstellen Sie auf `bsd.home` im Heimatverzeichnis von `user` die Datei `.fetchmailrc`:

```
poll example.net protocol pop3 fetchall pass MySecret;
```

Diese Datei sollte für niemandem außer `user` lesbar sein, weil sie das Passwort `MySecret` enthält.

Um Mails mit dem richtigen `from:`-Header zu versenden, müssen Sie **sendmail** mitteilen, dass es <`user@example.net`> und nicht <`user@bsd.home`> benutzen soll. Eventuell möchten Sie auch, dass **sendmail** alle Mails über `relay.example.net` versendet, um eine schnellere Übertragung von Mails zu gewährleisten.

Die folgende `.mc`-Datei sollte ausreichen:

```
VERSIONID(‘bsd.home.mc version 1.0’)
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(‘example.net’)dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(‘SMART_HOST’, ‘relay.example.net’)
Dmbsd.home
define(‘confDOMAIN_NAME’,‘bsd.home’)dnl
define(‘confDELIVERY_MODE’,‘deferred’)dnl
```

Im vorherigen Abschnitt finden Sie Details dazu, wie Sie aus dieser `.mc`-Datei eine Datei `sendmail.cf` erstellen können. Vergessen Sie auch nicht, **sendmail** neu zu starten, nachdem Sie `sendmail.cf` verändert haben.

29.11. SMTP-Authentifizierung

Geschrieben von James Gorham.

Ein Mail-Server, der SMTP-Authentifizierung verwendet, bietet einige Vorteile. Die erforderliche Authentifizierung erhöht die Sicherheit von **sendmail** und Benutzer, die auf wechselnden entfernten Rechnern arbeiten, können denselben Mail-Server verwenden ohne Ihr Benutzerprogramm jedes Mal neu zu konfigurieren.

1. Installieren Sie den Port `security/cyrus-sasl2`. Der Port verfügt über einige Optionen, die während der Übersetzung festgelegt werden. Für die in diesem Abschnitt beschriebene Methode zur SMTP-Authentifizierung muss die Option `LOGIN` aktiviert werden.
2. Editieren Sie nach der Installation von `security/cyrus-sasl2` die Datei `/usr/local/lib/sasl2/Sendmail.conf` (erstellen Sie die Datei, wenn sie nicht existiert) und fügen Sie die folgende Zeile hinzu:

   ```
   pwcheck_method: saslauthd
   ```
3. Danach installieren Sie den Port `security/cyrus-sasl2-saslauthd`, und fügen die folgende Zeile in `/etc/rc.conf` ein:

   ```
   saslauthd_enable="YES"
   ```

   Zuletzt müssen Sie noch den saslauthd-Daemon starten:

   ```
   # /usr/local/etc/rc.d/saslauthd start
   ```

   Dieser Daemon agiert als Broker zwischen **sendmail** und Ihrer FreeBSD-`passwd`-Datenbank. Dadurch müssen zum Versenden von E-Mails keine zusätzlichen Accounts und Passwörter angelegt werden. Die Benutzer verwenden dasselbe Passwort zum Anmelden wie zum Verschicken von E-Mails.
4. Fügen Sie jetzt in `/etc/make.conf` die nachstehenden Zeilen hinzu:

   ```
   SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
   SENDMAIL_LDFLAGS=-L/usr/local/lib
   SENDMAIL_LDADD=-lsasl2
   ```

   Beim Übersetzen von **sendmail** werden damit die `cyrus-sasl2`-Bibliotheken benutzt. Stellen Sie daher vor dem Übersetzen von **sendmail** sicher, dass der Port `cyrus-sasl2` installiert ist.
5. Übersetzen Sie **sendmail** mit den nachstehenden Kommandos:

   ```
   # cd /usr/src/lib/libsmutil
   # make cleandir && make obj && make
   # cd /usr/src/lib/libsm
   # make cleandir && make obj && make
   # cd /usr/src/usr.sbin/sendmail
   # make cleandir && make obj && make && make install
   ```

   sendmail sollte sich ohne Probleme übersetzen lassen, wenn die Dateien in `/usr/src` nicht verändert wurden und die benötigten Bibliotheken installiert sind.
6. Nachdem Sie **sendmail** installiert haben, editieren Sie `/etc/mail/freebsd.mc` beziehungsweise die verwendete `.mc`-Datei. Viele Administratoren verwenden die Ausgabe von hostname(1), um der `.mc`-Datei einen eindeutigen Namen zu geben. Fügen Sie die folgenden Zeilen in die `.mc`-Datei ein:

   ```
   dnl set SASL options
   TRUST_AUTH_MECH(‘GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN’)dnl
   define(‘confAUTH_MECHANISMS’, ‘GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN’)dnl
   ```

Diese Anweisungen konfigurieren die Methoden, die **sendmail** zur Authentifizierung verwendet. Lesen Sie die mitgelieferte Dokumentation, wenn Sie eine andere Methode als `pwcheck` verwenden wollen.

7. Abschließend rufen Sie make(1) im Verzeichnis `/etc/mail` auf. Damit wird aus der `.mc`-Datei eine neue `.cf`-Datei (zum Beispiel `freebsd.cf`) erzeugt. Das Kommando `make install restart` installiert die Datei nach `/etc/mail/sendmail.cf` und startet **sendmail** neu. Weitere Informationen entnehmen Sie bitte `/etc/mail/Makefile`.

Wenn alles funktioniert hat, tragen Sie in Ihrem Mail-Benutzerprogramm das Passwort für die Authentifizierung ein und versenden Sie zum Testen eine E-Mail. Wenn Sie Probleme haben, setzen Sie den `LogLevel` von **sendmail** auf `13` und untersuchen die Fehlermeldungen in `/var/log/maillog`.

Weitere Information erhalten Sie im WWW auf der Webseite von **sendmail** (http://www.sendmail.org/~ca/email/auth.html).

29.12. E-Mail-Programme

Beigetragen von Marc Silver.

Anwendungen, die E-Mails versenden und empfangen, werden als E-Mail-Programme oder Mail-User-Agents (MUA) bezeichnet. Mit der Entwicklung und Ausbreitung von E-Mail wachsen auch die E-Mail-Programme und bieten Benutzern mehr Funktionen und höhere Flexibilität. Unter FreeBSD laufen zahlreiche E-Mail-Programme, die Sie alle mit der FreeBSD Ports-Sammlung installieren können. Sie können wählen zwischen Programmen mit grafischer Benutzeroberfläche, wie **evolution** oder **balsa**, konsolenorientierten Programmen wie **mutt**, **alpine** oder `mail`, oder auch Programmen mit Web-Schnittstellen, die von einigen großen Institutionen benutzt werden.

29.12.1. mail

Das standardmäßig unter FreeBSD installierte E-Mail-Programm ist mail(1). Das Programm ist konsolenorientiert und enthält alle Funktionen, die zum Versand und Empfang textbasierter E-Mails erforderlich sind. Allerdings lassen sich Anhänge mit `mail` nur schwer bearbeiten und kann `mail` kann nur auf lokale Postfächer zugreifen.

`mail` kann nicht direkt auf POP- oder IMAP-Server zugreifen. Entfernte Postfächer können aber mit einer Anwendung wie **fetchmail** in die lokale Datei `mbox` geladen werden. **fetchmail** wird später in diesem Kapitel besprochen (Abschnitt 29.13).

Um E-Mails zu versenden oder zu empfangen, starten Sie einfach `mail` wie im nachstehenden Beispiel:

```
% mail
```

Das Werkzeug `mail` liest automatisch den Inhalt des Benutzer-Postfachs im Verzeichnis `/var/mail`. Sollte das Postfach leer sein, beendet sich `mail` mit der Nachricht, dass keine E-Mails vorhanden sind. Wenn das Postfach gelesen wurde, wird die Benutzeroberfläche gestartet und eine Liste der E-Mails angezeigt. Die E-Mails werden automatisch nummeriert wie im folgenden Beispiel gezeigt:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Einzelne Nachrichten können Sie jetzt mit dem `mail`-Kommando **t** gefolgt von der Nummer der Nachricht lesen. Im nachstehenden Beispiel lesen wir die erste E-Mail:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

Das ist eine Test-Nachricht.  Antworte bitte!
```

Die Taste **t** zeigt die Nachricht zusammen mit dem vollständigen Nachrichtenkopf an. Wenn Sie die Liste der E-Mails erneut sehen wollen, drücken Sie die Taste **h**.

Um auf eine E-Mail zu antworten, benutzen Sie im Programm `mail` entweder die Taste **R** oder die Taste **r**. Mit der Taste **R** weisen Sie `mail` an, dem Versender der Nachricht zu antworten. Mit der Taste **r** antworten Sie nicht nur dem Versender sondern auch allen Empfängern der Nachricht. Sie können zusammen mit diesen Kommandos eine Zahl angeben, um die E-Mail, auf die Sie antworten wollen, auszusuchen. Wenn Sie den Befehl abgesetzt haben, schreiben Sie Ihre Antwort und beenden die Eingabe mit einem einzelnen Punkt (`.`) auf einer neuen Zeile. Den Vorgang zeigt das nachstehende Beispiel:

```
& R 1
To: root@localhost
Subject: Re: test

Danke, ich habe deine E-Mail erhalten.
.
EOT
```

Neue E-Mails können Sie mit der Taste **m** verschicken. Geben Sie dabei die E-Mail-Adresse des Empfängers an. Sie können auch mehrere durch Kommata (`,`) getrennte Empfänger angeben. Geben Sie dann den Betreff (*subject*) der Nachricht gefolgt von der Nachricht selbst ein. Schließen Sie die Nachricht mit einem einzelnen Punkt (`.`) auf einer neuen Zeile ab.

```
& mail root@localhost
Subject: Ich habe die E-Mails im Griff!

Jetzt kann ich E-Mails versenden und empfangen ... :)
.
EOT
```

Die Taste **?** zeigt zu jeder Zeit einen Hilfetext an. Wenn Sie weitere Hilfe benötigen, lesen Sie bitte die Hilfeseite mail(1).

> **Anmerkung:** Wie vorhin gesagt, wurde das Programm mail(1) nicht für den Umgang mit Anhängen entworfen und kann daher sehr schlecht mit Anhängen umgehen. Neuere MUAs wie **mutt** gehen wesentlich besser mit Anhängen um. Sollten Sie dennoch das `mail`-Kommando benutzen wollen, werden Sie den Port `converters/mpack` sehr zu schätzen wissen.

29.12.2. mutt

mutt ist ein schlankes aber sehr leistungsfähiges E-Mail-Programm mit hervorrangenden Funktionen, unter anderem:

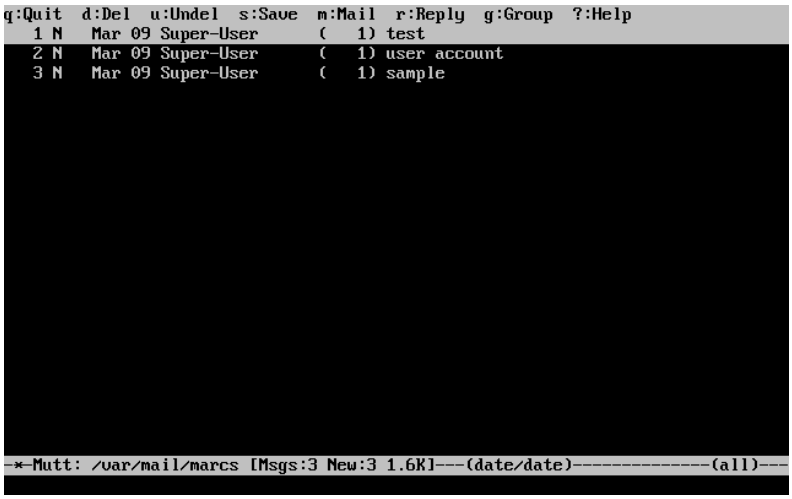
- **mutt** kann den Verlauf einer Diskussion (*threading*) darstellen.
- Durch die Integration von PGP können E-Mails signiert und verschlüsselt werden.
- MIME wird unterstützt.
- Postfächer können im Maildir-Format gespeichert werden.
- **mutt** lässt sich im höchsten Maße an lokale Bedürfnisse anpassen.

Wegen des Funktionsumfangs ist **mutt** eins der ausgefeiltesten E-Mail-Programme. Mehr über **mutt** erfahren Sie auf der Seite http://www.mutt.org.

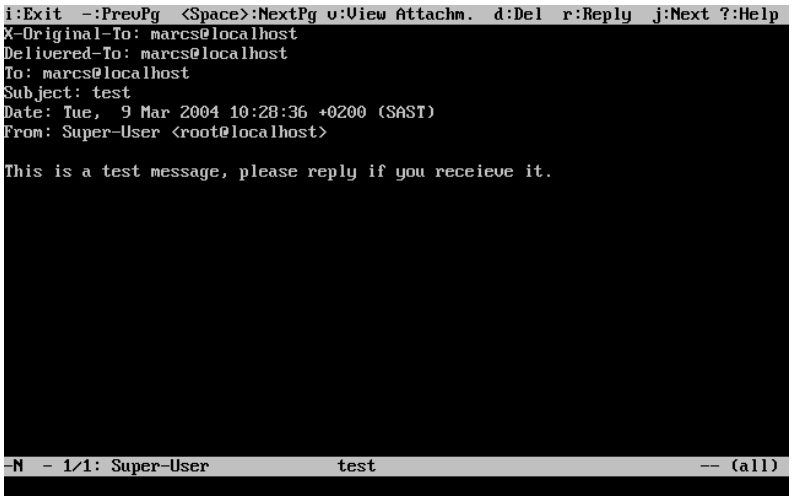
Der Port `mail/mutt` enthält die Produktionsversion von **mutt**, die aktuelle Entwicklerversion befindet sich im Port `mail/mutt-devel`. Wenn **mutt** installiert ist, wird das Programm mit dem nachstehenden Kommando gestartet:

```
% mutt
```

mutt liest automatisch den Inhalt des Benutzer-Postfachs im Verzeichnis `/var/mail`. Wenn E-Mails vorhanden sind, werden diese dargestellt. Sind keine E-Mails vorhanden, wartet **mutt** auf Benutzereingaben. Das folgende Beispiel zeigt, wie **mutt** eine Nachrichten-Liste darstellt:



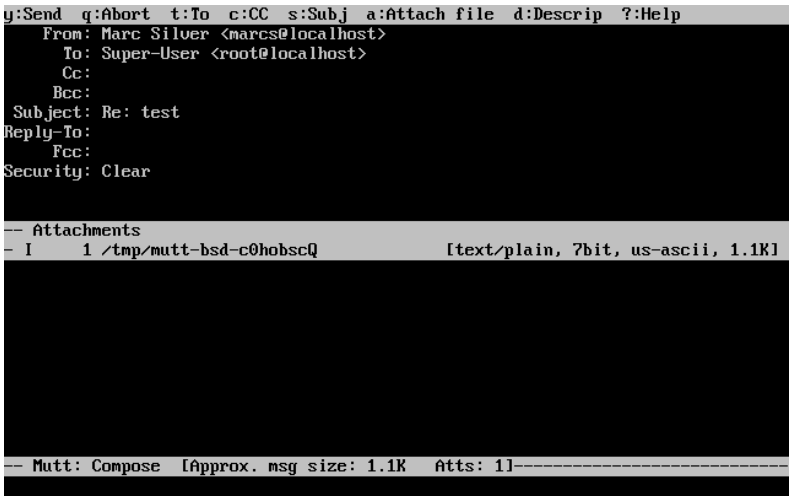
Wenn Sie eine Nachricht lesen wollen, wählen Sie die Nachricht einfach mit den Pfeiltasten aus und drücken **Enter**. **mutt** zeigt E-Mails wie folgt an:



Wenn Sie auf eine E-Mail antworten, können Sie, wie in mail(1), aussuchen, ob Sie nur dem Versender oder auch allen Empfängern antworten wollen. Wenn Sie nur dem Versender antworten wollen, drücken Sie die Taste **r**. Wenn sie dem Versender und allen Empfängern antworten wollen, drücken Sie die Taste **g**.

> **Anmerkung:** Zum Erstellen oder zum Beantworten von E-Mails ruft **mutt** den Editor vi(1) auf. Wenn Sie den von **mutt** verwendeten Editor ändern möchten, erstellen oder editieren Sie in Ihrem Heimatverzeichnis die Datei `.muttrc`. Den Editor können Sie in `.muttrc` mit der Variablen `editor` festlegen. Alternativ können Sie auch die Umgebungsvariable `EDITOR` setzen. Weitere Informationen zur Konfiguration von **mutt** finden Sie unter http://www.mutt.org/.

Drücken Sie die Taste **m**, wenn Sie eine neue Nachricht verfassen wollen. Nachdem Sie einen Betreff (*subject*) eingegeben haben, startet **mutt** den Editor vi(1) und Sie können die Nachricht eingeben. Wenn Sie fertig sind, speichern Sie die Nachricht und verlassen den Editor. **mutt** wird dann wieder aktiv und zeigt eine Zusammenfassung der zu sendenden Nachricht an. Drücken Sie **y**, um die E-Mail zu versenden. Der nachstehende Bildschirmabzug zeigt die Zusammenfassung der E-Mail:



mutt verfügt über eine umfangreiche Hilfestellung. Aus fast jedem Menü können Sie Hilfeseiten mit der Taste **?** aufrufen. In der oberen Statuszeile werden zudem die verfügbaren Tastenkombinationen angezeigt.

29.12.3. alpine

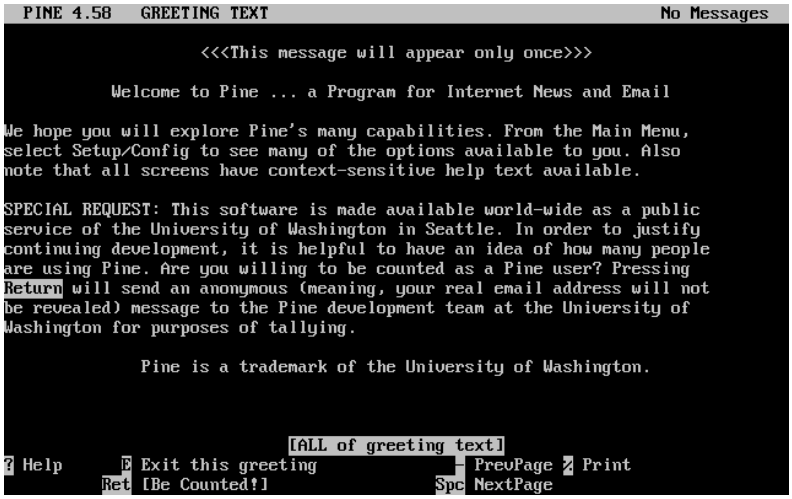
alpine wendet sich an Anfänger bietet aber ebenfalls einige Funktionen für Profis.

> **Warnung:** In der Vergangenheit wurden in **alpine** mehrere Schwachstellen gefunden. Die Schwachstellen gestatteten entfernten Benutzern, durch das Versenden einer besonders verfassten E-Mail, Programme auf dem lokalen System laufen zu lassen. Alle *bekannten* Schwachstellen sind beseitigt worden, doch wird im Quellcode von **alpine** ein sehr riskanter Programmierstil verwendet, sodass der FreeBSD-Security-Officer von weiteren unbekannten Schwachstellen ausgeht. Sie installieren **alpine** auf eigene Verantwortung!

Der Port `mail/alpine` enthält die aktuelle Version von **alpine**. Nach der Installation können Sie **alpine** mit dem nachstehenden Kommando starten:

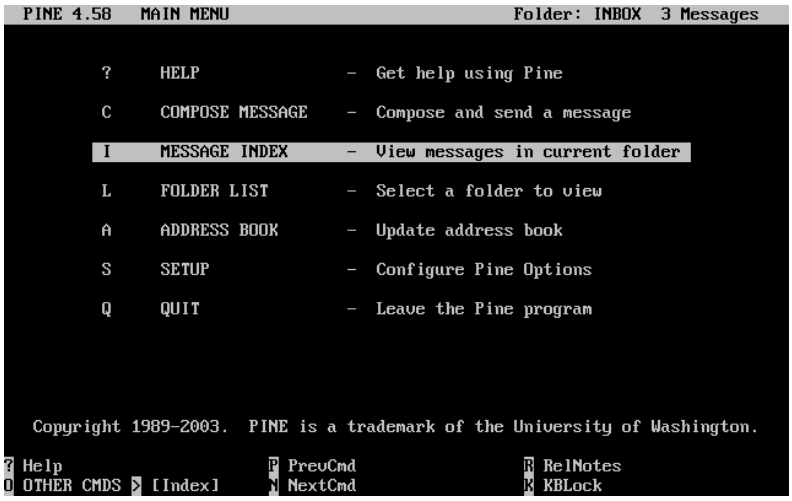
```
% alpine
```

Wenn Sie **alpine** das erste Mal starten, zeigt das Programm eine Seite mit einer kurzen Einführung an. Um die **alpine**-Benutzer zu zählen, bitten die Entwickler auf dieser Seite um eine anonyme E-Mail. Sie können diese anonyme E-Mail senden, indem Sie **Enter** drücken oder den Begrüßungsbildschirm mit der Taste **E** verlassen, ohne die anonyme E-Mail zu senden. Der Begrüßungsbildschirm sieht wie folgt aus:

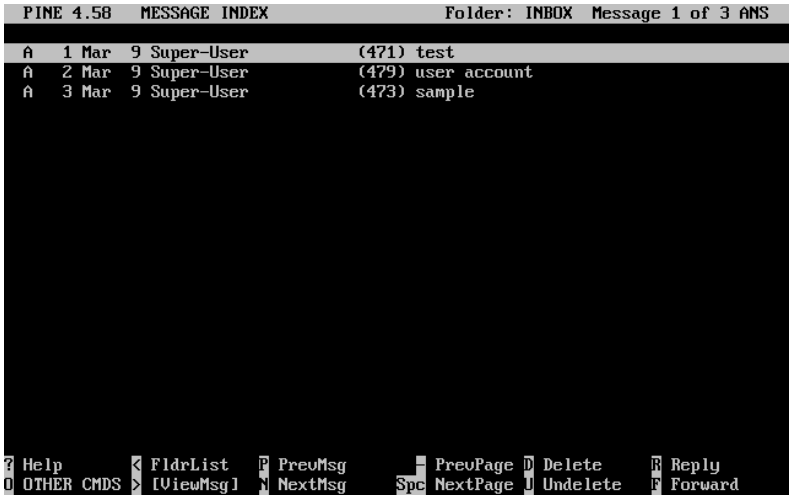


Nach dem Begrüßungsbildschirm wird das Hauptmenü dargestellt, das sich leicht mit den Pfeiltasten bedienen lässt. Mit Tastenkombinationen können Sie aus dem Hauptmenü neue E-Mails erstellen, Postfächer anzeigen und auch das Adressbuch verwalten. Unterhalb des Menüs werden die Tastenkombinationen für die verfügbaren Aktionen angezeigt.

In der Voreinstellung öffnet **pine** das Verzeichnis `inbox`. Die Taste **I** oder der Menüpunkt MESSAGE INDEX führt zu einer Nachrichten-Liste:

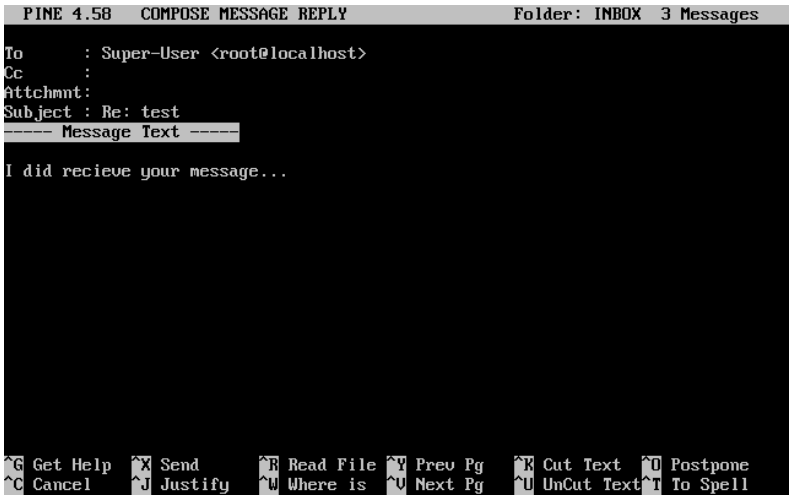


Die Liste zeigt die Nachrichten im Arbeitsverzeichnis. Sie können Nachrichten mit den Pfeiltasten markieren. Wenn Sie eine Nachricht lesen wollen, drücken Sie **Enter**.



Im nächsten Bildschirmabzug sehen Sie, wie **pine** eine Nachricht darstellt. Die unteren Bildschirmzeilen zeigen die verfügbaren Tastenkombinationen. Mit der Taste **r** können Sie zum Beispiel auf die gerade angezeigte Nachricht antworten.

Zum Antworten auf eine E-Mail wird in **pine** der Editor **pico**, der mit installiert wird, benutzt. **pico** ist leicht zu bedienen und gerade für Anfänger besser geeignet als vi(1) oder mail(1). Die Antwort wird mit der Tastenkombination **Ctrl**+**X** versendet. Vor dem Versand bittet **pine** noch um eine Bestätigung.



Über den Menüpunkt SETUP des Hauptmenüs können Sie **pine** an Ihre Bedürfnisse anpassen. Erläuterungen dazu finden Sie auf der Seite http://www.washington.edu/pine/.

29.13. E-Mails mit fetchmail abholen

Beigetragen von Marc Silver.

fetchmail ist ein vollwertiger IMAP- und POP-Client. Mit **fetchmail** können Benutzer E-Mails von entfernten IMAP- und POP-Servern in leichter zugängliche lokale Postfächer laden. **fetchmail** wird aus dem Port `mail/fetchmail` installiert. Das Programm bietet unter anderem folgende Funktionen:

- **fetchmail** beherrscht die Protokolle POP3, APOP, KPOP, IMAP, ETRN und ODMR.

- E-Mails können mit SMTP weiterverarbeitet werden. Dadurch ist garantiert, dass Filter, Weiterleitungen und Aliase weiterhin funktionieren.
- Das Programm kann als Dienst laufen und periodisch neue Nachrichten abrufen.
- **fetchmail** kann mehrere Postfächer abfragen und je nach Konfiguration die E-Mails an verschiedene lokale Benutzer zustellen.

Wegen des Funktionsumfangs von **fetchmail** können hier nur grundlegende Funktionen beschrieben werden. **fetchmail** benötigt die Konfigurationsdatei `.fetchmailrc`. In dieser Datei werden Informationen über Server wie auch Benutzerdaten und Passwörter hinterlegt. Wegen des kritischen Inhalts von `.fetchmailrc` sollte die Datei nur lesbar für den Benutzer sein:

```
% chmod 600 .fetchmailrc
```

Die folgende `.fetchmailrc` zeigt, wie das Postfach eines einzelnen Benutzers mit POP heruntergeladen wird. **fetchmail** wird angewiesen, eine Verbindung zu `example.com` herzustellen und sich dort als Benutzer `joesoap` mit dem Passwort `XXX` anzumelden. Das Beispiel setzt voraus, dass es der Benutzer `joesoap` auch auf dem lokalen System existiert.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

Im folgenden Beispiel werden mehrere POP- und IMAP-Server benutzt. Wo notwendig, werden E-Mails auf andere lokale Konten umgeleitet:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

Sie können **fetchmail** als Dienst starten. Verwenden Sie dazu die Kommandozeilenoption `-d` gefolgt von einer Zeitspanne in Sekunden, die angibt, wie oft die Server aus der Datei `.fetchmailrc` abgefragt werden sollen. Mit dem nachstehenden Befehl fragt **fetchmail** die Server alle 600 Sekunden ab:

```
% fetchmail -d 600
```

Mehr über **fetchmail** erfahren Sie auf der Seite http://fetchmail.berlios.de/.

29.14. E-Mails mit procmail filtern

Beigetragen von Marc Silver.

Mit **procmail** lässt sich eingehende E-Mail sehr gut filtern. Benutzer können Regeln für eingehende E-Mails definieren, die E-Mails zu anderen Postfächern oder anderen E-Mail-Adressen umleiten. **procmail** befindet sich im Port `mail/procmail`. **procmail** kann leicht in die meisten MTAs integriert werden. Lesen Sie dazu bitte die Dokumentation des verwendeten MTAs. Alternativ kann **procmail** in das E-Mail-System eingebunden werden, indem die nachstehende Zeile in die Datei `.forward` im Heimatverzeichnis eines Benutzers eingefügt wird:

```
"|exec /usr/local/bin/procmail || exit 75"
```

Im Folgenden zeigen wir einige einfache **procmail**-Regeln und beschreiben kurz den Zweck der Zweck der Regeln. Die Regeln müssen in die Datei `.procmailrc` im Heimatverzeichnis des Benutzers eingefügt werden.

Den Großteil dieser Regeln finden Sie auch in der Hilfeseite procmailex(5).

Alle E-Mail von <`user@example.com`> an die externe Adresse <`goodmail@example2.com`> weiterleiten:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Alle Nachrichten, die kürzer als 1000 Bytes sind, an <`goodmail@example2.com`> weiterleiten:

```
:0
* < 1000
! goodmail@example2.com
```

Jede E-Mail, die an <`alternate@example.com`> geschickt wurde, im Postfach `alternate` speichern:

```
:0
* ^TOalternate@example.com
alternate
```

Jede E-Mail, die im Betreff `Spam` enthält, nach `/dev/null` schieben:

```
:0
^Subject:.*Spam
/dev/null
```

Zuletzt ein nützliches Rezept, das eingehende E-Mails von den `FreeBSD.org`-Mailinglisten in ein separates Postfach für jede Liste einsortiert:

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
        LISTNAME=${MATCH}
        :0
        * LISTNAME??^\/[^@]+
        FreeBSD-${MATCH}
}
```

Fußnoten

1. Mailbox = Post- beziehungsweise Briefkasten

Kapitel 30. Netzwerkserver

Überarbeitet von Murray Stokely. Übersetzt von Johann Kois.

30.1. Übersicht

Dieses Kapitel beschreibt einige der häufiger verwendeten Netzwerkdienste auf UNIX-Systemen. Beschrieben werden Installation und Konfiguration sowie Test und Wartung verschiedener Netzwerkdienste. Zusätzlich sind im ganzen Kapitel Beispielkonfigurationsdateien vorhanden, von denen Sie sicherlich profitieren werden.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Den **inetd**-Daemon konfigurieren können.
- Wissen, wie man ein Netzwerkdateisystem einrichtet.
- Einen *Network Information Server* einrichten können, um damit Benutzerkonten im Netzwerk zu verteilen.
- Rechner durch Nutzung von DHCP automatisch für ein Netzwerk konfigurieren können.
- In der Lage sein, einen *Domain Name Server* einzurichten.
- Den **Apache** HTTP-Server konfigurieren können.
- Wissen, wie man einen *File Transfer Protocol* (FTP)-Server einrichtet.
- Mit **Samba** einen Datei- und Druckserver für Windows-Clients konfigurieren können.
- Unter Nutzung des NTP-Protokolls Datum und Uhrzeit synchronisieren sowie einen Zeitserver installieren können.
- Wissen, wie man den Standard-Protokollierungsdienst, `syslogd`, konfiguriert, um Protokolle von anderen Hosts zu akzeptieren.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Grundlagen der `/etc/rc`-Skripte verstanden haben.
- Mit der grundlegenden Netzwerkterminologie vertraut sein.
- Wissen, wie man zusätzliche Softwarepakete von Drittherstellern installiert (Kapitel 5).

30.2. Der inetd “Super-Server”

Beigetragen von Chern Lee. Aktualisiert vom FreeBSD Documentation Project.

30.2.1. Überblick

inetd(8) wird manchmal auch als “Internet Super-Server” bezeichnet, weil er Verbindungen für mehrere Dienste verwaltet. Wenn eine Verbindung eintrifft, bestimmt **inetd**, welches Programm für die eingetroffene Verbindung zuständig ist, aktiviert den entsprechenden Prozess und reicht den Socket an ihn weiter (der Socket dient dabei als Standardein- und -ausgabe sowie zur Fehlerbehandlung). Der Einsatz des **inetd**-Daemons an Stelle viele einzelner Daemonen kann auf nicht komplett ausgelasteten Servern zu einer Verringerung der Systemlast führen.

inetd wird vor allem dazu verwendet, andere Daemonen zu aktivieren, einige Protokolle werden aber auch direkt verwaltet. Dazu gehören **chargen**, **auth**, sowie **daytime**.

Dieser Abschnitt beschreibt die Konfiguration von **inetd** durch Kommandozeilenoptionen sowie die Konfigurationsdatei `/etc/inetd.conf`.

30.2.2. Einstellungen

inetd wird durch das rc(8)-System initialisiert. Die Option `inetd_enable` ist in der Voreinstellung zwar auf `NO` gesetzt, sie kann aber in Abhängigkeit von der vom Benutzer bei der Installation gewählten Konfiguration von **sysinstall** aktiviert werden. Die Verwendung von

```
inetd_enable="YES"
```

oder

```
inetd_enable="NO"
```

in `/etc/rc.conf` deaktiviert oder startet **inetd** beim Systemstart. Über den Befehl

```
# /etc/rc.d/inetd rcvar
```

können Sie die aktuelle Konfiguration abfragen.

Weitere Optionen können über die Option `inetd_flags` an **inetd** übergeben werden.

30.2.3. Kommandozeilenoptionen

Wie die meisten anderen Server-Daemonen lässt sich auch **inetd** über verschiedene Optionen steuern. Die vollständige Syntax für **inetd** lautet:

```
inetd [-d] [-l] [-w] [-W] [-c maximum] [-C rate] [-a address | hostname] [-p filename]
[-R rate] [-s maximum] [configuration file]
```

Die verschiedenen Optionen können über die Option `inetd_flags` der Datei `/etc/rc.conf` an **inetd** übergeben werden. In der Voreinstellung hat diese Option den Wert `-wW -C 60`. Durch das Setzen dieser Werte wird das TCP-Wrapping für alle **inetd**-Dienste aktiviert. Zusätzlich kann eine einzelne IP-Adresse jeden Dienst nur maximal 60 Mal pro Minute anfordern.

Für Einsteiger ist es erfreulich, dass diese Parameter in der Regel nicht angepasst werden müssen. Da diese Parameter aber dennoch von Interesse sein können (beispielsweise, wenn Sie eine enorme Anzahl von Verbindungsanfragen erhalten), werden einige dieser einschränkenden Parameter im Folgenden näher erläutert. Eine vollständige Auflistung aller Optionen finden Sie hingegen in inetd(8).

-c maximum

> Legt die maximale Anzahl von parallen Aufrufen eines Dienstes fest; in der Voreinstellung gibt es keine Einschränkung. Diese Einstellung kann für jeden Dienst durch Setzen des `max-child` -Parameters festgelegt werden.

-C rate

Legt fest, wie oft ein Dienst von einer einzelnen IP-Adresse in einer Minute aufgerufen werden kann; in der Voreinstellung gibt es keine Einschränkung. Dieser Wert kann für jeden Dienst durch Setzen des Parameters `max-connections-per-ip-per-minute` festgelegt werden.

-R rate

Legt fest, wie oft ein Dienst in der Minute aktiviert werden kann; in der Voreinstellung sind dies 256 Aktivierungen pro Minute. Ein Wert von 0 erlaubt unbegrenzt viele Aktivierungen.

-s maximum

Legt fest, wie oft ein Dienst in der Minute von einer einzelnen IP-Adresse aus aktiviert werden kann; in der Voreinstellung gibt es hier keine Beschränkung. Diese Einstellung kann für jeden Dienst durch die Angabe `max-child-per-ip` angepasst werden.

30.2.4. `inetd.conf`

Die Konfiguration von **inetd** erfolgt über die Datei `/etc/inetd.conf`.

Wenn `/etc/inetd.conf` geändert wird, kann **inetd** veranlasst werden, seine Konfigurationsdatei neu einzulesen.

Beispiel 30-1. Die inetd-Konfiguration neu einlesen

```
# /etc/rc.d/inetd reload
```

Jede Zeile der Konfigurationsdatei beschreibt jeweils einen Daemon. Kommentare beginnen mit einem “#”. Ein Eintrag der Datei `/etc/inetd.conf` hat folgenden Aufbau:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][/login-class]
server-program
server-program-arguments
```

Ein Eintrag für den IPv4 verwendenden ftpd(8)-Daemon könnte so aussehen:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

service-name

Der Dienstname eines bestimmten Daemons. Er muss einem in `/etc/services` aufgelisteten Dienst entsprechen. In dieser Datei wird festgelegt, welchen Port **inetd** abhören muss. Wenn ein neuer Dienst erzeugt wird, muss er zuerst in die Datei `/etc/services` eingetragen werden.

socket-type

Entweder `stream`, `dgram`, `raw`, oder `seqpacket`. `stream` muss für verbindungsorientierte TCP-Daemonen verwendet werden, während `dgram` das UDP-Protokoll verwaltet.

protocol

Eines der folgenden:

Protokoll	Bedeutung
tcp, tcp4	TCP (IPv4)
udp, udp4	UDP (IPv4)
tcp6	TCP (IPv6)
udp6	UDP (IPv6)
tcp46	TCP sowohl unter IPv4 als auch unter IPv6
udp46	UDP sowohl unter IPv4 als auch unter IPv6

{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]

`wait|nowait` gibt an, ob der von **inetd** aktivierte Daemon seinen eigenen Socket verwalten kann oder nicht. `dgram`-Sockets müssen die Option `wait` verwenden, während Daemonen mit Stream-Sockets, die normalerweise auch aus mehreren Threads bestehen, die Option `nowait` verwenden sollten. Die Option `wait` gibt in der Regel mehrere Sockets an einen einzelnen Daemon weiter, während `nowait` für jeden neuen Socket einen Childdaemon erzeugt.

Die maximale Anzahl an Child-Daemonen, die **inetd** erzeugen kann, wird durch die Option `max-child` festgelegt. Wenn ein bestimmter Daemon 10 Instanzen benötigt, sollte der Wert `/10` hinter die Option `nowait` gesetzt werden. Geben Sie hingegen den Wert `/0` an, gibt es keine Beschränkung.

Zusätzlich zu `max-child` kann die maximale Anzahl von Verbindungen eines Rechners mit einem bestimmten Daemon durch zwei weitere Optionen beschränkt werden. Die Option `max-connections-per-ip-per-minute` legt die maximale Anzahl von Verbindungsversuchen fest, die von einer bestimmten IP-Adresse aus unternommen werden können. Ein Wert von zehn würde die maximale Anzahl von Verbindungsversuchen einer IP-Adresse mit einem bestimmten Dienst auf zehn Versuche in der Minute beschränken. Durch die Angabe der Option `max-child-per-ip` können Sie hingegen festlegen, wie viele Child-Daemonen von einer bestimmten IP-Adresse aus gestartet werden können. Durch diese Optionen lassen sich ein absichtlicher oder unabsichtlicher Ressourcenverbrauch sowie die Auswirkungen eines `Denial of Service (DoS)`-Angriffs auf einen Rechner begrenzen.

Sie müssen hier entweder `wait` oder `nowait` angeben. Die Angabe von `max-child`, `max-connections-per-ip-per-minute` und `max-child-per-ip` ist hingegen optional.

Ein multithread-Daemon vom Streamtyp ohne die Optionen `max-child`, `max-connections-per-ip-per-minute` oder `max-child-per-ip` sieht so aus: `nowait`

Der gleiche Daemon mit einer maximal möglichen Anzahl von 10 parallelen Daemonen würde so aussehen: `nowait/10`

Wird zusätzlich die Anzahl der möglichen Verbindungen pro Minute für jede IP-Adresse auf 20 sowie die mögliche Gesamtzahl von Childdaemonen auf 10 begrenzt, so sieht der Eintrag so aus: `nowait/10/20`

All diese Optionen werden vom fingerd(8)-Daemon bereits in der Voreinstellung verwendet:

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -s
```

Will man die maximale Anzahl von Child-Daemonen auf 100 beschränken, wobei von jeder IP-Adresse aus maximal 5 Child-Daemonen gestartet werden dürfen, verwendet man den folgenden Eintrag:

`nowait/100/0/5`.

user

Der Benutzername, unter dem der jeweilige Daemon laufen soll. Meistens laufen Daemonen als User `root`. Aus Sicherheitsgründen laufen einige Server aber auch als User `daemon`, oder als am wenigsten privilegierter User `nobody`.

server-program

Der vollständige Pfad des Daemons, der eine Verbindung entgegennimmt. Wird der Daemon von **inetd** intern bereitgestellt, sollte die Option `internal` verwendet werden.

server-program-arguments

Dieser Eintrag legt (gemeinsam mit `server-program` und beginnend mit `argv[0]`), die Argumente fest, die bei der Aktivierung an den Daemon übergeben werden. Wenn die Anweisung auf der Kommandozeile also `mydaemon -d` lautet, wäre `mydaemon -d` auch der Wert der Option `server program arguments`. Wenn es sich beim Daemon um einen internen Dienst handelt, sollte wiederum die Option `internal` verwendet werden.

30.2.5. Sicherheit

Abhängig von der bei der Installation festgelegten Konfiguration werden viele der von **inetd** verwalteten Dienste automatisch aktiviert! Wenn Sie einen bestimmten Daemon nicht benötigen, sollten Sie ihn deaktivieren! Dazu kommentieren Sie den jeweiligen Daemon in `/etc/inetd.conf` mit einem "#" aus, um danach die inetd-Konfiguration neu einzulesen. Einige Daemonen, zum Beispiel **fingerd**, sollten generell deaktiviert werden, da sie zu viele Informationen an einen potentiellen Angreifer liefern.

Einige Daemonen haben unsichere Einstellungen, etwa große oder nichtexistierende Timeouts für Verbindungsversuche, die es einem Angreifer erlauben, über lange Zeit langsam Verbindungen zu einem bestimmten Daemon aufzubauen, um dessen verfügbare Ressourcen zu verbrauchen. Es ist daher eine gute Idee, diese Daemonen durch die Optionen `max-connections-per-ip-per-minute`, `max-child` sowie `max-child-per-ip` zu beschränken, wenn Sie sehr viele Verbindungsversuche mit Ihrem System registrieren.

TCP-Wrapping ist in der Voreinstellung aktiviert. Lesen Sie hosts_access(5), wenn Sie weitere Informationen zum Setzen von TCP-Beschränkungen für verschiedene von **inetd** aktivierte Daemonen benötigen.

30.2.6. Verschiedenes

Bei **daytime**, **time**, **echo**, **discard**, **chargen**, und **auth** handelt es sich um intern von **inetd** bereitgestellte Dienste.

Der **auth**-Dienst bietet Identifizierungsdienste über das Netzwerk an und ist bis zu einem bestimmten Grad konfigurierbar, während die meisten anderen Dienste nur aktiviert oder deaktiviert werden können.

Eine ausführliche Beschreibung finden Sie in inetd(8).

30.3. NFS – Network File System

Reorganisiert und erweitert von Tom Rhodes. Geschrieben von Bill Swingle.

Eines der vielen von FreeBSD unterstützten Dateisysteme ist das Netzwerkdateisystem, das auch als NFS bekannt ist. NFS ermöglicht es einem System, Dateien und Verzeichnisse über ein Netzwerk mit anderen zu teilen. Über NFS können Benutzer und Programme auf Daten entfernter Systeme zugreifen, und zwar genauso, wie wenn es sich um lokale Daten handeln würde.

Einige der wichtigsten Vorteile von NFS sind:

- Lokale Arbeitsstationen benötigen weniger Plattenplatz, da gemeinsam benutzte Daten nur auf einem einzigen Rechner vorhanden sind. Alle anderen Stationen greifen über das Netzwerk auf diese Daten zu.
- Benutzer benötigen nur noch ein zentrales Heimatverzeichnis auf einem NFS-Server. Diese Verzeichnisse sind über das Netzwerk auf allen Stationen verfügbar.
- Speichergeräte wie Disketten-, CD-ROM- oder Zip®-Laufwerke können über das Netzwerk von anderen Arbeitstationen genutzt werden. Dadurch sind für das gesamte Netzwerk deutlich weniger Speichergeräte nötig.

30.3.1. Wie funktioniert NFS?

NFS besteht aus zwei Hauptteilen: Einem Server und einem oder mehreren Clients. Der Client greift über das Netzwerk auf die Daten zu, die auf dem Server gespeichert sind. Damit dies korrekt funktioniert, müssen einige Prozesse konfiguriert und gestartet werden:

Der Server benötigt folgende Daemonen:

Daemon	Beschreibung
nfsd	Der NFS-Daemon. Er bearbeitet Anfragen der NFS-Clients.
mountd	Der NFS-Mount-Daemon. Er bearbeitet die Anfragen, die nfsd(8) an ihn weitergibt.
rpcbind	Der Portmapper-Daemon. Durch ihn erkennen die NFS-Clients, welchen Port der NFS-Server verwendet.

Der Client kann ebenfalls einen Daemon aufrufen, und zwar den **nfsiod**-Daemon. Der **nfsiod**-Daemon bearbeitet Anfragen vom NFS-Server. Er ist optional und verbessert die Leistung des Netzwerks. Für eine normale und korrekte Arbeit ist er allerdings nicht erforderlich. Mehr erfahren Sie in der Hilfeseite nfsiod(8).

30.3.2. NFS einrichten

NFS lässt sich leicht einrichten. Die nötigen Prozesse werden durch einige Änderungen in `/etc/rc.conf` bei jedem Systemstart gestartet.

Stellen Sie sicher, dass auf dem NFS-Server folgende Optionen in der Datei `/etc/rc.conf` gesetzt sind:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

mountd läuft automatisch, wenn der NFS-Server aktiviert ist.

Auf dem Client muss in `/etc/rc.conf` folgende Option gesetzt sein:

```
nfs_client_enable="YES"
```

`/etc/exports` legt fest, welche Dateisysteme NFS exportieren (manchmal auch als “teilen” bezeichnet) soll. Jede Zeile in `/etc/exports` legt ein Dateisystem sowie die Arbeitsstationen, die darauf Zugriff haben, fest. Außerdem ist es möglich, Zugriffsoptionen festzulegen. Es gibt viele verschiedene Optionen, allerdings werden hier nur einige von ihnen erwähnt. Wenn Sie Informationen zu weiteren Optionen benötigen, lesen Sie exports(5).

Nun folgen einige Beispieleinträge für `/etc/exports`:

Die folgenden Beispiele geben Ihnen Anhaltspunkte zum Exportieren von Dateisystemen, obwohl diese Einstellungen natürlich von Ihrer Arbeitsumgebung und Ihrer Netzwerkkonfiguration abhängen. Das nächste Beispiel exportiert das Verzeichnis `/cdrom` für drei Rechner, die sich in derselben Domäne wie der Server befinden oder für die entsprechende Einträge in `/etc/hosts` existieren. Die Option `-ro` kennzeichnet das exportierte Dateisystem als schreibgeschützt. Durch dieses Flag ist das entfernte System nicht in der Lage, das exportierte Dateisystem zu verändern.

```
/cdrom -ro host1 host2 host3
```

Die nächste Zeile exportiert `/home` auf drei durch IP-Adressen bestimmte Rechner. Diese Einstellung ist nützlich, wenn Sie über ein privates Netzwerk ohne DNS-Server verfügen. Optional könnten interne Rechnernamen auch in `/etc/hosts` konfiguriert werden. Benötigen Sie hierzu weitere Informationen, lesen Sie bitte hosts(5). Durch das Flag `-alldirs` wird es möglich, auch Unterverzeichnisse als Mountpunkte festzulegen. Dies bedeutet aber nicht, dass alle Unterverzeichnisse eingehängt werden, vielmehr wird es dem Client ermöglicht, nur diejenigen Verzeichnisse einzuhängen, die auch benötigt werden.

```
/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

Die nächste Zeile exportiert `/a`, damit Clients von verschiedenen Domänen auf das Dateisystem zugreifen können. Das `-maproot=root`-Flag erlaubt es dem Benutzer `root` des entfernten Systems, als `root` auf das exportierte Dateisystem zu schreiben. Wenn dieses Flag nicht gesetzt ist, kann selbst `root` nicht auf das exportierte Dateisystem schreiben.

```
/a  -maproot=root  host.example.com box.example.org
```

Damit ein Client auf ein exportiertes Dateisystem zugreifen kann, muss ihm dies explizit gestattet werden. Stellen Sie also sicher, dass der Client in `/etc/exports` aufgeführt wird.

Jede Zeile in `/etc/exports` entspricht der Exportinformation für ein Dateisystem auf einen Rechner. Ein entfernter Rechner kann für jedes Dateisystem nur einmal festgelegt werden, und kann auch nur einen Standardeintrag haben. Nehmen wir an, dass `/usr` ein einziges Dateisystem ist. Dann wären folgende Zeilen ungültig:

```
#Nicht erlaubt, wenn /usr ein einziges Dateisystem ist
/usr/src   client
/usr/ports client
```

Das Dateisystem `/usr` wird hier zweimal auf den selben Rechner (`client`) exportiert. Dies ist aber nicht zulässig. Der korrekte Eintrag sieht daher so aus:

```
/usr/src /usr/ports  client
```

Die Eigenschaften eines auf einen anderen Rechner exportierten Dateisystems müssen alle in einer Zeile stehen. Zeilen, in denen kein Rechner festgelegt wird, werden als einzelner Rechner behandelt. Dies schränkt die Möglichkeiten zum Export von Dateisystemen ein, für die meisten Anwender ist dies aber kein Problem.

Eine gültige Exportliste, in der `/usr` und `/exports` lokale Dateisysteme sind, sieht so aus:

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports                  client02
# The client machines have root and can mount anywhere
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

Der Daemon **mountd** muss die Datei `/etc/exports` nach jeder Änderung neu einlesen, damit die Änderungen wirksam werden. Dies kann durch das Senden des HUP-Signals an den `mountd`-Prozess erfolgen:

```
# kill -HUP ‘cat /var/run/mountd.pid‘
```

Alternativ können Sie das `mountd`-rc(8)-Skript auch mit dem passenden Parameter aufrufen:

```
# /etc/rc.d/mountd onereload
```

Lesen Sie bitte Abschnitt 12.7 des Handbuchs für Informationen zum Einsatz der rc-Skripte.

Eine weitere Möglichkeit, diese Änderungen zu übernehmen, wäre der Neustart des Systems. Dies ist allerdings nicht nötig. Wenn Sie die folgenden Befehle als `root` ausführen, sollte alles korrekt gestartet werden.

Auf dem NFS-Server:

```
# rpcbind
# nfsd -u -t -n 4
# mountd -r
```

Auf dem NFS-Client:

```
# nfsiod -n 4
```

Nun sollte alles bereit sein, um ein entferntes Dateisystem einhängen zu können. In unseren Beispielen nennen wir den Server `server`, den Client `client`. Wenn Sie ein entferntes Dateisystem nur zeitweise einhängen wollen, oder nur Ihre Konfiguration testen möchten, führen Sie auf dem Client als `root` einen Befehl ähnlich dem folgenden aus:

```
# mount server:/home /mnt
```

Dadurch wird das Verzeichnis `/home` des Servers auf dem Client unter `/mnt` eingehängt. Wenn alles korrekt konfiguriert wurde, sehen Sie auf dem Client im Verzeichnis `/mnt` alle Dateien des Servers.

Wenn Sie ein entferntes Dateisystem nach jedem Systemstart automatisch einhängen wollen, fügen Sie das Dateisystem in `/etc/fstab` ein. Dazu ein Beispiel:

```
server:/home    /mnt    nfs     rw      0       0
```

Eine Beschreibung aller Optionen enthält die Hilfeseite fstab(5).

30.3.3. Dateien sperren (*Locking*)

Einige Anwendungen (beispielsweise **mutt**) erfordern die Sperrung von Dateien, damit sie korrekt arbeiten. Verwenden Sie NFS, so können Sie für die Sperrung von Dateien **rpc.lockd** einsetzen. Um diesen Daemon zu aktivieren, müssen Sie in `/etc/rc.conf` (sowohl auf Client- als auch auf Serverseite) folgende Zeilen aufnehmen (wobei vorausgesetzt wird, dasss NFS auf beiden Systemen bereits konfiguriert ist):

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Danach starten Sie die Anwendung zur Verwaltung der Dateisperren durch folgenden Befehl:

```
# /etc/rc.d/lockd start
# /etc/rc.d/statd start
```

Benötigen Sie keine echten Dateisperren zwischen den NFS-Clients und dem NFS-Server, können Sie den NFS-Client durch die Übergabe der Option `-L` an mount_nfs(8) zu einer lokalen Sperrung von Dateien zwingen. Lesen Sie dazu auch die Manualpage mount_nfs(8).

30.3.4. Praktische Anwendungen

NFS ist in vielen Situationen nützlich. Einige Anwendungsbereiche finden Sie in der folgenden Liste:

- Mehrere Maschinen können sich ein CD-ROM-Laufwerk oder andere Medien teilen. Dies ist billiger und außerdem praktischer, um Programme auf mehreren Rechnern zu installieren.
- In größeren Netzwerken ist es praktisch, einen zentralen NFS-Server einzurichten, auf dem die Heimatverzeichnisse der Benutzer gespeichert werden. Diese Heimatverzeichnisse werden über das Netzwerk exportiert. Dadurch haben die Benutzer immer das gleiche Heimatverzeichnis zur Verfügung, unabhängig davon, an welchem Arbeitsplatz sie sich anmelden.
- Verschiedene Rechner können auf ein gemeinsames Verzeichnis `/usr/ports/distfiles` zugreifen. Wenn Sie nun einen Port auf mehreren Rechnern installieren wollen, greifen Sie einfach auf dieses Verzeichnis zu, ohne die Quelldateien auf jede Maschine zu kopieren.

30.3.5. AMD

Beigetragen von Wylie Stilwell. Überarbeitet von Chern Lee.

amd(8) (Automatic Mounter Daemon) hängt ein entferntes Dateisystem automatisch ein, wenn auf eine Datei oder ein Verzeichnis in diesem Dateisystem zugegriffen wird. Dateisysteme, die über einen gewissen Zeitraum inaktiv sind, werden von **amd** automatisch abgehängt. **amd** ist eine einfache Alternative zum dauerhaften Einhängen von Dateisystemen in `/etc/fstab`.

In der Voreinstellung stellt **amd** die Verzeichnisse `/host` und `/net` als NFS-Server bereit. Wenn auf eine Datei in diesen Verzeichnissen zugegriffen wird, sucht **amd** den entsprechenden Mountpunkt und hängt das Dateisystem automatisch ein. `/net` wird zum Einhängen von exportierten Dateisystemen von einer IP-Adresse verwendet, während `/host` zum Einhängen von exportierten Dateisystemen eines durch seinen Namen festgelegten Rechners dient.

Ein Zugriff auf eine Datei in `/host/foobar/usr` würde **amd** veranlassen, das von `foobar` exportierte Dateisystem `/usr` einzuhängen.

Beispiel 30-2. Ein exportiertes Dateisystem mit amd in den Verzeichnisbaum einhängen

Sie können sich die verfügbaren Mountpunkte eines entfernten Rechners mit `showmount` ansehen. Wollen Sie sich die Mountpunkte des Rechners `foobar` ansehen, so verwenden Sie:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /host/foobar/usr
```

Wie Sie an diesem Beispiel erkennen können, zeigt `showmount /usr` als exportiertes Dateisystem an. Wenn man in das Verzeichnis `/host/foobar/usr` wechselt, versucht **amd** den Rechnernamen `foobar` aufzulösen und den gewünschten Export in den Verzeichnisbaum einzuhängen.

amd kann durch das Einfügen der folgenden Zeile in `/etc/rc.conf` automatisch gestartet werden:

```
amd_enable="YES"
```

Mit der Option `amd_flags` kann **amd** angepasst werden. Die Voreinstellung für `amd_flags` sieht so aus:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map"
```

`/etc/amd.map` legt die Standardoptionen fest, mit denen exportierte Dateisysteme in den Verzeichnisbaum eingehängt werden. `/etc/amd.conf` hingegen legt einige der erweiterten Optionen von **amd** fest.

Weitere Informationen finden Sie in den Hilfeseiten amd(8) und amd.conf(5).

30.3.6. Integrationsprobleme mit anderen Systemen

Beigetragen von John Lind.

Bestimmte ISA-Ethernetadapter haben Beschränkungen, die zu ernsthaften Netzwerkproblemen, insbesondere mit NFS führen können. Es handelt sich dabei nicht um ein FreeBSD-spezifisches Problem, aber FreeBSD-Systeme sind davon ebenfalls betroffen.

Das Problem tritt fast ausschließlich dann auf, wenn (FreeBSD)-PC-Systeme mit Hochleistungsrechnern verbunden werden, wie Systemen von Silicon Graphics, Inc. oder Sun Microsystems, Inc. Das Einhängen via NFS funktioniert problemlos, auch einige Dateioperationen können erfolgreich sein. Plötzlich aber wird der Server nicht mehr auf den Client reagieren, obwohl Anfragen von anderen Rechnern weiterhin bearbeitet werden. Dieses Problem betrifft stets den Client, egal ob es sich beim Client um das FreeBSD-System oder den Hochleistungsrechner handelt. Auf vielen Systemen gibt es keine Möglichkeit mehr, den Client ordnungsgemäß zu beenden. Die einzige Lösung ist es oft, den Rechner neu zu starten, da dieses NFS-Problem nicht mehr behoben werden kann.

Die “korrekte” Lösung für dieses Problem ist es, sich eine schnellere Ethernetkarte für FreeBSD zu kaufen. Allerdings gibt es auch eine einfache und meist zufriedenstellende Lösung, um dieses Problem zu umgehen. Wenn es sich beim FreeBSD-System um den *Server* handelt, verwenden Sie beim Einhängen in den Verzeichnisbaum auf der Clientseite zusätzlich die Option `-w=1024` . Wenn es sich beim FreeBSD-System um den *Client* handelt, dann hängen Sie das NFS-Dateisystem mit der zusätzlichen Option `-r=1024` ein. Diese Optionen können auf der Clientseite auch durch das vierte Feld der Einträge in `/etc/fstab` festgelegt werden, damit die Dateisysteme

automatisch eingehängt werden. Um die Dateisysteme manuell einzuhängen, verwendet man bei mount(8) zusätzlich die Option `-o`.

Es gibt ein anderes Problem, das oft mit diesem verwechselt wird. Dieses andere Problem tritt auf, wenn sich über NFS verbundene Server und Clients in verschiedenen Netzwerken befinden. Wenn dies der Fall ist, stellen Sie *sicher*, dass Ihre Router die nötigen UDP-Informationen weiterleiten, oder Sie werden nirgends hingelangen, egal was Sie machen.

In den folgenden Beispielen ist `fastws` der Name des Hochleistungsrechners (bzw. dessen Schnittstelle), `freebox` hingegen ist der Name des FreeBSD-Systems, das über eine Netzkarte mit geringer Leistung verfügt. `/sharedfs` ist das exportierte NFS -Dateisystem (lesen Sie dazu auch exports(5)). Bei `/project` handelt es sich um den Mountpunkt, an dem das exportierte Dateisystem auf der Clientseite eingehängt wird. In allen Fällen können zusätzliche Optionen, wie z.B. `hard`, `soft` oder `bg` wünschenswert sein.

FreeBSD als Client (eingetragen in `/etc/fstab` auf `freebox`):

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

Manuelles Einhängen auf `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

FreeBSD als Server (eingetragen in `/etc/fstab` auf `fastws`):

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

Manuelles Einhängen auf `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Nahezu alle 16-bit Ethernetadapter erlauben Operationen ohne obengenannte Einschränkungen auf die Lese- oder Schreibgröße.

Für alle technisch Interessierten wird nun beschrieben, was passiert, wenn dieser Fehler auftritt, und warum er irreversibel ist. NFS arbeitet üblicherweise mit einer “Blockgröße” von 8 kByte (obwohl es kleinere Fragmente zulassen würde). Da die maximale Rahmengröße von Ethernet 1500 Bytes beträgt, wird der NFS-“Block” in einzelne Ethernetrahmen aufgeteilt, obwohl es sich nach wie vor um eine Einheit handelt, die auch als Einheit empfangen, verarbeitet und *bestätigt* werden muss. Der Hochleistungsrechner verschickt die Pakete, aus denen der NFS-Block besteht, so eng hintereinander, wie es der Standard erlaubt. Auf der anderen Seite (auf der sich die langsamere Netzkarte befindet), überschreiben die späteren Pakete ihre Vorgänger, bevor diese vom System verarbeitet werden (Überlauf!). Dies hat zur Folge, dass der NFS-Block nicht mehr rekonstruiert und bestätigt werden kann. Als Folge davon glaubt der Hochleistungsrechner, dass der andere Rechner nicht erreichbar ist (Timeout!) und versucht die Sendung zu wiederholen. Allerdings wird wiederum der komplette NFS-Block verschickt, so dass sich der ganze Vorgang wiederholt, und zwar immer wieder (oder bis zum Systemneustart).

Indem wir die Einheitengröße unter der maximalen Größe der Ethernetpakete halten, können wir sicherstellen, dass jedes vollständig erhaltene Ethernetpaket individuell angesprochen werden kann und vermeiden die Blockierung des Systems.

Überläufe können zwar nach wie vor auftreten, wenn ein Hochleistungsrechner Daten auf ein PC-System transferiert. Durch die besseren (und schnelleren) Netzkarten treten solche Überläufe allerdings nicht mehr *zwingend* auf, wenn NFS-“Einheiten” übertragen werden. Tritt nun ein Überlauf auf, wird die betroffene Einheit erneut verschickt, und es besteht eine gute Chance, dass sie nun erhalten, verarbeitet und bestätigt werden kann.

30.4. NIS/YP – Network Information Service

Beigetragen von Bill Swingle. Erweitert von Eric Ogren und Udo Erdelhoff.

30.4.1. Was ist NIS?

NIS wurde von Sun Microsystems entwickelt, um UNIX-Systeme (ursprünglich SunOS) zentral verwalten zu können. Mittlerweile hat es sich zu einem Industriestandard entwickelt, der von allen wichtigen UNIX-Systemen (Solaris, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD und anderen) unterstützt wird.

NIS war ursprünglich als *Yellow Pages* bekannt, aus markenrechtlichen Gründen wurde der Name aber geändert. Die alte Bezeichnung (sowie die Abkürzung YP) wird aber nach wie vor häufig verwendet.

Bei NIS handelt es sich um ein RPC-basiertes Client/Server-System. Eine Gruppe von Rechnern greift dabei innerhalb einer NIS-Domäne auf gemeinsame Konfigurationsdateien zu. Ein Systemadministrator wird dadurch in die Lage versetzt, NIS-Clients mit minimalem Aufwand einzurichten, sowie Änderungen an der Systemkonfiguration von einem zentralen Ort aus durchzuführen.

Die Funktion entspricht dem Domänensystem von Windows NT®; auch wenn sich die interne Umsetzung unterscheidet, sind die Basisfunktionen vergleichbar.

30.4.2. Wichtige Prozesse und Begriffe

Es gibt verschiedene Begriffe und Anwenderprozesse, auf die Sie stoßen werden, wenn Sie NIS unter FreeBSD einrichten, egal ob Sie einen Server oder einen Client konfigurieren:

Begriff	Beschreibung
NIS-Domänenname	Ein NIS-Masterserver sowie alle Clients (inklusive der Slaveserver) haben einen NIS-Domänennamen. Dieser hat (ähnlich den Windows NT-Domänennamen) nichts mit DNS zu tun.
rpcbind	Muss laufen, damit RPC (Remote Procedure Call, ein von NIS verwendetes Netzwerkprotokoll) funktioniert. NIS-Server sowie Clients funktionieren ohne **rpcbind** nicht.
ypbind	“Bindet” einen NIS-Client an seinen NIS-Server. Der Client bezieht den NIS-Domänennamen vom System und stellt über das RPC-Protokoll eine Verbindung zum NIS-Server her. **ypbind** ist der zentrale Bestandteil der Client-Server-Kommunikation in einer NIS-Umgebung. Wird **>ypbind** auf einem Client beendet, ist dieser nicht mehr in der Lage, auf den NIS-Server zuzugreifen.
ypserv	Sollte nur auf dem NIS-Server laufen, da es sich um den Serverprozess selbst handelt. Wenn ypserv(8) nicht mehr läuft, kann der Server nicht mehr auf NIS-Anforderungen reagieren (wenn ein Slaveserver existiert, kann dieser als Ersatz fungieren). Einige NIS-Systeme (allerdings nicht das von FreeBSD) versuchen allerdings erst gar nicht, sich mit einem anderen Server zu verbinden, wenn der bisher verwendete Server nicht mehr reagiert. Die einzige Lösung dieses Problems besteht dann darin, den Serverprozess (oder gar den Server selbst) oder den **ypbind**-Prozess auf dem Client neu zu starten.

Begriff	Beschreibung
rpc.yppasswdd	Ein weiterer Prozess, der nur auf dem NIS-Masterserver laufen sollte. Es handelt sich um einen Daemonprozess, der es NIS-Clients ermöglicht, sich auf dem NIS-Masterserver anzumelden, um ihr Passwort zu ändern.

30.4.3. Wie funktioniert NIS?

In einer NIS-Umgebung gibt es drei Rechnerarten: Masterserver, Slaveserver und Clients. Server dienen als zentraler Speicherort für Rechnerkonfigurationen. Masterserver speichern die maßgebliche Kopie dieser Informationen, während Slaveserver diese Informationen aus Redundanzgründen spiegeln. Die Clients beziehen ihre Informationen immer vom Server.

Auf diese Art und Weise können Informationen aus verschiedenen Dateien von mehreren Rechnern gemeinsam verwendet werden. `master.passwd`, `group`, und `hosts` werden oft gemeinsam über NIS verwendet. Immer, wenn ein Prozess auf einem Client auf Informationen zugreifen will, die normalerweise in lokalen Dateien vorhanden wären, wird stattdessen eine Anfrage an den NIS-Server gestellt, an den der Client gebunden ist.

30.4.3.1. Arten von NIS-Rechnern

- Ein *NIS-Masterserver* verwaltet, ähnlich einem Windows NT-Domänencontroller, die von allen NIS-Clients gemeinsam verwendeten Dateien. `passwd`, `group`, sowie verschiedene andere von den Clients verwendete Dateien existieren auf dem Masterserver.

  Anmerkung: Ein Rechner kann auch für mehrere NIS-Domänen als Masterserver fungieren. Dieser Abschnitt konzentriert sich im Folgenden allerdings auf eine relativ kleine NIS-Umgebung.

- *NIS-Slaveserver*. Ähnlich einem Windows NT-Backupdomänencontroller, verwalten NIS-Slaveserver Kopien der Daten des NIS-Masterservers. NIS-Slaveserver bieten die Redundanz, die für kritische Umgebungen benötigt wird. Zusätzlich entlasten Slaveserver den Masterserver: NIS-Clients verbinden sich immer mit dem NIS-Server, der zuerst reagiert. Dieser Server kann auch ein Slaveserver sein.

- *NIS-Clients*. NIS-Clients identifizieren sich gegenüber dem NIS-Server (ähnlich den Windows NT-Workstations), um sich am Server anzumelden.

30.4.4. NIS/YP konfigurieren

Dieser Abschnitt beschreibt an Hand eines Beispiels die Einrichtung einer NIS-Umgebung.

30.4.4.1. Planung

Nehmen wir an, Sie seien der Administrator eines kleinen Universitätsnetzes. Dieses Netz besteht aus fünfzehn FreeBSD-Rechnern, für die derzeit keine zentrale Verwaltung existiert, jeder Rechner hat also eine eigene Version von `/etc/passwd` und `/etc/master.passwd`. Diese Dateien werden manuell synchron gehalten; legen Sie einen neuen Benutzer an, so muss dies auf allen fünfzehn Rechnern manuell erledigt werden (unter Verwendung von `adduser`). Da diese Lösung sehr ineffizient ist, soll das Netzwerk in Zukunft NIS verwenden, wobei zwei der Rechner als Server dienen sollen.

In Zukunft soll das Netz also wie folgt aussehen:

Rechnername	IP-Adresse	Rechneraufgabe
`ellington`	`10.0.0.2`	NIS-Master
`coltrane`	`10.0.0.3`	NIS-Slave
`basie`	`10.0.0.4`	Workstation der Fakultät
`bird`	`10.0.0.5`	Clientrechner
`cli[1-11]`	`10.0.0.[6-17]`	Verschiedene andere Clients

Wenn Sie NIS das erste Mal einrichten, ist es ratsam, sich zuerst über die Vorgangsweise Gedanken zu machen. Unabhängig von der Größe Ihres Netzwerks müssen Sie stets einige Entscheidungen treffen.

30.4.4.1.1. Einen NIS-Domänennamen wählen

Dies muss nicht der “Domainname” sein. Es handelt sich vielmehr um den “NIS-Domainnamen”. Wenn ein Client Informationen anfordert, ist in dieser Anforderung der Name der NIS-Domäne enthalten. Dadurch weiß jeder Server im Netzwerk, auf welche Anforderung er antworten muss. Stellen Sie sich den NIS-Domänennamen als den Namen einer Gruppe von Rechnern vor, die etwas gemeinsam haben.

Manchmal wird der Name der Internetdomäne auch für die NIS-Domäne verwendet. Dies ist allerdings nicht empfehlenswert, da dies bei der Behebung von Problemen verwirrend sein kann. Der Name der NIS-Domäne sollte innerhalb Ihres Netzwerks einzigartig sein. Hilfreich ist es, wenn der Name die Gruppe der in ihr zusammengefassten Rechner beschreibt. Die Kunstabteilung von Acme Inc. hätte daher die NIS-Domäne “acme-art”. Für unser Beispiel verwenden wir den NIS-Domänennamen `test-domain`.

Es gibt jedoch auch Betriebssysteme (vor allem SunOS), die als NIS-Domänennamen den Name der Internetdomäne verwenden. Wenn dies für einen oder mehrere Rechner Ihres Netzwerks zutrifft, *müssen* Sie den Namen der Internetdomäne als Ihren NIS-Domänennamen verwenden.

30.4.4.1.2. Anforderungen an den Server

Wenn Sie einen NIS-Server einrichten wollen, müssen Sie einige Dinge beachten. Eine unangenehme Eigenschaft von NIS ist die Abhängigkeit der Clients vom Server. Wenn sich der Client nicht über den Server mit seiner NIS-Domäne verbinden kann, wird der Rechner oft unbenutzbar, da das Fehlen von Benutzer- und Gruppeninformationen zum Einfrieren des Clients führt. Daher sollten Sie für den Server einen Rechner auswählen, der nicht regelmäßig neu gestartet werden muss und der nicht für Testversuche verwendet wird. Idealerweise handelt es sich um einen alleinstehenden Rechner, dessen einzige Aufgabe es ist, als NIS-Server zu dienen. Wenn Sie ein Netzwerk haben, das nicht zu stark ausgelastet ist, ist es auch möglich, den NIS-Server als weiteren Dienst auf einem anderen Rechner laufen zu lassen. Denken Sie aber daran, dass ein Ausfall des NIS-Servers *alle* NIS-Clients betrifft.

30.4.4.2. NIS-Server

Die verbindlichen Kopien aller NIS-Informationen befinden sich auf einem einzigen Rechner, dem NIS-Masterserver. Die Datenbanken, in denen die Informationen gespeichert sind, bezeichnet man als NIS-Maps. Unter FreeBSD werden diese Maps unter `/var/yp/[domainname]` gespeichert, wobei `[domainname]` der Name der NIS-Domäne ist. Ein einzelner NIS-Server kann gleichzeitig mehrere NIS-Domänen verwalten, daher können auch mehrere Verzeichnisse vorhanden sein. Jede Domäne verfügt über ein eigenes Verzeichnis sowie einen eigenen, von anderen Domänen unabhängigen Satz von NIS-Maps.

NIS-Master- und Slaveserver verwenden den `ypserv`-Daemon, um NIS-Anfragen zu bearbeiten. `ypserv` empfängt eingehende Anfragen der NIS-Clients, ermittelt aus der angeforderten Domäne und Map einen Pfad zur entsprechenden Datenbank, und sendet die angeforderten Daten von der Datenbank zum Client.

30.4.4.2.1. Einen NIS-Masterserver einrichten

Abhängig von Ihren Anforderungen ist die Einrichtung eines NIS-Masterservers relativ einfach, da NIS von FreeBSD bereits in der Standardkonfiguration unterstützt wird. Sie müssen nur folgende Zeilen in `/etc/rc.conf` einfügen:

1. nisdomainname="test-domain"

   Diese Zeile setzt den NIS-Domänennamen auf `test-domain`, wenn Sie das Netzwerk initialisieren (beispielsweise nach einem Systemstart).

2. `nis_server_enable="YES"`
   Dadurch werden die NIS-Serverprozesse gestartet.

3. `nis_yppasswdd_enable="YES"`
   Durch diese Zeile wird der `rpc.yppasswdd`-Daemon aktiviert, der, wie bereits erwähnt, die Änderung von NIS-Passwörtern von einem Client aus ermöglicht.

> **Anmerkung:** In Abhängigkeit von Ihrer NIS-Konfiguration können weitere Einträge erforderlich sein. Weitere Informationen finden Sie im Abschnitt NIS-Server, die auch als NIS-Clients arbeiten.

Nachdem Sie obige Parameter konfiguriert haben, müssen Sie nur noch `/etc/netstart` als Superuser ausführen, um alles entsprechend Ihren Vorgaben in der Datei `/etc/rc.conf` einzurichten. Bevor Sie die NIS-Maps einrichten können, müssen Sie nun noch den **ypserv**-Daemon manuell starten:

```
# /etc/rc.d/ypserv start
```

30.4.4.2.2. Die NIS-Maps initialisieren

NIS-Maps sind Datenbanken, die sich im Verzeichnis `/var/yp` befinden. Sie werden am NIS-Masterserver aus den Konfigurationsdateien unter `/etc` erzeugt. Einzige Ausnahme: `/etc/master.passwd`. Dies ist auch sinnvoll, da Sie die Passwörter für Ihr `root`- oder andere Administratorkonten nicht an alle Server der NIS-Domäne verteilen wollen. Bevor Sie also die NIS-Maps des Masterservers einrichten, sollten Sie Folgendes tun:

```
# cp /etc/master.passwd /var/yp/master.passwd
```

```
# cd /var/yp
# vi master.passwd
```

Entfernen Sie alle Systemkonten (wie `bin`, `tty`, `kmem` oder `games`), sowie alle Konten, die Sie nicht an die NIS-Clients weitergeben wollen (beispielsweise `root` und alle Konten mit der UID 0 (=Superuser).

> **Anmerkung:** Stellen Sie sicher, dass `/var/yp/master.passwd` weder von der Gruppe noch von der Welt gelesen werden kann (Zugriffsmodus 600)! Ist dies nicht der Fall, ändern Sie dies mit `chmod`.

Nun können Sie die NIS-Maps initialisieren. FreeBSD verwendet dafür das Skript `ypinit` (lesen Sie dazu auch ypinit(8)). Dieses Skript ist auf fast allen UNIX-Betriebssystemen verfügbar. Bei Digitals Unix/Compaq Tru64 UNIX nennt es sich allerdings `ypsetup`. Da wir Maps für einen NIS-Masterserver erzeugen, verwenden wir `ypinit` mit der Option `-m`. Nachdem Sie die beschriebenen Aktionen durchgeführt haben, erzeugen Sie nun die NIS-Maps:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..output from map generation..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

Dadurch erzeugt `ypinit` `/var/yp/Makefile` aus der Datei `/var/yp/Makefile.dist`. Durch diese Datei wird festgelegt, dass Sie in einer NIS-Umgebung mit nur einem Server arbeiten und dass alle Clients unter FreeBSD laufen. Da `test-domain` aber auch über einen Slaveserver verfügt, müssen Sie `/var/yp/Makefile` entsprechend anpassen:

```
ellington# vi /var/yp/Makefile
```

Sie sollten die Zeile

```
NOPUSH = "True"
```

auskommentieren (falls dies nicht bereits der Fall ist).

30.4.4.2.3. Einen NIS-Slaveserver einrichten

Ein NIS-Slaveserver ist noch einfacher einzurichten als ein Masterserver. Melden Sie sich am Slaveserver an und ändern Sie `/etc/rc.conf` analog zum Masterserver. Der einzige Unterschied besteht in der Verwendung der Option `-s`, wenn Sie `ypinit` aufrufen. Die Option `-s` erfordert den Namen des NIS-Masterservers, daher sieht unsere Ein- und Ausgabe wie folgt aus:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
```

```
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
Don't forget to update map ypservers on ellington.
```

Sie sollten nun über das Verzeichnis `/var/yp/test-domain` verfügen. Die Kopien der NIS-Masterserver-Maps sollten sich in diesem Verzeichnis befinden. Allerdings müssen Sie diese auch aktuell halten. Die folgenden Einträge in `/etc/crontab` erledigen diese Aufgabe:

```
20      *       *       *       *       root   /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root   /usr/libexec/ypxfr passwd.byuid
```

Diese zwei Zeilen zwingen den Slaveserver, seine Maps mit denen des Masterservers zu synchronisieren. Diese Einträge sind nicht zwar nicht unbedingt nötig, da der Masterserver automatisch versucht, alle Änderungen seiner NIS-Maps an seine Slaveserver weiterzugeben. Da Passwortinformationen aber auch für nur vom Slaveserver abhängige Systeme vital sind, ist es eine gute Idee, diese Aktualisierungen zu erzwingen. Besonders wichtig ist dies in stark ausgelasteten Netzen, in denen Map-Aktualisierungen unvollständig sein könnten.

Führen Sie nun `/etc/netstart` auch auf dem Slaveserver aus, um den NIS-Server erneut zu starten.

30.4.4.3. NIS-Clients

Ein NIS-Client `bindet` sich unter Verwendung des `ypbind`-Daemons an einen NIS-Server. `ypbind` prüft die Standarddomäne des Systems (die durch `domainname` gesetzt wird), und beginnt RPCs über das lokale Netzwerk zu verteilen (broadcast). Diese Anforderungen legen den Namen der Domäne fest, für die `ypbind` eine Bindung erzeugen will. Wenn der Server der entsprechenden Domäne eine solche Anforderung erhält, schickt er eine Antwort an `ypbind`. `ybind` speichert daraufhin die Adresse des Servers. Wenn mehrere Server verfügbar sind (beispielsweise ein Master- und mehrere Slaveserver), verwendet `ypbind` die erste erhaltene Adresse. Ab diesem Zeitpunkt richtet der Client alle Anfragen an genau diesen Server. `ypbind` "pingt" den Server gelegentlich an, um sicherzustellen, dass der Server funktioniert. Antwortet der Server innerhalb eines bestimmten Zeitraums nicht (Timeout), markiert `ypbind` die Domäne als ungebunden und beginnt erneut, RPCs über das Netzwerk zu verteilen, um einen anderen Server zu finden.

30.4.4.3.1. Einen NIS-Client konfigurieren

Einen FreeBSD-Rechner als NIS-Client einzurichten, ist recht einfach.

1. Fügen Sie folgende Zeilen in `/etc/rc.conf` ein, um den NIS-Domänennamen festzulegen, und um `ypbind` bei der Initialisierung des Netzwerks zu starten:

   ```
   nisdomainname="test-domain"
   nis_client_enable="YES"
   ```

2. Um alle Passworteinträge des NIS-Servers zu importieren, löschen Sie alle Benutzerkonten in `/etc/master.passwd` und fügen mit `vipw` folgende Zeile am Ende der Datei ein:

```
+::::::::::
```

> **Anmerkung:** Diese Zeile legt für alle gültigen Benutzerkonten der NIS-Server-Maps einen Zugang an. Es gibt verschiedene Wege, Ihren NIS-Client durch Änderung dieser Zeile zu konfigurieren. Lesen Sie dazu auch den Abschnitt über Netzgruppen weiter unten. Weitere detaillierte Informationen finden Sie im Buch `Managing NFS and NIS` von O'Reilly.

> **Anmerkung:** Sie sollten zumindest ein lokales Benutzerkonto, das nicht über NIS importiert wird, in Ihrer `/etc/master.passwd` behalten. Dieser Benutzer sollte außerdem ein Mitglied der Gruppe `wheel` sein. Wenn es mit NIS Probleme gibt, können Sie diesen Zugang verwenden, um sich anzumelden, `root` zu werden und das Problem zu beheben.

3. Um alle möglichen Gruppeneinträge vom NIS-Server zu importieren, fügen sie folgende Zeile in `/etc/group` ein:

   ```
   +:*::
   ```

Um den NIS-Client sofort zu starten, führen Sie als Superuser die folgenden Befehle aus:

```
# /etc/netstart
# /etc/rc.d/ypbind start
```

Nachdem Sie diese Schritte erledigt haben, sollten Sie mit `ypcat passwd` die `passwd-Map` des NIS-Servers anzeigen können.

30.4.5. Sicherheit unter NIS

Im Allgemeinen kann jeder entfernte Anwender einen RPC an ypserv(8) schicken, um den Inhalt Ihrer NIS-Maps abzurufen, falls er Ihren NIS-Domänennamen kennt. Um solche unautorisierten Transaktionen zu verhindern, unterstützt ypserv(8) "securenets", durch die man den Zugriff auf bestimmte Rechner beschränken kann. ypserv(8) versucht, beim Systemstart die Informationen über `securenets` aus der Datei `/var/yp/securenets` zu laden.

> **Anmerkung:** Die Datei `securenets` kann auch in einem anderen Verzeichnis stehen, das mit der Option `-p` angegeben wird. Diese Datei enthält Einträge, die aus einer Netzwerkadresse und einer Netzmaske bestehen, die durch Leerzeichen getrennt werden. Kommentarzeilen beginnen mit "#". `/var/yp/securnets` könnte beispielsweise so aussehen:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
```

```
10.0.0.0        255.255.240.0
```

Wenn ypserv(8) eine Anforderung von einer zu diesen Regeln passenden Adresse erhält, wird die Anforderung bearbeitet. Gibt es keine passende Regel, wird die Anforderung ignoriert und eine Warnmeldung aufgezeichnet. Wenn `/var/yp/securenets` nicht vorhanden ist, erlaubt `ypserv` Verbindungen von jedem Rechner aus.

`ypserv` unterstützt auch das **TCP-Wrapper**-Paket von Wietse Venema. Mit diesem Paket kann der Administrator für Zugriffskontrollen die Konfigurationsdateien von **TCP-Wrapper** anstelle von `/var/yp/securenets` verwenden.

> **Anmerkung:** Während beide Kontrollmechanismen einige Sicherheit gewähren, beispielsweise durch privilegierte Ports, sind sie gegenüber “IP spoofing”-Attacken verwundbar. Jeder NIS-Verkehr sollte daher von Ihrer Firewall blockiert werden.
>
> Server, die `/var/yp/securenets` verwenden, können Schwierigkeiten bei der Anmeldung von Clients haben, die ein veraltetes TCP/IP-Subsystem besitzen. Einige dieser TCP/IP-Subsysteme setzen alle Rechnerbits auf Null, wenn Sie einen `Broadcast` durchführen und/oder können die Subnetzmaske nicht auslesen, wenn sie die Broadcast-Adresse berechnen. Einige Probleme können durch Änderungen der Clientkonfiguration behoben werden. Andere hingegen lassen sich nur durch das Entfernen des betreffenden Rechners aus dem Netzwerk oder den Verzicht auf `/var/yp/securenets` umgehen.
>
> Die Verwendung von `/var/yp/securenets` auf einem Server mit einem solch veralteten TCP/IP-Subsystem ist eine sehr schlechte Idee, die zu einem Verlust der NIS-Funktionalität für große Teile Ihres Netzwerks führen kann.
>
> Die Verwendung der **TCP-Wrapper** verlangsamt die Reaktion Ihres NIS-Servers. Diese zusätzliche Reaktionszeit kann in Clientprogrammen zu Timeouts führen. Dies vor allem in Netzwerken, die stark ausgelastet sind, oder nur über langsame NIS-Server verfügen. Wenn ein oder mehrere Ihrer Clientsysteme dieses Problem aufweisen, sollten Sie die betreffenden Clients in NIS-Slaveserver umwandeln, und diese an sich selbst binden.

30.4.6. Bestimmte Benutzer an der Anmeldung hindern

In unserem Labor gibt es den Rechner `basie`, der nur für Mitarbeiter der Fakultät bestimmt ist. Wir wollen diesen Rechner nicht aus der NIS-Domäne entfernen, obwohl `passwd` des NIS-Masterservers Benutzerkonten sowohl für Fakultätsmitarbeiter als auch für Studenten enthält. Was können wir also tun?

Es gibt eine Möglichkeit, bestimmte Benutzer an der Anmeldung an einem bestimmten Rechner zu hindern, selbst wenn diese in der NIS-Datenbank vorhanden sind. Dazu müssen Sie lediglich an diesem Rechner den Eintrag *`-Benutzername`* an das Ende von `/etc/master.passwd` setzen, wobei *`Benutzername`* der zu blockierende Benutzername ist. Diese Änderung sollte bevorzugt durch `vipw` erledigt werden, da `vipw` Ihre Änderungen an `/etc/master.passwd` auf Plausibilität überprüft und nach erfolgter Änderung die Passwortdatenbank automatisch aktualisiert. Um also den Benutzer `bill` an der Anmeldung am Rechner `basie` zu hindern, gehen wir wie folgt vor:

```
basie# vipw
[add -bill to the end, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0::0:0:The super-user:/root:/bin/csh
```

```
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5::0:0:System &:/:/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/share/man:/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/sbin/nologin
+::::::::::
-bill

basie#
```

30.4.7. Netzgruppen verwenden

Beigetragen von Udo Erdelhoff.

Die im letzten Abschnitt beschriebene Methode eignet sich besonders, wenn Sie spezielle Regeln für wenige Benutzer oder wenige Rechner benötigen. In großen Netzwerken werden Sie allerdings *mit Sicherheit* vergessen, einige Benutzer von der Anmeldung an bestimmten Rechnern auszuschließen. Oder Sie werden gezwungen sein, jeden Rechner einzeln zu konfigurieren. Dadurch verlieren Sie aber den Hauptvorteil von NIS, die *zentrale* Verwaltung.

Die Lösung für dieses Problem sind *Netzgruppen*. Ihre Aufgabe und Bedeutung ist vergleichbar mit normalen, von UNIX-Dateisystemen verwendeten Gruppen. Die Hauptunterschiede sind das Fehlen einer numerischen ID sowie die Möglichkeit, Netzgruppen zu definieren, die sowohl Benutzer als auch andere Netzgruppen enthalten.

Netzgruppen wurden entwickelt, um große, komplexe Netzwerke mit Hunderten Benutzern und Rechnern zu verwalten. Sie sind also von Vorteil, wenn Sie von dieser Situation betroffen sind. Andererseits ist es dadurch beinahe unmöglich, Netzgruppen mit einfachen Beispielen zu erklären. Das hier verwendete Beispiel veranschaulicht dieses Problem.

Nehmen wir an, dass Ihre erfolgreiche Einführung von NIS die Aufmerksamkeit Ihrer Vorgesetzten geweckt hat. Ihre nächste Aufgabe besteht nun darin, Ihre NIS-Domäne um zusätzliche Rechner zu erweitern. Die folgenden Tabellen enthalten die neuen Benutzer und Rechner inklusive einer kurzen Beschreibung.

Benutzername(n)	Beschreibung
`alpha`, `beta`	Beschäftigte der IT-Abteilung
`charlie`, `delta`	Die neuen Lehrlinge der IT-Abteilung
`echo`, `foxtrott`, `golf`, ...	Normale Mitarbeiter
`able`, `baker`, ...	Externe Mitarbeiter

Rechnername(n)	Beschreibung

Rechnername(n)	Beschreibung
war, death, famine, pollution	Ihre wichtigsten Server. Nur IT-Fachleute dürfen sich an diesen Rechnern anmelden.
pride, greed, envy, wrath, lust, sloth	Weniger wichtige Server. Alle Mitarbeiter der IT-Abteilung dürfen sich auf diesen Rechnern anmelden.
one, two, three, four, ...	Gewöhnliche Arbeitsrechner. Nur die *wirklichen* Mitarbeiter dürfen diese Rechner verwenden.
trashcan	Ein sehr alter Rechner ohne kritische Daten. Sogar externe Mitarbeiter dürfen diesen Rechner verwenden.

Wollten Sie diese Einschränkungen umsetzen, indem Sie jeden Benutzer einzeln blockieren, müssten Sie auf jedem System für jeden Benutzer eine entsprechende Zeile in `passwd` einfügen. Wenn Sie nur einen Eintrag vergessen, haben Sie ein Problem. Es mag noch angehen, dies während der ersten Installation zu erledigen, im täglichen Betrieb werden Sie allerdings *mit Sicherheit* einmal vergessen, die entsprechenden Einträge anzulegen. Vergessen Sie nicht: Murphy war Optimist.

Die Verwendung von Netzgruppen hat in dieser Situation mehrere Vorteile. Sie müssen nicht jeden Benutzer einzeln verwalten; weisen Sie stattdessen den Benutzer einer Netzgruppe zu und erlauben oder verbieten Sie allen Mitglieder dieser Gruppe die Anmeldung an einem Server. Wenn Sie einen neuen Rechner hinzufügen, müssen Sie Zugangsbeschränkungen nur für die Netzgruppen festlegen. Legen Sie einen neuen Benutzer an, müssen Sie ihn nur einer oder mehrere Netzgruppen zuweisen. Diese Veränderungen sind voneinander unabhängig; Anweisungen der Form “für diese Kombination aus Benutzer und Rechner mache Folgendes ...” sind nicht mehr nötig. Wenn Sie die Einrichtung von NIS sorgfältig geplant haben, müssen Sie nur noch eine zentrale Konfigurationsdatei bearbeiten, um den Zugriff auf bestimmte Rechner zu erlauben oder zu verbieten.

Der erste Schritt ist die Initialisierung der NIS-Maps der Netzgruppe. ypinit(8) kann dies unter FreeBSD nicht automatisch durchführen. Sind die Maps aber erst einmal erzeugt, werden sie jedoch von NIS problemlos unterstützt. Um eine leere Map zu erzeugen, geben Sie Folgendes ein:

```
ellington# vi /var/yp/netgroup
```

Danach legen Sie die Einträge an. Für unser Beispiel benötigen wir mindestens vier Netzgruppen: IT-Beschäftige, IT-Lehrlinge, normale Beschäftigte sowie Externe.

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
        (,golf,test-domain)
INTERNS (,able,test-domain)     (,baker,test-domain)
```

Bei `IT_EMP`, `IT_APP` usw. handelt es sich um Netzgruppennamen. In den Klammern werden diesen Netzgruppen jeweils ein oder mehrere Benutzerkonten hinzugefügt. Die drei Felder in der Klammer haben folgende Bedeutung:

1. Der Name des Rechners, auf dem die folgenden Werte gültig sind. Legen Sie keinen Rechnernamen fest, ist der Eintrag auf allen Rechnern gültig. Dadurch gehen Sie vielen Problemen aus dem Weg.
2. Der Name des Benutzerkontos, der zu dieser Netzgruppe gehört.
3. Die NIS-Domäne für das Benutzerkonto. Sie können Benutzerkonten von anderen NIS-Domänen in Ihre Netzgruppe importieren, wenn Sie mehrere NIS-Domänen verwalten.

Jedes Feld kann Wildcards enthalten. Die Einzelheiten entnehmen Sie bitte netgroup(5).

> **Anmerkung:** Netzgruppennamen sollten nicht länger als 8 Zeichen sein, vor allem dann, wenn Sie Rechner mit verschiedenen Betriebssystemen in Ihrer NIS-Domäne haben. Es wird zwischen Groß- und Kleinschreibung unterschieden. Die Verwendung von Großbuchstaben für Netzgruppennamen ermöglicht eine leichte Unterscheidung zwischen Benutzern, Rechnern und Netzgruppen.
>
> Einige NIS-Clients (dies gilt nicht für FreeBSD) können keine Netzgruppen mit einer großen Anzahl von Einträgen verwalten. Einige ältere Versionen von SunOS haben beispielsweise Probleme, wenn Netzgruppen mehr als fünfzehn *Einträge* enthalten. Sie können dieses Problem umgehen, indem Sie mehrere Subnetzgruppen mit weniger als fünfzehn Benutzern anlegen und diese Subnetzgruppen wiederum in einer Netzgruppe zusammenfassen:
>
> ```
> BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
> BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
> BIGGRP3  (,joe31,domain)  (,joe32,domain)
> BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
> ```
>
> Sie können diesen Vorgang wiederholen, wenn Sie mehr als 255 Benutzer in einer einzigen Netzgruppe benötigen.

Das Aktivieren und Verteilen Ihre neuen NIS-Map ist einfach:

```
ellington# cd /var/yp
ellington# make
```

Dadurch werden die NIS-Maps `netgroup`, `netgroup.byhost` und `netgroup.byuser` erzeugt. Prüfen Sie die Verfügbarkeit Ihrer neuen NIS-Maps mit ypcat(1).

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

Die Ausgabe des ersten Befehls gibt den Inhalt von `/var/yp/netgroup` wieder. Der zweite Befehl erzeugt nur dann eine Ausgabe, wenn Sie rechnerspezifische Netzgruppen erzeugt haben. Der dritte Befehl gibt die Netzgruppen nach Benutzern sortiert aus.

Die Einrichtung der Clients ist einfach. Sie müssen lediglich auf dem Server `war` vipw(8) aufrufen und die Zeile

```
+::::::::::
```

durch

```
+@IT_EMP:::::::::
```

ersetzen.

Ab sofort werden nur noch die Daten der in der Netzgruppe `IT_EMP` vorhandenen Benutzer in die Passwortdatenbank von `war` importiert. Nur diese Benutzer dürfen sich am Server anmelden.

Unglücklicherweise gilt diese Einschränkung auch für die ~-Funktion der Shell und für alle Routinen, die auf Benutzernamen und numerische Benutzer-IDs zugreifen. Oder anders formuliert, `cd ~`*`user`* ist nicht möglich, `ls -l` zeigt die numerische Benutzer-ID statt dem Benutzernamen und `find . -user joe -print` erzeugt die

Fehlermeldung `No such user`. Um dieses Problem zu beheben, müssen Sie alle Benutzereinträge importieren, *ohne ihnen jedoch zu erlauben, sich an Ihrem Server anzumelden*.

Dazu fügen Sie eine weitere Zeile in `/etc/master.passwd` ein. Diese Zeile sollte ähnlich der folgenden aussehen:

`+::::::::::/sbin/nologin`, was in etwa "Importiere alle Einträge, aber ersetze die Shell in den importierten Einträgen durch `/sbin/nologin`" entspricht. Sie können jedes Feld dieses Eintrages ersetzen, indem Sie einen Standardwert in `/etc/master.passwd` eintragen.

> **Warnung:** Stellen Sie sicher, dass die Zeile `+::::::::::/sbin/nologin` *nach* der Zeile `+@IT_EMP:::::::::` eingetragen ist. Sonst haben alle via NIS importierten Benutzerkonten `/sbin/nologin` als Loginshell.

Danach müssen Sie nur mehr eine einzige NIS-Map ändern, wenn ein neuer Mitarbeiter berücksichtigt werden muss. Für weniger wichtige Server gehen Sie analog vor, indem Sie den alten Eintrag `+:::::::::` in den lokalen Versionen von `/etc/master.passwd` durch folgende Einträge ersetzen:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+::::::::::/sbin/nologin
```

Die entsprechenden Zeilen für normale Arbeitsplätze lauten:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+::::::::::/sbin/nologin
```

Ab jetzt wäre alles wunderbar, allerdings ändert sich kurz darauf die Firmenpolitik: Die IT-Abteilung beginnt damit, externe Mitarbeiter zu beschäftigen. Externe dürfen sich an normalen Arbeitsplätzen sowie an den weniger wichtigen Servern anmelden. Die IT-Lehrlinge dürfen sich nun auch an den Hauptservern anmelden. Sie legen also die neue Netzgruppe `IT_INTERN` an, weisen Ihr die neuen IT-Externen als Benutzer zu und beginnen damit, die Konfiguration auf jedem einzelnen Rechner zu ändern ... Halt. Sie haben gerade die alte Regel "Fehler in der zentralisierten Planung führen zu globaler Verwirrung." bestätigt.

Da NIS in der Lage ist, Netzgruppen aus anderen Netzgruppen zu bilden, lassen sich solche Situationen leicht vermeiden. Eine Möglichkeit ist die Erzeugung rollenbasierter Netzgruppen. Sie könnten eine Netzgruppe `BIGSRV` erzeugen, um den Zugang zu den wichtigsten Servern zu beschränken, eine weitere Gruppe `SMALLSRV` für die weniger wichtigen Server und eine dritte Netzgruppe `USERBOX` für die normalen Arbeitsplatzrechner. Jede dieser Netzgruppen enthält die Netzgruppen, die sich auf diesen Rechnern anmelden dürfen. Die Einträge der Netzgruppen in der NIS-Map sollten ähnlich den folgenden aussehen:

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS
```

Diese Methode funktioniert besonders gut, wenn Sie Rechner in Gruppen mit identischen Beschränkungen einteilen können. Unglücklicherweise ist dies die Ausnahme und nicht die Regel. Meistens werden Sie die Möglichkeit zur rechnerspezischen Zugangsbeschränkung benötigen.

Rechnerspezifische Netzgruppen sind die zweite Möglichkeit, um mit den oben beschriebenen Änderungen umzugehen. In diesem Szenario enthält `/etc/master.passwd` auf jedem Rechner zwei mit "+" beginnende Zeilen. Die erste Zeile legt die Netzgruppe mit den Benutzern fest, die sich auf diesem Rechner anmelden dürfen. Die zweite

Zeile weist allen anderen Benutzern `/sbin/nologin` als Shell zu. Verwenden Sie auch hier (analog zu den Netzgruppen) Großbuchstaben für die Rechnernamen. Die Zeilen sollten also ähnlich den folgenden aussehen:

```
+@BOXNAME:::::::::
+::::::::::/sbin/nologin
```

Wenn Sie dies für alle Rechner erledigt haben, werden Sie die lokalen Versionen von `/etc/master.passwd` nie mehr verändern müssen. Alle weiteren Änderungen geschehen über die NIS-Maps. Nachfolgend ein Beispiel für eine mögliche Netzgruppen-Map, die durch einige Besonderheiten erweitert wurde:

```
# Define groups of users first
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Now, define some groups based on roles
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# And a groups for a special tasks
# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)
#
# This one is really important and needs more access restrictions
DEATH     IT_EMP
#
# The anti-virus-machine mentioned above
ONE       SECURITY
#
# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]
```

Wenn Sie eine Datenbank verwenden, um Ihre Benutzerkonten zu verwalten, sollten Sie den ersten Teil der NIS-Map mit Ihren Datenbanktools erstellen können. Auf diese Weise haben neue Benutzer automatisch Zugriff auf die Rechner.

Eine letzte Warnung: Es ist nicht immer ratsam, rechnerbasierte Netzgruppen zu verwenden. Wenn Sie Dutzende oder gar Hunderte identische Rechner einrichten müssen, sollten Sie rollenbasierte Netzgruppen verwenden, um die Grösse der NISs-Maps in Grenzen zu halten.

30.4.8. Weitere wichtige Punkte

Nachdem Sie Ihre NIS-Umgebung eingerichtet haben, müssen Sie einige Dinge anders als bisher erledigen.

- Jedes Mal, wenn Sie einen neuen Benutzer anlegen wollen, tun Sie dies *ausschließlich* am NIS-Masterserver. Außerdem *müssen* Sie anschließend die NIS-Maps neu erzeugen. Wenn Sie diesen Punkt vergessen, kann sich der neue Benutzer *nur* am NIS-Masterserver anmelden. Wenn Sie also den neuen Benutzer `jsmith` anlegen, gehen Sie folgerndermassen vor:

  ```
  # pw useradd jsmith
  # cd /var/yp
  # make test-domain
  ```

  Statt `pw useradd jsmith` könnten Sie auch `adduser jsmith` verwenden.

- *Tragen Sie die Administratorkonten nicht in die NIS-Maps ein*. Administratorkonten und Passwörter dürfen nicht auf Rechnern verbreitet werden, auf denen sich Benutzer anmelden können, die auf diese Konten keine Zugriff haben sollen.

- *Sichern Sie die NIS-Master- und Slaveserver und minimieren Sie die Ausfallzeiten*. Wenn diese Rechner gehackt oder einfach nur ausgeschaltet werden, haben viele Leute keinen Netzwerkzugriff mehr.

  Dies ist die größte Schwäche jeder zentralen Verwaltung. Wenn Sie Ihre NIS-Server nicht schützen, werden Sie viele verärgerte Anwender haben.

30.4.9. Kompatibilität zu NIS v1

ypserv unterstützt NIS v1 unter FreeBSD nur eingeschränkt. Die NIS-Implementierung von FreeBSD verwendet nur NIS v2, andere Implementierungen unterstützen aus Gründen der Abwärtskompatibilität mit älteren Systemen auch NIS v1. Die mit diesen Systemen gelieferten **ypbind**-Daemonen versuchen, sich an einen NIS-v1-Server zu binden (Dies selbst dann, wenn sie ihn nie benötigen. Außerdem versuchen Sie auch dann, einen v1-Server zu erreichen, wenn Sie zuvor eine Antwort von einem v2-Server erhalten.). Während normale Clientaufrufe unter FreeBSD unterstützt werden, sind Anforderungen zum Transfer von v1-Maps nicht möglich. Daher kann FreeBSD nicht als Client oder Server verwendet werden, wenn ein NIS-Server vorhanden ist, der nur NIS v1 unterstützt. Glücklicherweise sollte es heute keine Server mehr geben, die nur NIS v1 unterstützen.

30.4.10. NIS-Server, die auch als NIS-Clients arbeiten

Wenn Sie **ypserv** in einer Multi-Serverdomäne verwenden, in der NIS-Server gleichzeitig als NIS-Clients arbeiten, ist es eine gute Idee, diese Server zu zwingen, sich an sich selbst zu binden. Damit wird verhindert, dass Bindeanforderungen gesendet werden und sich die Server gegenseitig binden. Sonst könnten seltsame Fehler auftreten, wenn ein Server ausfällt, auf den andere Server angewiesen sind. Letztlich werden alle Clients einen Timeout melden, und versuchen, sich an andere Server zu binden. Die dadurch entstehende Verzögerung kann beträchtlich sein. Außerdem kann der Fehler erneut auftreten, da sich die Server wiederum aneinander binden könnten.

Sie können einen Rechner durch die Verwendung von `ypbind` sowie der Option `-S` zwingen, sich an einen bestimmten Server zu binden. Um diesen Vorgang zu automatisieren, können Sie folgende Zeilen in `/etc/rc.conf` einfügen:

```
nis_client_enable="YES" # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

Lesen Sie ypbind(8), wenn Sie weitere Informationen benötigen.

30.4.11. Passwortformate

Unterschiedliche Passwortformate sind das Hauptproblem, das beim Einrichten eines NIS-Servers auftreten kann. Wenn der NIS-Server mit DES verschlüsselte Passwörter verwendet, werden nur Clients unterstützt, die ebenfalls DES benutzen. Wenn sich auf Ihrem Netzwerk beispielsweise Solaris NIS-Clients befinden, müssen die Passwörter mit DES verschlüsselt werden.

Welches Format die Server und Clients verwenden, steht in `/etc/login.conf`. Wenn ein System Passwörter mit DES verschlüsselt, enthält die `default`-Klasse einen Eintrag wie den folgenden:

```
default:\
       :passwd_format=des:\
       :copyright=/etc/COPYRIGHT:\
       [weitere Einträge]
```

Mögliche Werte für `passwd_format` sind unter anderem `blf` und `md5` (mit Blowfish und MD5 verschlüsselte Passwörter).

Wenn die Datei `/etc/login.conf` geändert wird, muss die Login-Capability Datenbank neu erstellt werden. Geben Sie dazu als `root` den folgenden Befehl ein:

```
# cap_mkdb /etc/login.conf
```

> **Anmerkung:** Das Format der schon in `/etc/master.passwd` befindlichen Passwörter wird erst aktualisiert, wenn ein Benutzer sein Passwort ändert, *nachdem* die Datenbank neu erstellt wurde.

Damit die Passwörter auch im gewählten Format abgespeichert werden, muss mit `crypt_default` in der Datei `/etc/auth.conf` die richtige Priorität der Formate eingestellt werden. Das gewählte Format sollte als Erstes in der Liste stehen. Sollen die Passwörter mit DES verschlüsselt werden, verwenden Sie den folgenden Eintrag:

```
crypt_default   =       des blf md5
```

Wenn Sie alle FreeBSD NIS-Server und NIS-Clients entsprechend den obigen Schritten eingestellt haben, wird im ganzen Netzwerk dasselbe Passwortformat verwendet. Falls Sie Probleme mit der Authentifizierung eines NIS-Clients haben, kontrollieren Sie die verwendeten Passwortformate. In einer heterogenen Umgebung werden Sie DES benutzen müssen, da dies der meist unterstützte Standard ist.

30.5. Automatische Netzwerkkonfiguration mit DHCP

Geschrieben von Greg Sutter.

30.5.1. Was ist DHCP?

Über DHCP, das Dynamic Host Configuration Protocol, kann sich ein System mit einem Netzwerk verbinden und die für die Kommunikation mit diesem Netzwerk nötigen Informationen beziehen. FreeBSD verwendet den von OpenBSD 3.7 stammenden `dhclient`. Die Informationen in diesem Abschnitt beziehen sich daher sowohl auf den `dhclient` von ISC als auch auf den von OpenBSD. Als DHCP-Server wird in beiden Fällen der DHCP-Server der ISC-Distribution verwendet.

30.5.2. Übersicht

Dieser Abschnitt beschreibt sowohl die Clientseite des ISC- als auch des OpenBSD-Clients sowie die Serverseite des DHCP-Systems von ISC. Das Clientprogramm `dhclient` ist in FreeBSD integriert, das Serverprogramm kann über den Port `net/isc-dhcp42-server` installiert werden. Weiter Informationen finden Sie in dhclient(8), dhcp-options(5) sowie dhclient.conf(5).

30.5.3. Wie funktioniert DHCP?

Der DHCP-Client `dhclient` beginnt von einem Clientrechner aus über den UDP-Port 68 Konfigurationsinformationen anzufordern. Der Server antwortet auf dem UDP-Port 67, indem er dem Client eine IP-Adresse zuweist und ihm weitere wichtige Informationen über das Netzwerk, wie Netzmasken, Router und DNS-Server mitteilt. Diese Informationen werden als *DHCP-Lease* bezeichnet und sind nur für eine bestimmte Zeit, die vom Administrator des DHCP-Servers vorgegeben wird, gültig. Dadurch fallen verwaiste IP-Adressen, deren Clients nicht mehr mit dem Netzwerk verbunden sind, automatisch an den Server zurück.

DHCP-Clients können sehr viele Informationen von einem DHCP-Server erhalten. Eine ausführliche Liste finden Sie in dhcp-options(5).

30.5.4. Integration in FreeBSD

FreeBSD verwendet den DHCP-Client von OpenBSD. Sowohl während der Installation als auch im Basissystem steht der DHCP-Client zur Verfügung. In Netzen mit DHCP-Servern wird dadurch die Konfiguration von Systemen erheblich vereinfacht.

DHCP wird von **sysinstall** unterstützt. Wenn Sie eine Netzwerkkarte mit **sysinstall** konfigurieren, lautet die zweite Frage “Do you want to try DHCP configuration of the interface?”. Wenn Sie diese Frage bejahen, wird `dhclient` aufgerufen, und die Netzkarte wird automatisch eingerichtet.

Um DHCP beim Systemstart zu aktivieren, müssen Sie zwei Dinge erledigen:

- Stellen Sie sicher, dass `bpf` in Ihren Kernel kompiliert ist. Dazu fügen Sie die Zeile `device bpf` in Ihre Kernelkonfigurationsdatei ein und erzeugen einen neuen Kernel. Weitere Informationen zur Kernelkonfiguration finden Sie in Kapitel 9 des Handbuchs.

  Das Gerät `bpf` ist im `GENERIC`-Kernel bereits enthalten. Für die Nutzung von DHCP muss also kein angepasster Kernel erzeugt werden.

Anmerkung: Wenn Sie um die Sicherheit Ihres Systems besorgt sind, sollten Sie wissen, dass `bpf` auch zur Ausführung von Paketsniffern erforderlich ist (obwohl diese dennoch als `root` ausgeführt werden müssen). `bpf` *muss* vorhanden sein, damit DHCP funktioniert. Sind Sie sehr sicherheitsbewusst, sollten Sie `bpf` aus Ihrem Kernel entfernen, wenn Sie DHCP nicht verwenden.

- Standardmässig läuft die DHCP-Konfiguration bei FreeBSD im Hintergrund oder auch *asynchron*. Andere Startskripte laufen weiter, während DHCP fertig abgearbeitet wird, was den Systemstart beschleunigt.

  DHCP im Hintergrund funktioniert gut, wenn der DHCP-Server schnell auf Anfragen antwortet und der DHCP-Konfigurationsprozess ebenso schnell abläuft. Jedoch kann DHCP eine lange Zeit benötigen, um auf manchen Systemen fertig zu werden. Falls Netzwerkdienste versuchen, vor DHCP zum Ende zu kommen, werden diese fehlschlagen. Durch die Verwendung von DHCP im *asynchronen*-Modus wird das Problem verhindert, so dass die Startskripte pausiert werden, bis die DHCP-Konfiguration abgeschlossen ist.

  Um sich zu einem DHCP-Server im Hintergrund zu verbinden, während andere Startskripte fortfahren (asynchroner Modus), benutzen Sie den "`DHCP`"-Wert in `/etc/rc.conf`:

  ```
  ifconfig_fxp0="DHCP"
  ```

  Um den Start zu pausieren, damit DHCP vorher abgeschlossen werden kann, benutzen Sie den synchronen Modus mit dem Eintrag "`SYNCDHCP`":

  ```
  ifconfig_fxp0="SYNCDHCP"
  ```

  Anmerkung: Ersetzen Sie *`fxp0`*, das in diesen Beispielen verwendet wurde, durch den Namen Ihrer Netzwerkschnittstelle, so wie es in Abschnitt 12.8 beschrieben ist.

  Wenn Sie `dhclient` an einem anderen Ort installiert haben, oder zusätzliche Flags an `dhclient` übergeben wollen, fügen Sie auch folgende (entsprechend angepasste) Zeilen ein:

  ```
  dhclient_program="/sbin/dhclient"
  dhclient_flags=""
  ```

Der DHCP-Server **dhcpd** ist als Teil des Ports `net/isc-dhcp42-server` verfügbar. Dieser Port enthält die komplette ISC-DHCP-Distribution, inklusive der Dokumentation.

30.5.5. Dateien

- `/etc/dhclient.conf`

  `dhclient` benötigt die Konfigurationsdatei `/etc/dhclient.conf`. Diese Datei enthält normalerweise nur Kommentare, da die Vorgabewerte zumeist ausreichend sind. Lesen Sie dazu auch dhclient.conf(5).

- `/sbin/dhclient`

  `dhclient` ist statisch gelinkt und befindet sich in `/sbin`. Weitere Informationen finden Sie in dhclient(8).

- `/sbin/dhclient-script`

  Bei `dhclient-script` handelt es sich um das FreeBSD-spezifische Konfigurationsskript des DHCP-Clients. Es wird in dhclient-script(8) beschrieben und kann meist unverändert übernommen werden.

- `/var/db/dhclient.leases`

  Der DHCP-Client verfügt über eine Datenbank, die alle derzeit gültigen Leases enthält und als Logdatei erzeugt wird. Weitere Informationen finden Sie in dhclient(8).

30.5.6. Weitere Informationen

Das DHCP-Protokoll wird vollständig im RFC 2131 (http://www.freesoft.org/CIE/RFC/2131/) beschrieben. Eine weitere, lehrreiche Informationsquelle existiert unter http://www.dhcp.org/.

30.5.7. Einen DHCP-Server installieren und einrichten

30.5.7.1. Übersicht

Dieser Abschnitt beschreibt die Einrichtung eines FreeBSD-Systems als DHCP-Server. Dazu wird die DHCP-Implementation von ISC (Internet Systems Consortium) verwendet.

Der DHCP-Server ist nicht im Basissystem von FreeBSD enthalten, daher müssen Sie als Erstes den Port `net/isc-dhcp42-server` installieren. Lesen Sie Kapitel 5, wenn Sie weitere Informationen zur Ports-Sammlung benötigen.

30.5.7.2. Den DHCP-Server installieren

Stellen Sie sicher, dass bpf(4) in Ihren Kernel kompiliert ist. Dazu fügen Sie die Zeile `device bpf` Ihre Kernelkonfigurationsdatei ein und erzeugen einen neuen Kernel. Die Kernelkonfiguration wird in Kapitel 9 beschrieben.

Das Gerät `bpf` ist im `GENERIC`-Kernel bereits enthalten. Für die Nutzung von DHCP muss also kein angepasster Kernel erzeugt werden.

> **Anmerkung:** Wenn Sie um die Sicherheit Ihres Systems besorgt sind, sollten Sie wissen, dass `bpf` auch zur Ausführung von Paketsniffern erforderlich ist (obwohl diese dennoch als `root` ausgeführt werden müssen). `bpf` *muss* vorhanden sein, damit DHCP funktioniert. Sind Sie sehr sicherheitsbewusst, sollten Sie `bpf` aus Ihrem Kernel entfernen, wenn Sie DHCP nicht verwenden.

Danach müssen Sie die vom Port `net/isc-dhcp42-server` erzeugte Vorlage für `dhcpd.conf` anpassen. Die bei der Installation erzeugte Datei `/usr/local/etc/dhcpd.conf.sample` sollten Sie nach `/usr/local/etc/dhcpd.conf` kopieren, bevor Sie Veränderungen vornehmen.

30.5.7.3. Den DHCP-Server einrichten

`dhcpd.conf` besteht aus Festlegungen zu Subnetzen und Rechnern und lässt sich am besten an einem Beispiel erklären:

```
option domain-name "example.com";❶
option domain-name-servers 192.168.4.100;❷
option subnet-mask 255.255.255.0;❸
```

```
default-lease-time 3600;❹
max-lease-time 86400;❺
ddns-update-style none;❻

subnet 192.168.4.0 netmask 255.255.255.0 {
  range 192.168.4.129 192.168.4.254;❼
  option routers 192.168.4.1;❽
}

host mailhost {
  hardware ethernet 02:03:04:05:06:07;❾
  fixed-address mailhost.example.com;(10)
}
```

❶ Diese Option beschreibt die Domäne, die den Clients als Standardsuchdomäne zugewiesen wird. Weitere Informationen finden Sie in man.resolv.conf.5;.

❷ Diese Option legt eine, durch Kommata getrennte Liste von DNS-Servern fest, die von den Clients verwendet werden sollen.

❸ Die den Clients zugewiesene Netzmaske.

❹ Ein Client kann eine Lease einer bestimmten Dauer anfordern. Geschieht dies nicht, weist der Server eine Lease mit einer vorgegebenen Ablaufdauer (in Sekunden) zu.

❺ Die maximale Zeitdauer, für die der Server Konfigurationsinformationen vergibt. Sollte ein Client eine längere Zeitspanne anfordern, wird dennoch nur der Wert `max-lease-time` in Sekunden zugewiesen.

❻ Diese Option legt fest, ob der DHCP-Server eine DNS-Aktualisierung versuchen soll, wenn Konfigurationsdateien vergeben oder zurückgezogen werden. In der ISC-Implementation *muss* diese Option gesetzt sein.

❼ Dadurch werden die IP-Adressen festgelegt, die den Clients zugewiesen werden können. IP-Adressen zwischen diesen Grenzen sowie die einschließenden Adressen werden den Clients zugewiesen.

❽ Legt das Standard-Gateway fest, das den Clients zugewiesen wird.

❾ Die (Hardware-)MAC-Adresse eines Rechners (durch die der DHCP-Server den Client erkennt, der eine Anforderung an ihn stellt).

(10) Einem Rechner soll immer die gleiche IP-Adresse zugewiesen werden. Beachten Sie, dass hier auch ein Rechnername gültig ist, da der DHCP-Server den Rechnernamen auflöst, bevor er die Konfigurationsinformationen zuweist.

Nachdem Sie `dhcpd.conf` fertig konfiguriert haben, sollten Sie den DHCP-Server aktivieren, indem Sie folgende Zeilen in `/etc/rc.conf` aufnehmen:

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Dabei müssen Sie den Geräteeintrag `dc0` durch die Gerätedatei (mehrere Gerätedateien müssen durch Leerzeichen getrennt werden) ersetzen, die Ihr DHCP-Server auf Anfragen von DHCP-Clients hin überwachen soll.

Danach können Sie den Server durch Eingabe des folgenden Befehls starten:

```
# /usr/local/etc/rc.d/isc-dhcpd start
```

Sollten Sie die Konfiguration Ihres Servers einmal verändern müssen, reicht es nicht aus, ein `SIGHUP`-Signal an **dhcpd** zu senden, weil damit die Konfiguration *nicht* erneut geladen wird (im Gegensatz zu den meisten Daemonen). Sie müssen den Prozess vielmehr mit dem Signal `SIGTERM` stoppen, um ihn anschließend neu zu starten.

30.5.7.4. Dateien

- `/usr/local/sbin/dhcpd`

  dhcpd ist statisch gelinkt und befindet sich in `/usr/local/sbin`. Lesen Sie auch die mit dem Port installierte Hilfeseite dhcpd(8), wenn Sie weitere Informationen zu **dhcpd** benötigen.

- `/usr/local/etc/dhcpd.conf`

  dhcpd benötigt die Konfigurationsdatei `/usr/local/etc/dhcpd.conf`, damit der Server den Clients seine Dienste anbieten kann. Diese Datei muss alle Informationen enthalten, die an die Clients weitergegeben werden soll. Außerdem sind hier Informationen zur Konfiguration des Servers enthalten. Die mit dem Port installierte Hilfeseite dhcpd.conf(5) enthält weitere Informationen.

- `/var/db/dhcpd.leases`

  Der DHCP-Server hat eine Datenbank, die alle vergebenen Leases enthält. Diese wird als Logdatei erzeugt. Weitere Informationen finden Sie in der vom Port installierten Hilfeseite dhcpd.leases(5).

- `/usr/local/sbin/dhcrelay`

  dhcrelay wird in komplexen Umgebungen verwendet, in denen ein DHCP-Server eine Anfrage eines Clients an einen DHCP-Server in einem separaten Netzwerk weiterleitet. Wenn Sie diese Funktion benötigen, müssen Sie den Port `net/isc-dhcp42-relay` installieren. Weitere Informationen zu diesem Thema finden Sie in dhcrelay(8).

30.6. DNS – Domain Name Service

Beigetragen von Chern Lee, Tom Rhodes und Daniel Gerzo.

30.6.1. Überblick

DNS ist das für die Umwandlung von Rechnernamen in IP-Adressen zuständige Protokoll. FreeBSD verwendet dazu BIND (Berkeley Internet Name Domain), die am häufigsten verwendete Implementierung von DNS). Eine Anfrage nach `www.FreeBSD.org` gibt die IP-Adresse des FreeBSD-Webservers, eine Anfrage nach `ftp.FreeBSD.org` die IP-Adresse des entsprechenden FTP-Servers zurück. Der umgekehrte Weg ist ebenso möglich, eine IP-Adresse kann also auch in ihren Rechnernamen aufgelöst werden. Um eine DNS-Abfrage durchzuführen, muss auf dem jeweiligen Rechner kein Nameserver installiert sein.

FreeBSD verwendet derzeit in der Voreinstellung BIND9 als DNS-Serversoftware. Unsere Installation bietet Ihnen eine erhöhte Sicherheit, ein neues Dateisystemlayout sowie eine automatisierte chroot(8)-Konfiguration.

Im Internet wird DNS durch ein komplexes System von autoritativen Root-Nameservern, Top Level Domain-Servern (TLD) sowie anderen kleineren Nameservern verwaltet, die individuelle Rechnerinformationen speichern und untereinander abgleichen.

Derzeit wird BIND vom Internet Systems Consortium (https://www.isc.org/) verwaltet.

30.6.2. Begriffsbestimmungen

Um dieses Dokument besser verstehen zu können, müssen einige DNS-spezifische Begriffe genauer definiert werden.

Begriff	Bedeutung
Forward-DNS	Rechnernamen in IP-Adressen umwandeln.
Origin (Ursprung)	Die in einer bestimmten Zonendatei beschriebene Domäne.
named, BIND	Gebräuchliche Namen für das unter FreeBSD verwendete BIND-Nameserverpaket.
Resolver	Ein Systemprozess, durch den ein Rechner Zoneninformationen von einem Nameserver anfordert.
Reverse-DNS	die Umwandlung von IP-Adressen in Rechnernamen
Root-Zone	Der Beginn der Internet-Zonenhierarchie. Alle Zonen befinden sich innerhalb der Root-Zone. Dies ist analog zu einem Dateisystem, in dem sich alle Dateien und Verzeichnisse innerhalb des Wurzelverzeichnisses befinden.
Zone	Eine individuelle Domäne, Unterdomäne, oder ein Teil von DNS, der von der gleichen Autorität verwaltet wird.

Es folgen nun einige Zonenbeispiele:

- Innerhalb der Dokumentation wird die Root-Zone in der Regel mit `.` bezeichnet.
- `org.` ist eine Top level Domain (TLD) innerhalb der Root-Zone.
- `example.org.` ist eine Zone innerhalb der `org.`-TLD.
- `1.168.192.in-addr.arpa.` ist die Zone mit allen IP-Adressen des `192.168.1.*`-IP-Bereichs.

Wie man an diesen Beispielen erkennen kann, befindet sich der spezifischere Teil eines Rechnernamens auf der linken Seite der Adresse. `example.org.` beschreibt einen Rechner also genauer als `org.`, während `org.` genauer als die Root-Zone ist. Jeder Teil des Rechnernamens hat Ähnlichkeiten mit einem Dateisystem, in dem etwa `/dev` dem Wurzelverzeichnis untergeordnet ist.

30.6.3. Gründe für die Verwendung eines Nameservers

Es gibt zwei Arten von Nameservern: Autoritative Nameserver sowie zwischenspeichernde (cachende, auch bekannt als auflösende) Nameserver.

Ein autoritativer Nameserver ist notwendig, wenn

- Sie anderen verbindliche DNS-Auskünfte erteilen wollen.
- eine Domain, beispielsweise `example.org`, registriert wird, und den zu dieser Domain gehörenden Rechnern IP-Adressen zugewiesen werden müssen.
- ein IP-Adressblock reverse-DNS-Einträge benötigt, um IP-Adressen in Rechnernamen auflösen zu können.
- ein Backup-Nameserver (auch Slaveserver genannt) oder ein zweiter Nameserver auf Anfragen antworten soll.

Ein cachender Nameserver ist notwendig, weil

- ein lokaler DNS-Server Daten zwischenspeichern und daher schneller auf Anfragen reagieren kann als ein entfernter Server.

Wird nach `www.FreeBSD.org` gesucht, leitet der Resolver diese Anfrage an den Nameserver des ISPs weiter und nimmt danach das Ergebnis der Abfrage entgegen. Existiert ein lokaler, zwischenspeichernder DNS-Server, muss dieser die Anfrage nur einmal nach außen weitergeben. Für alle weiteren Anfragen ist dies nicht mehr nötig, da diese Information nun lokal gespeichert ist.

30.6.4. Wie funktioniert DNS?

Unter FreeBSD wird der BIND-Daemon als **named** bezeichnet.

Datei	Beschreibung
named	Der BIND-Daemon.
rndc(8)	Das Steuerprogramm für **named**.
`/etc/namedb`	Das Verzeichnis, in dem sich die Zoneninformationen für BIND befinden.
`/etc/namedb/named.conf`	Die Konfigurationsdatei für **named**.

Je nachdem, wie eine Zone auf dem Server konfiguriert wurde, finden sich die zur Zone gehörendenden Dateien in den Unterverzeichnissen `master`, `slave`, oder `dynamic` des Verzeichnisses `/etc/namedb`. Diese Dateien enthalten die DNS-Informationen, die der Nameserver für die Beantwortung von Anfragen benötigt.

30.6.5. BIND starten

Da BIND automatisch installiert wird, ist die Konfiguration relativ einfach.

In der Voreinstellung wird ein in einer chroot(8)-Umgebung betriebener **named**-Server zur einfachen Namensauflösung eingerichtet, der nur im lokalen IPv4-Loopback-Adressbereich (127.0.0.1) lauscht. Um den Server manuell zu starten, verwenden Sie den folgenden Befehl:

```
# /etc/rc.d/named onestart
```

Um den **named**-Daemon beim Systemstart automatisch zu starten, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
named_enable="YES"
```

`/etc/namedb/named.conf` bietet zahlreiche Konfigurationsoptionen, die in diesem Dokument nicht alle beschrieben werden können. Wollen Sie die Startoptionen von **named** unter FreeBSD anpassen, sollten Sie sich die `named_*`-Flags in der Datei `/etc/defaults/rc.conf` sowie die Manualpage zu rc.conf(5) näher ansehen. Zusätzliche Informationen bietet Ihnen auch der Abschnitt Abschnitt 12.7 des Handbuchs.

30.6.6. Konfigurationsdateien

Die Konfigurationsdateien von **named** finden sich unter `/etc/namedb` und müssen in der Regel an Ihre Bedürfnisse angepasst werden. Es sei denn, Sie benötigen nur einen einfachen Resolver. Ein Großteil der Konfigurationsarbeiten erfolgt dabei in diesem Verzeichnis.

30.6.6.1. `/etc/namedb/named.conf`

```
// $FreeBSD$
//
// Refer to the named.conf(5) and named(8) man pages, and the documentation
// in /usr/share/doc/bind9 for more details.
//
// If you are going to set up an authoritative server, make sure you
// understand the hairy details of how DNS works.  Even with
// simple mistakes, you can break connectivity for affected parties,
// or cause huge amounts of useless Internet traffic.

options {
// All file and path names are relative to the chroot directory,
// if any, and should be fully qualified.
    directory  "/etc/namedb/working";
       pid-file        "/var/run/named/pid";
       dump-file       "/var/dump/named_dump.db";
       statistics-file "/var/stats/named.stats";

// If named is being used only as a local resolver, this is a safe default.
// For named to be accessible to the network, comment this option, specify
// the proper IP address, or delete this option.
       listen-on       { 127.0.0.1; };

// If you have IPv6 enabled on this system, uncomment this option for
// use as a local resolver.  To give access to the network, specify
// an IPv6 address, or the keyword "any".
//     listen-on-v6    { ::1; };

// These zones are already covered by the empty zones listed below.
// If you remove the related empty zones below, comment these lines out.
       disable-empty-zone "255.255.255.255.IN-ADDR.ARPA";
       disable-empty-zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.AR
       disable-empty-zone "1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.AR

// If you’ve got a DNS server around at your upstream provider, enter
// its IP address here, and enable the line below.  This will make you
// benefit from its cache, thus reduce overall DNS traffic in the Internet.
/*
       forwarders {
               127.0.0.1;
       };
*/

// If the ’forwarders’ clause is not empty the default is to ’forward first’
// which will fall back to sending a query from your local server if the name
// servers in ’forwarders’ do not have the answer.  Alternatively you can
// force your name server to never initiate queries of its own by enabling the
// following line:
//     forward only;

// If you wish to have forwarding configured automatically based on
```

```
// the entries in /etc/resolv.conf, uncomment the following line and
// set named_auto_forward=yes in /etc/rc.conf.  You can also enable
// named_auto_forward_only (the effect of which is described above).
//     include "/etc/namedb/auto_forward.conf";
```

Um vom Cache Ihres Internetproviders zu profitieren, können hier `forwarders` aktiviert werden. Normalerweise sucht ein Nameserver das Internet rekursiv ab, bis er die gesuchte Antwort findet. Durch diese Option wird stets der Nameserver Ihres Internetproviders zuerst abgefragt, um von dessen Cache zu profitieren. Wenn es sich um einen schnellen, viel benutzten Nameserver handelt, kann dies zu einer Geschwindigkeitssteigerung führen.

> **Warnung:** `127.0.0.1` funktioniert hier *nicht*. Ändern Sie diese Adresse in einen Nameserver Ihres Einwahlproviders.

```
        /*
          Modern versions of BIND use a random UDP port for each outgoing
          query by default in order to dramatically reduce the possibility
          of cache poisoning.  All users are strongly encouraged to utilize
          this feature, and to configure their firewalls to accommodate it.

          AS A LAST RESORT in order to get around a restrictive firewall
          policy you can try enabling the option below.  Use of this option
          will significantly reduce your ability to withstand cache poisoning
          attacks, and should be avoided if at all possible.

          Replace NNNNN in the example with a number between 49160 and 65530.
       */
       // query-source address * port NNNNN;
};

// If you enable a local name server, don't forget to enter 127.0.0.1
// first in your /etc/resolv.conf so this server will be queried.
// Also, make sure to enable it in /etc/rc.conf.

// The traditional root hints mechanism. Use this, OR the slave zones below.
zone "." { type hint; file "/etc/namedb/named.root"; };

/*     Slaving the following zones from the root name servers has some
       significant advantages:
       1. Faster local resolution for your users
       2. No spurious traffic will be sent from your network to the roots
       3. Greater resilience to any potential root server failure/DDoS

       On the other hand, this method requires more monitoring than the
       hints file to be sure that an unexpected failure mode has not
       incapacitated your server.  Name servers that are serving a lot
       of clients will benefit more from this approach than individual
       hosts.  Use with caution.

       To use this mechanism, uncomment the entries below, and comment
       the hint zone above.
```

```
        As documented at http://dns.icann.org/services/axfr/ these zones:
        "." (the root), ARPA, IN-ADDR.ARPA, IP6.ARPA, and ROOT-SERVERS.NET
        are availble for AXFR from these servers on IPv4 and IPv6:
        xfr.lax.dns.icann.org, xfr.cjr.dns.icann.org
*/
/*
zone "." {
        type slave;
        file "/etc/namedb/slave/root.slave";
        masters {
                192.5.5.241;    // F.ROOT-SERVERS.NET.
        };
        notify no;
};
zone "arpa" {
        type slave;
        file "/etc/namedb/slave/arpa.slave";
        masters {
                192.5.5.241;    // F.ROOT-SERVERS.NET.
        };
        notify no;
};
*/

/*      Serving the following zones locally will prevent any queries
        for these zones leaving your network and going to the root
        name servers.  This has two significant advantages:
        1. Faster local resolution for your users
        2. No spurious traffic will be sent from your network to the roots
*/
// RFCs 1912 and 5735 (and BCP 32 for localhost)
zone "localhost"        { type master; file "/etc/namedb/master/localhost-forward.db"; };
zone "127.in-addr.arpa" { type master; file "/etc/namedb/master/localhost-reverse.db"; };
zone "255.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// RFC 1912-style zone for IPv6 localhost address
zone "0.ip6.arpa"       { type master; file "/etc/namedb/master/localhost-reverse.db"; };

// "This" Network (RFCs 1912 and 5735)
zone "0.in-addr.arpa"   { type master; file "/etc/namedb/master/empty.db"; };

// Private Use Networks (RFCs 1918 and 5735)
zone "10.in-addr.arpa"     { type master; file "/etc/namedb/master/empty.db"; };
zone "16.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "17.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "18.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "19.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "20.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "21.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "22.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "23.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "24.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
```

```
zone "25.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "26.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "27.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "28.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "29.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "30.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "31.172.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "168.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// Link-local/APIPA (RFCs 3927 and 5735)
zone "254.169.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IETF protocol assignments (RFCs 5735 and 5736)
zone "0.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// TEST-NET-[1-3] for Documentation (RFCs 5735 and 5737)
zone "2.0.192.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "100.51.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "113.0.203.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Range for Documentation (RFC 3849)
zone "8.b.d.0.1.0.0.2.ip6.arpa" { type master; file "/etc/namedb/master/empty.db"; };


// Domain Names for Documentation and Testing (BCP 32)
zone "test" { type master; file "/etc/namedb/master/empty.db"; };
zone "example" { type master; file "/etc/namedb/master/empty.db"; };
zone "invalid" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.com" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.net" { type master; file "/etc/namedb/master/empty.db"; };
zone "example.org" { type master; file "/etc/namedb/master/empty.db"; };


// Router Benchmark Testing (RFCs 2544 and 5735)
zone "18.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "19.198.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };

// IANA Reserved - Old Class E Space (RFC 5735)
zone "240.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "241.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "242.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "243.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "244.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "245.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "246.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "247.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "248.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "249.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "250.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "251.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "252.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "253.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
zone "254.in-addr.arpa" { type master; file "/etc/namedb/master/empty.db"; };
```

```
// IPv6 Unassigned Addresses (RFC 4291)
zone "1.ip6.arpa"       { type master; file "/etc/namedb/master/empty.db"; };
zone "3.ip6.arpa"       { type master; file "/etc/namedb/master/empty.db"; };
zone "4.ip6.arpa"       { type master; file "/etc/namedb/master/empty.db"; };
zone "5.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "6.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "7.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "8.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "9.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "a.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "b.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "c.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "d.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "e.ip6.arpa"               { type master; file "/etc/namedb/master/empty.db"; };
zone "0.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "1.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "2.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "3.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "4.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "5.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "6.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "7.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "8.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "9.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "a.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "b.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "0.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "1.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "2.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "3.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "4.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "5.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "6.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "7.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 ULA (RFC 4193)
zone "c.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };
zone "d.f.ip6.arpa"             { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Link Local (RFC 4291)
zone "8.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "9.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "a.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "b.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };

// IPv6 Deprecated Site-Local Addresses (RFC 3879)
zone "c.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "d.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "e.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };
zone "f.e.f.ip6.arpa"           { type master; file "/etc/namedb/master/empty.db"; };

// IP6.INT is Deprecated (RFC 4159)
```

```
zone "ip6.int"                { type master; file "/etc/namedb/master/empty.db"; };

// NB: Do not use the IP addresses below, they are faked, and only
// serve demonstration/documentation purposes!
//
// Example slave zone config entries.  It can be convenient to become
// a slave at least for the zone your own domain is in.  Ask
// your network administrator for the IP address of the responsible
// master name server.
//
// Do not forget to include the reverse lookup zone!
// This is named after the first bytes of the IP address, in reverse
// order, with ".IN-ADDR.ARPA" appended, or ".IP6.ARPA" for IPv6.
//
// Before starting to set up a master zone, make sure you fully
// understand how DNS and BIND work.  There are sometimes
// non-obvious pitfalls.  Setting up a slave zone is usually simpler.
//
// NB: Don't blindly enable the examples below. :-)  Use actual names
// and addresses instead.

/* An example dynamic zone
key "exampleorgkey" {
       algorithm hmac-md5;
       secret "sf87HJqjkqh8ac87a02lla==";
};
zone "example.org" {
       type master;
       allow-update {
               key "exampleorgkey";
       };
       file "/etc/named/dynamic/example.org";
};
*/

/* Example of a slave reverse zone
zone "1.168.192.in-addr.arpa" {
       type slave;
       file "/etc/namedb/slave/1.168.192.in-addr.arpa";
       masters {
               192.168.1.1;
       };
};
*/
```

Hierbei handelt es sich um Slave-Einträge für eine Reverse- und Forward-DNS-Zone, die in der Datei `named.conf` definiert sind.

Für jede neue Zone muss ein zusätzlicher Eintrag in `named.conf` erstellt werden.

Ein einfacher Eintrag für eine Zone `example.org` könnte beispielsweise so aussehen:

```
zone "example.org" {
        type master;
```

```
        file "master/example.org";
};
```

Die Option `type` legt fest, dass es sich um eine Master-Zone handelt, deren Zoneninformationen sich in der Datei `/etc/namedb/master/example.org` befinden. Diese Datei wird durch die Option `file` festgelegt.

```
zone "example.org" {
        type slave;
        file "slave/example.org";
};
```

Hier handelt es sich um einen Slaveserver, der seine Informationen vom Masterserver der betreffenden Zone bezieht und diese in der angegebenen Datei speichert. Wenn der Masterserver nicht erreichbar ist, verfügt der Slaveserver über die transferierten Zoneninformationen und kann diese an andere Rechner weitergeben.

30.6.6.2. Zonendateien

Die in der Datei `/etc/namedb/master/example.org` definierte Zonendatei für `example.org` könnte etwa so aussehen:

```
$TTL 3600        ; 1 hour default TTL
example.org.    IN      SOA      ns1.example.org. admin.example.org. (
                                2006051501      ; Serial
                                10800           ; Refresh
                                3600            ; Retry
                                604800          ; Expire
                                300             ; Negative Response TTL
                        )

; DNS Servers
                IN      NS      ns1.example.org.
                IN      NS      ns2.example.org.

; MX Records
                IN      MX 10   mx.example.org.
                IN      MX 20   mail.example.org.

                IN      A       192.168.1.1

; Machine Names
localhost       IN      A       127.0.0.1
ns1             IN      A       192.168.1.2
ns2             IN      A       192.168.1.3
mx              IN      A       192.168.1.4
mail            IN      A       192.168.1.5

; Aliases
www             IN      CNAME   example.org.
```

Beachten Sie, dass jeder mit einem “.” endende Rechnername ein exakter Rechnername ist, während sich alles ohne einen abschließenden “.” relativ auf den Ursprung bezieht. `ns1` steht daher beispielsweise für `ns1.`*`example.org.`*.

Eine Zonendatei hat folgenden Aufbau:

```
recordname      IN recordtype   value
```

Die am häufigsten verwendeten DNS-Einträge sind:

SOA

: Start der Zonenautorität

NS

: Ein autoritativer Nameserver

A

: Eine Rechneradresse

CNAME

: Der kanonische Name eines Alias

MX

: Mail Exchanger

PTR

: Ein (bei Reverse-DNS verwendeter) Domain Name Pointer

```
example.org. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh after 3 hours
                        3600            ; Retry after 1 hour
                        604800          ; Expire after 1 week
                        300 )           ; Negative Response TTL
```

`example.org.`

: Der Name der Domäne und damit der Ursprung dieser Zonendatei.

`ns1.example.org.`

: Der primäre/autoritative Nameserver dieser Zone.

`admin.example.org.`

: Die für diese Zone verantwortliche Person. Das Zeichen "@" wird dabei ersetzt (<`admin@example.org`> wird also zu `admin.example.org`).

`2006051501`

: Die Seriennummer der Datei. Sie muss stets inkrementiert werden, wenn die Zonendatei geändert wird. Viele Administratoren bevorzugen ein `JJJJMMTTRR`-Format, um die Seriennummer festzulegen. `2006051501` steht also für den 15.05.2006, die beiden letzten Stellen für die erste Modifikation der Zonendatei an diesem Tag. Die Seriennummer ist von großer Bedeutung, da Slaveserver daran eine aktualisierte Zonendatei erkennen können.

```
        IN NS           ns1.example.org.
```

Ein NS-Eintrag. Jeder Nameserver, der für eine Zone verantwortlich ist, muss über einen solchen Eintrag verfügen.

```
        localhost       IN      A       127.0.0.1
        ns1             IN      A       192.168.1.2
        ns2             IN      A       192.168.1.3
        mx              IN      A       192.168.1.4
        mail            IN      A       192.168.1.5
```

Der Eintrag `A` bezieht sich auf Rechnernamen. `ns1.example.org` würde also zu `192.168.1.2` aufgelöst werden.

```
        IN      A       192.168.1.1
```

Diese Zeile weist die IP-Adresse `192.168.1.1` dem aktuellen Ursprung, in unserem Fall also `example.org`, zu.

```
        www             IN CNAME        @
```

Der Eintrag für den kanonischen Namen wird dazu verwendet, Aliase für einen Rechner zu vergeben. Im Beispiel ist `www` ein Alias für den "Master"-Rechner, dessen Name dem Domainnamen `example.org` (oder `192.168.1.1`) entspricht. CNAMEs können daher niemals gleichzeitig mit einem anderen Eintrag für denselben Hostname eingerichtet werden.

```
        IN MX   10      mail.example.org.
```

Die Option MX legt fest, welcher Mailserver für eintreffende Mails der Zone verantwortlich ist. `mail.example.org` ist der Rechnername des Mailservers, der eine Priorität von 10 hat.

Es können auch mehrere Mailserver mit verschiedener Priorität (10, 20, ...) vorhanden sein. Ein Mailserver, der eine Mail an `example.org` verschicken will, verwendet zuerst den MX mit der höchsten Priorität (das heißt den mit der niedrigsten Prioritätsnummer), danach den mit der nächsthöheren Priorität. Und dies solange, bis die E-Mail zugestellt werden kann.

Für (bei Reverse-DNS verwendete) `in-addr.arpa`-Zonendateien wird das gleiche Format verwendet. Der einzige Unterschied besteht in der Verwendung der Option PTR an Stelle der Optionen A und CNAME.

```
$TTL 3600

1.168.192.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
                        2006051501      ; Serial
                        10800           ; Refresh
                        3600            ; Retry
                        604800          ; Expire
                        300 )           ; Negative Response TTL

        IN      NS      ns1.example.org.
        IN      NS      ns2.example.org.

1       IN      PTR     example.org.
2       IN      PTR     ns1.example.org.
3       IN      PTR     ns2.example.org.
4       IN      PTR     mx.example.org.
5       IN      PTR     mail.example.org.
```

Durch diese Datei werden den Rechnernamen der fiktiven Domäne IP-Adressen zugewiesen.

Beachten Sie bitte, dass es sich bei allen Namen auf der rechten Seite eines PTR-Eintrags um absolute (*fully qualified*) Domainnamen handeln muss, die mit "." enden.

30.6.7. Zwischenspeichernde (cachende) Nameserver

Ein cachender Nameserver hat primär die Aufgabe, rekursive Abfragen aufzulösen. Er stellt lediglich eigene Anfragen und speichert deren Ergebnisse ab.

30.6.8. DNSSEC

Domain Name System Security Extensions, oder kurz DNSSEC, ist eine Sammlung von Spezifikationen, um auflösende Nameserver von gefälschten DNS-Daten, wie beispielsweise vorgetäuschte DNS-Einträge, zu schützen. Durch die Verwendung von digitalen Signaturen kann ein Resolver die Integrität des Eintrages überprüfen. Wichtig dabei ist, dass DNSSEC nur die Integrität über digital signierte Resource Records (RRe) bereitstellt. Weder wird die Vertraulichkeit noch der Schutz vor falschen Annahmen des Endbenutzers sichergestellt. Dies bedeutet, dass es Leute nicht davor schützen kann, zu `example.net` anstatt zu `example.com` zu gelangen. Das einzige, was DNSSEC tut, ist die Authentifizierung, dass die Daten während der Übertragung nicht verändert wurden. Die Sicherheit von DNS ist ein wichtiger Schritt in der generellen Absicherung des Internets. Für weitere, tiefergehende Details über die Funktionsweise von DNSSEC sind die dazugehörigen RFCs ein guter Einstieg in die Thematik. Sehen Sie sich dazu die Liste in Abschnitt 30.6.10 an.

Der folgende Abschnitt wird zeigen, wie man DNSSEC für einen autoritativen DNS-Server und einen rekursiven (oder cachenden) DNS-Server, der jeweils BIND 9 verwenden, einrichten kann. Obwohl alle Versionen von BIND 9 DNSSEC unterstützen, ist es notwendig, mindestens die Version 9.6.2 zu verwenden, um in der Lage zu sein, die signierten Root-Zonen zu benutzen, wenn DNS-Abfragen geprüft werden. Der Grund dafür ist, dass früheren Versionen die Algorithmen fehlen, um die Überprüfung des Root-Zonenschlüssels zu aktivieren. Es wird dringend empfohlen, die letzte Version von BIND 9.7 oder höher einzusetzen, um von den Vorteilen der automatischen Schlüsselaktualisierung des Root-Zonenschlüssels Gebrauch zu machen, genauso wie andere Eigenschaften, um automatisch Zonen signieren zu lassen und Signaturen aktuell zu halten. Unterschiede zwischen den Versionen 9.6.2 und 9.7 und höher werden an den betreffenden Stellen angesprochen.

30.6.8.1. Rekursive DNS-Server Konfiguration

Die Aktivierung der DNSSEC-Überprüfung von Anfragen, die von einem rekursiven DNS-Server stammen, benötigt ein paar Änderungen in der `named.conf`. Bevor man jedoch diese Änderungen durchführt, muss der Root-Zonenschlüssel oder Vertrauensanker erworben werden. Momentan ist der Root-Zonenschlüssel nicht in einem Dateiformat verfügbar, dass von BIND benutzt werden kann, so dass dieser manuell in das richtige Format konvertiert werden muss. Der Schlüssel selbst kann durch Abfrage an die Root-Zone erhalten werden, indem man dazu **dig** verwendet. Durch Aufruf von

```
% dig +multi +noall +answer DNSKEY . > root.dnskey
```

wird der Schlüssel in `root.dnskey` abgelegt. Der Inhalt sollte so ähnlich wie folgt aussehen:

```
. 93910 IN DNSKEY 257 3 8 (
       AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQ
```

```
       bSEW0O8gcCjFFVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh
       /RStIoO8g0NfnfL2MTJRkxoXbfDaUeVPQuYEhg37NZWA
       JQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaDX6RS6CXp
       oY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3
       LQpzW5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGO
       Yl7OyQdXfZ57relSQageu+ipAdTTJ25AsRTAoub8ONGc
       LmqrAmRLKBP1dfwhYB4N7knNnulqQxA+Uk1ihz0=
       ) ; key id = 19036
. 93910 IN DNSKEY 256 3 8 (
       AwEAAcaGQEA+OJmOzfzVfoYN249JId7gx+OZMbxy69Hf
       UyuGBbRN0+HuTOpBxxBCkNOL+EJB9qJxt+0FEY6ZUVjE
       g58sRr4ZQ6Iu6b1xTBKgc193zUARk4mmQ/PPGxn7Cn5V
       EGJ/1h6dNaiXuRHwR+7oWh7DnzkIJChcTqlFrXDW3tjt
) ; key id = 34525
```

Seien Sie nicht alarmiert, wenn der von Ihnen bezogene Schlüssel anders als in diesem Beispiel aussieht. Diese könnten sich in der Zwischenzeit geändert haben. In dieser Ausgabe sind eigentlich zwei Schlüssel enthalten. Der erste Schüssel mit dem Wert 257 nach dem DNSKEY-Eintrag ist derjenige, der benötigt wird. Der Wert zeigt an, dass es sich um einen sicheren Einstiegspunkt (SEP), gemein auch als Schlüsselsignierungsschlüssel (KSK) bekannt, handelt. Der zweite Schüssel mit dem Wert 256 ist der untergeordnete Schlüssel, im allgemeinen auch als Zonen-Signaturschlüssel (ZSK) bezeichnet. Weitere Schlüsselarten werden später in Abschnitt 30.6.8.2 erläutert.

Nun muss der Schlüssel verifiziert und so formatiert werden, dass BIND diesen verwenden kann. Um den Schlüssel zu verifizieren, erzeugen Sie einen DS RR-Satz. Erstellen Sie eine Datei, welche die RRs enthält, mittels

```
% dnssec-dsfromkey -f root-dnskey . > root.ds
```

Diese Einträge verwenden SHA-1 sowie SHA-256 und sollten ähnlich zu folgendem Beispiel aussehen, in dem der längere, SHA-256, benutzt wird.

```
. IN DS 19036 8 1 B256BD09DC8DD59F0E0F0D8541B8328DD986DF6E
. IN DS 19036 8 2 49AAC11D7B6F6446702E54A1607371607A1A41855200FD2CE1CDDE32F24E8FB5
```

Der SHA-256 RR kann nun mit dem Abriss in https://data.iana.org/root-anchors/root-anchors.xml verglichen werden. Um absolut sicher zu sein, dass der Schlüssel nicht zusammen mit den XML-Daten verändert wurde, kann die Datei mittels der PGP Signatur in https://data.iana.org/root-anchors/root-anchors.asc überprüft werden.

Als nächstes muss der Schlüssel in das passende Format gebracht werden. Dies unterscheidet sich ein bisschen von den BIND Versionen 9.6.2 und 9.7 und höhere. In Version 9.7 wurde die Ünterstützung zur automatischen Verfolgung und notwendigen Aktualisierung von Änderungen am Schlüssel eingebaut. Dies wird durch den Einsatz von `managed-keys` erreicht, wie in dem Beispiel unten gezeigt ist. Wenn die ältere Version eingesetzt wird, kann der Schlüssel durch eine `trusted-keys`-Anweisung eingebaut werden und die Aktualisierung muss händisch erfolgen. In BIND 9.6.2 sollte das Format folgendermassen aussehen:

```
trusted-keys {
       "." 257 3 8
       "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
       FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
       bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
       X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
       W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
       Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
       QxA+Uk1ihz0=";
```

```
};
```

In 9.7 wird das Format stattdessen wie folgt aussehen:

```
managed-keys {
       "." initial-key 257 3 8
       "AwEAAagAIKlVZrpC6Ia7gEzahOR+9W29euxhJhVVLOyQbSEW0O8gcCjF
       FVQUTf6v58fLjwBd0YI0EzrAcQqBGCzh/RStIoO8g0NfnfL2MTJRkxoX
       bfDaUeVPQuYEhg37NZWAJQ9VnMVDxP/VHL496M/QZxkjf5/Efucp2gaD
       X6RS6CXpoY68LsvPVjR0ZSwzz1apAzvN9dlzEheX7ICJBBtuA6G3LQpz
       W5hOA2hzCTMjJPJ8LbqF6dsV6DoBQzgul0sGIcGOYl7OyQdXfZ57relS
       Qageu+ipAdTTJ25AsRTAoub8ONGcLmqrAmRLKBP1dfwhYB4N7knNnulq
       QxA+Uk1ihz0=";
};
```

Der Root-Schlüssel kann nun zu `named.conf` hinzugefügt werden, entweder direkt oder durch Inkludierung der Datei, die den Schlüssel enthält. Nachdem diese Schritte absolviert sind, muss BIND konfiguriert werden, um DNSSEC-Validierung für Anfragen durchzuführen, indem `named.conf` bearbeitet und die folgende `options`-Direktive hinzugefügt wird:

```
dnssec-enable yes;
dnssec-validation yes;
```

Um zu prüfen, dass es tatsächlich funktioniert, benutzen Sie **dig**, um eine Anfrage zu einer signierten Zone durch den Resolver, der gerade konfiguriert wurde, zu stellen. Eine erfolgreiche Antwort wird den `AD`-Eintrag aufweisen, um anzudeuten, dass die Daten authentisiert sind. Eine Anfrage wie

```
% dig @resolver +dnssec se ds
```

sollte den DS RR für die `.se`-Zone zurückgeben. In dem Abschnitt `flags:` sollte der `AD`-Eintrag gesetzt sein, wie im folgenden zu sehen ist:

```
...
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
...
```

Der Resolver ist nun in der Lage, Anfragen ans DNS zu authentisieren.

30.6.8.2. Autoritative DNS-Server Konfiguration

Um einen autoritativen Nameserver dazu zu bringen, als eine DNSSEC-signierte Zone zu fungieren, ist ein wenig mehr Aufwand nötig. Eine Zone ist durch kryptographische Schlüssel signiert, die erzeugt werden müssen. Es ist möglich, nur einen Schlüssel dazu zu verwenden. Die vorgeschlagene Methode ist jedoch, einen starken, gut geschützten Schlüsselsignierungsschlüssel (KSK) einzusetzen, der nicht oft gewechselt wird und einen Zonensignierungsschlüssel (ZSK), der öfter ausgewechselt wird. Informationen zu vorgeschlagenen Einsatzarten können in RFC 4641: DNSSEC Operational Practices (http://tools.ietf.org/rfc/rfc4641.txt) nachgelesen werden. Einsatzszenarien, welche die Root-Zone betreffen, finden Sie in DNSSEC Practice Statement for the Root Zone KSK operator (http://www.root-dnssec.org/wp-content/uploads/2010/06/icann-dps-00.txt) sowie DNSSEC Practice Statement for the Root Zone ZSK operator (http://www.root-dnssec.org/wp-content/uploads/2010/06/vrsn-dps-00.txt). Der KSK wird dazu verwendet, um eine Kette von Autorität für die Daten, die diese Validierung benötigen, zu erschaffen und wird als solche auch als

sicherer Einstiegspunkt (SEP)-Schlüssel bezeichnet. Ein Nachrichtenabriss dieses Schlüssels, der auch Delegation Signer (DS)-Eintrag genannt wird, muss in der Elternzone veröffentlicht werden, um die Vertrauenskette herzustellen. Wie dies erreicht wird, hängt von dem Besitzer der Elternzone ab. Der ZSK wird verwendet, um die Zone zu signieren und muss nur dort öffentlich zugänglich gemacht werden.

Um DNSSEC für die `example.com`-Zone, welche in den vorherigen Beispielen verwendet wird, zu aktivieren, muss als erster Schritt **dnssec-keygen** benutzt werden, um das KSK und ZSK Schlüsselpaar zu generieren. Dieses Schlüsselpaar kann unterschiedliche kryptographische Algorithmen nutzen. Es wird empfohlen, RSA/SHA256 für die Schlüssel zu nutzen. Eine Schlüssellänge von 2048 Bits sollte genügen. Um den KSK für `example.com` zu generieren, geben Sie

```
% dnssec-keygen -f KSK -a RSASHA256 -b 2048 -n ZONE example.com
```

ein und um den ZSK zu erzeugen, setzen Sie folgenden Befehl ab:

```
% dnssec-keygen -a RSASHA256 -b 2048 -n ZONE example.com
```

dnssec-keygen gibt zwei Dateien aus, den öffentlichen und den privaten Schlüssel und zwar in Dateinamen, die ähnlich lauten wie `Kexample.com.+005+nnnnn.key` (öffentlich) und `Kexample.com.+005+nnnnn.private` (privat). Der `nnnnn`-Teil des Dateinamens ist eine fünfstellige Schlüsselkennung. Passen Sie genau auf, welche Kennung zu welchem Schlüssel gehört. Das ist besonders wichtig, wenn mehrere Schlüssel in einer Zone vorliegen. Es ist auch möglich, die Schlüssel umzubenennen. Für jede KSK-Datei tun Sie folgendes:

```
% mv Kexample.com.+005+nnnnn.key Kexample.com.+005+nnnnn.KSK.key
% mv Kexample.com.+005+nnnnn.private Kexample.com.+005+nnnnn.KSK.private
```

Für die ZSK-Dateien ersetzen Sie `KSK` für `ZSK` wenn nötig. Die Dateien können nun in der Zonendatei inkludiert werden, indem die `$include` Anweisung verwendet wird. Es sollte folgendermassen aussehen:

```
$include Kexample.com.+005+nnnnn.KSK.key    ; KSK
$include Kexample.com.+005+nnnnn.ZSK.key    ; ZSK
```

Schliesslich signieren Sie die Zone und weisen BIND an, die signierte Zonendatei zu benutzen. Um eine Zone zu signieren, wird **dnssec-signzone** eingesetzt. Der Befehl, um eine Zone `example.com` zu signieren, die in `example.com.db` liegt, sollte wie folgt aussehen:

```
% dnssec-signzone -o example.com -k Kexample.com.+005+nnnnn.KSK example.com.db Kexample.com.+005+nnnnn.ZSK.k
```

Der Schlüssel, welcher mit dem Argument `-k` übergeben wird, ist der KSK und die andere Schlüsseldatei ist der ZSK, welcher für die Signatur benutzt werden soll. Es ist möglich, mehr als einen KSK und ZSK anzugeben, was das Ergebnis zur Folge hat, dass die Zone mit allen übergebenen Schlüsseln signiert wird. Dies kann dann benötigt werden, um Zonendaten mit mehr als einem Algorithmus zur Signierung zu verwenden. Die Ausgabe von **dnssec-signzone** ist eine Zonendatei mit allen signierten RRs. Diese Ausgabe wird in einer Datei mit der Endung `.signed` abgelegt, wie beispielsweise `example.com.db.signed`. Die DS-Einträge werden ebenfalls in eine separate Datei `dsset-example.com` geschrieben. Um diese signierte Zone zu verwenden, ändern Sie die Zonendirektive in `named.conf`, so dass `example.com.db.signed` benutzt wird. Standardmässig sind die Signaturen nur 30 Tage gültig, was bedeutet, dass die Zone in etwa 15 Tagen erneut signiert werden muss, um sicher zu stellen, dass Resolver keine Einträge mit veralteten Signaturen zwischenspeichern. Es ist möglich, ein Skript und einen cron-Job zu schreiben, um dies zu erledigen. Lesen Sie dazu die relevanten Anleitungen, um Details zu erfahren.

Stellen Sie sicher, dass die privaten Schlüssel vertraulich bleiben, genau wie mit allen anderen kryptographischen Schlüsseln auch. Wenn ein Schlüssel geändert wird, ist es gute Praxis den neuen Schlüssel in die Zone zu inkludieren, noch während der alte Schlüssel noch zum signieren eingesetzt wird, um dann auf den neuen Schlüssel zum signieren zu wechseln. Nachdem diese Schritte erfolgt sind, kann der alte Schlüssel aus der Zone entfernt werden. Wenn das nicht geschieht, können DNS-Daten für einige Zeit nicht verfügbar sein, bis der neue Schlüssel durch die DNS-Hierarchie propagiert wurde. Für weitere Informationen bezüglich Schlüsselübergabe und andere DNSSEC-Einsatzszenarien lesen Sie RFC 4641: DNSSEC Operational practices (http://www.ietf.org/rfc/rfc4641.txt).

30.6.8.3. Automatisierung mittels BIND 9.7 oder höher

Beginnend mit der Version 9.7 von BIND wurde eine neue Eigenschaft vorgestellt, die *Smart Signing* genannt wird. Diese zielt darauf ab, das Schlüsselmanagement und den Signierungsprozess einfacher zu gestalten und zu automatisieren. Durch ablegen der Schlüssel in ein Verzeichnis, genannt *key repository* und die Verwendung der neuen Option `auto-dnssec`, ist es möglich eine dynamische Zone zu erzeugen, welche dann erneut signiert wird, wenn dazu der Bedarf besteht. Um diese Zone zu aktualisieren, benutzen Sie **nsupdate** mit der neuen Option `-l`. Es hat also **rndc** die Fähigkeit gewonnen, Zonen mit Schlüsseln im Key Repository zu verwenden, indem die Option `sign` eingesetzt wird. Um BIND anzuweisen, diese automatische Signierung und Zonenaktualisierung für `example.com` zu nutzen, fügen Sie die folgenden Zeilen zur `named.conf` hinzu:

```
zone example.com {
       type master;
       key-directory "/etc/named/keys";
       update-policy local;
       auto-dnssec maintain;
       file "/etc/named/dynamic/example.com.zone";
};
```

Nachdem diese Änderungen durchgeführt wurden, erzeugen Sie die Schlüssel für die Zone wie in Abschnitt 30.6.8.2 beschrieben wird, legen diese Schlüssel im Key Repository ab, dass als Argument `key-directory` in der Zonenkonfiguration steht und die Zone wird automatisch signiert. Aktualisierungen für eine Zone, die auf diese Art und Weise konfiguriert wurde, muss mittels **nsupdate** erfolgen, dass sich um die erneute Signierung der Zone mit den hinzugefügten Daten kümmern wird. Für weitere Details, lesen Sie Abschnitt 30.6.10 und die Dokumentation von BIND.

30.6.9. Sicherheit

Obwohl BIND die am meisten verwendete (und kontrollierte) Implementierung von DNS darstellt, werden dennoch manchmal neue Sicherheitsprobleme entdeckt.

Zwar startet FreeBSD **named** automatisch in einer chroot(8)-Umgebung, es gibt aber noch weitere Sicherheitsmechanismen, mit denen Sie potentielle DNS-Serviceattacken erschweren können.

Es ist daher eine gute Idee, die Sicherheitshinweise von CERT (http://www.cert.org/) zu lesen sowie die Mailingliste FreeBSD security notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications) zu abonnieren, um sich über Sicherheitsprobleme im Zusammenhang mit dem Internet und FreeBSD zu informieren.

> **Tipp:** Tritt ein Problem auf, kann es nie schaden, die Quellen zu aktualisieren und **named** neu zu kompilieren.

30.6.10. Weitere Informationsquellen

Hilfeseiten zu BIND/**named**: rndc(8) named(8) named.conf(5) nsupdate(8) dnssec-signzone(8) dnssec-keygen(8)

- Offizielle ISC-Seite zu BIND (https://www.isc.org/software/bind)
- Offizielles Forum zu ISC- BIND (https://www.isc.org/software/guild)
- O'Reilly DNS and BIND 5th Edition (http://www.oreilly.com/catalog/dns5/)
- Root DNSSEC (http://www.root-dnssec.org/documentation/)
- DNSSEC Vertrauensanker-Publikation für die Root-Zone (http://data.iana.org/root-anchors/draft-icann-dnssec-trust-anchor.html)
- RFC1034 - Domain Names - Concepts and Facilities (http://tools.ietf.org/html/rfc1034)
- RFC1035 - Domain Names - Implementation and Specification (http://tools.ietf.org/html/rfc1035)
- RFC4033 - DNS Security Introduction and Requirements (http://tools.ietf.org/html/rfc4033)
- RFC4034 - Resource Records for the DNS Security Extensions (http://tools.ietf.org/html/rfc4034)
- RFC4035 - Protocol Modifications for the DNS Security Extensions (http://tools.ietf.org/html/rfc4035)
- RFC4641 - DNSSEC Operational Practices (http://tools.ietf.org/html/rfc4641)
- RFC 5011 - Automated Updates of DNS Security (DNSSEC) Trust Anchors (http://tools.ietf.org/html/rfc5011)

30.7. Der Apache HTTP-Server

Beigetragen von Murray Stokely.

30.7.1. Überblick

Einige der weltgrößten Internetauftritte laufen unter FreeBSD. Die Mehrzahl der Webserver im Internet nutzt den **Apache** HTTP-Server. Die Installationspakete für den **Apache** sollten auf Ihrem Installationsmedium vorhanden sein. Wenn Sie den **Apache** noch nicht installiert haben, können Sie dies jederzeit über den Port `www/apache13` oder `www/apache22` nachholen.

Nachdem der **Apache** erfolgreich installiert wurde, muss er noch konfiguriert werden.

> **Anmerkung:** Dieser Abschnitt beschreibt die Version 1.3.X des **Apache** HTTP-Servers, da diese Version unter FreeBSD am häufigsten verwendet wird. **Apache** 2.X bringt zwar viele Verbesserungen mit sich, wird hier aber nicht beschrieben. Sollten Sie an **Apache** 2.X interessiert sein, informieren Sie sich bitte auf http://httpd.apache.org/.

30.7.2. Konfiguration

Der **Apache** HTTP-Server wird unter FreeBSD primär über die Datei `/usr/local/etc/apache/httpd.conf` konfiguriert. Bei dieser Datei handelt es sich um eine typische UNIX-Konfigurationsdatei, in der Kommentarzeilen mit einem #-Zeichen beginnen. Eine komplette Beschreibung aller Optionen würde den Rahmen dieses Handbuchs sprengen, daher beschreiben wir hier nur die am häufigsten verwendeten Optionen.

`ServerRoot "/usr/local"`

Legt das Standardwurzelverzeichnis für die **Apache**-Installation fest. Binärdateien werden in die Verzeichnisse `bin` und `sbin` unterhalb des Serverwurzelverzeichnisses installiert, während sich Konfigurationsdateien im Verzeichnis `etc/apache` befinden.

`ServerAdmin you@your.address`

Die E-Mail-Adresse, an die Mitteilungen über Serverprobleme geschickt werden sollen. Diese Adresse erscheint auf vom Server erzeugten Seiten, beispielsweise auf Fehlerseiten.

`ServerName www.example.com`

Über die Option `ServerName` können Sie einen Rechnernamen festlegen, den Ihr Server an die Clients sendet, wenn sich dieser von tatsächlichen Rechnernamen unterscheidet (sie könnten etwa `www` statt des richtigen Rechnernamens verwenden).

`DocumentRoot "/usr/local/www/data"`

`DocumentRoot`: Das Verzeichnis, in dem Sie Ihre Dokumente ablegen. In der Voreinstellung befinden sich alle Seiten in diesem Verzeichnis, durch symbolische Links oder Aliase lassen sich aber auch andere Orte festlegen.

Es ist empfehlenswert, eine Sicherungskopie Ihrer Konfigurationsdatei anzulegen, bevor Sie Änderungen durchführen. Nachdem Sie die Konfiguration beendet haben, können Sie den **Apache** starten.

30.7.3. Den Apache betreiben

Der **Apache** wird, im Gegensatz zu vielen anderen Netzwerkservern, nicht vom **inetd**-Super-Server verwaltet, sondern wird als eigenständiger Server betrieben, um die Leistung für eintreffende HTTP-Anfragen von den Clients (also von Internetbrowsern) zu verbessern. Gestartet, beendet oder neu gestartet wird der Server über einen Shellskript-Wrapper. Um den **Apache** erstmals zu starten, geben Sie einfach Folgendes ein:

```
# /usr/local/sbin/apachectl start
```

Wenn Sie den Server beenden wollen, geben Sie Folgendes ein:

```
# /usr/local/sbin/apachectl stop
```

Wenn Sie die Konfigurationsdatei verändern, müssen Sie den Server neu starten:

```
# /usr/local/sbin/apachectl restart
```

Um den **Apache** ohne den Abbruch bestehender Verbindungen neu zu starten, geben Sie Folgendes ein:

```
# /usr/local/sbin/apachectl graceful
```

Diese und weitere Optionen werden in apachectl(8) beschrieben.

Um den **Apache** beim Systemstart zu starten, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
apache_enable="YES"
```

Um **Apache** 2.2 zu starten, fügen Sie hingegen folgende Zeile ein:

```
apache22_enable="YES"
```

Wenn Sie während des Systemstarts weitere Parameter an den **Apache**-`httpd`-Daemon übergeben wollen, können Sie diese durch eine zusätzliche Zeile in `rc.conf` angeben:

```
apache_flags=""
```

Nachdem der Webserver gestartet ist, können Sie sich Ihre Internetseite ansehen, indem Sie in Ihren Browser die Adresse `http://localhost/` eingeben. Die vordefinierte Standardstartseite ist `/usr/local/www/data/index.html`.

30.7.4. Virtual Hosting

Der **Apache** unterstützt zwei Formen des *Virtual Hostings*. Die erste Möglichkeit bezeichnet man als namenbasiertes virtuelles Hosting. Dabei wird der HTTP/1.1-Header der Clients dazu verwendet, den Rechnernamen zu bestimmen. Dadurch wird es möglich, mehrere Domains unter der gleichen IP-Adresse zu betreiben.

Damit der **Apache** namenbasierte virtuelle Domains verwalten kann, fügen Sie die folgende Zeile in `httpd.conf` ein:

```
NameVirtualHost *
```

Wenn Ihr Webserver `www.domain.tld` heißt und Sie die virtuelle Domain `www.someotherdomain.tld` einrichten wollen, ergänzen Sie `httpd.conf` um folgende Einträge:

```
<VirtualHost *>
ServerName www.domain.tld
DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
ServerName www.someotherdomain.tld
DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Ersetzen Sie dabei die Adressen sowie den Pfad zu den Dokumenten durch Ihre eigenen Einstellungen.

Ausführliche Informationen zum Einrichten von virtuellen Domains finden Sie in der offiziellen **Apache**-Dokumentation unter http://httpd.apache.org/docs/vhosts/.

30.7.5. Häufig verwendete Apache-Module

Es gibt viele verschiedene **Apache**-Module, die den Server um zusätzliche Funktionen erweitern. Die FreeBSD-Ports-Sammlung ermöglicht es Ihnen, den **Apache** gemeinsam mit einigen der beliebtesten Zusatzmodule zu installieren.

30.7.5.1. mod_ssl

Das Modul **mod_ssl** verwendet die OpenSSL-Bibliothek, um, unter Nutzung der Protokolle Secure Sockets Layer (SSL v2/v3) sowie Transport Layer Security (TLS v1) starke Verschlüsselung zu ermöglichen. Durch dieses Modul können Sie ein signiertes Zertifikat von einer Zertifizierungsstelle anfordern, damit Sie einen sicheren Webserver unter FreeBSD betreiben können.

Wenn Sie den **Apache** 1.3.X noch nicht installiert haben, können Sie über den Port `www/apache13-modssl` eine **Apache**-Version installieren, in die **mod_ssl** als Modul einkompiliert wurde. Bevorzugen Sie den **Apache** 2.X, installieren Sie stattdessen den Port `www/apache22`, bei dem die SSL-Unterstützung bereits in der Voreinstellung aktiviert ist.

30.7.5.2. Skriptsprachen

Für die wichtigsten Skriptsprachen existieren Module, die es erlauben, **Apache**-Module nahezu vollständig in einer Skriptsprache zu programmieren. Derartige Module dienen oft dazu, einen Sprach-Interpreter in den Webserver einzubetten. Dadurch wird ein zusätzlicher externer Interpreter überflüssig, was die Startzeit von dynamischen Internetseiten deutlich verringert.

30.7.6. Dynamische Webseiten

In den vergangenen Jahren haben immer mehr Unternehmen das Internet als Mittel für die Steigerung ihrer Einnahmen sowie für die Erhöhung ihrer Reichweite entdeckt. Dadurch stieg auch die Nachfrage nach interaktiven Internetinhalten. Neben einigen Unternehmen, darunter Microsoft, die dafür proprietäre Produkte entwickelt haben, hat auch die Open Source Community auf diesen Umstand reagiert und unter anderem mit Django, Ruby on Rails, **mod_perl**, und **mod_php** Möglichkeiten zur Generierung dynamischer Internetseiten geschaffen.

30.7.6.1. Django

Bei *Django* handelt es sich um ein unter der BSD-Lizenz verfügbares Framework zur schnellen Erstellung von mächtigen Internet-Applikationen. Es beinhaltet einen objekt-relationalen Mapper (wodurch Datentypen als Phyton-Objekte entwickelt werden können) sowie eine API für den dynamischen Datenbankzugriff auf diese Objekte, ohne dass Entwickler jemals SQL-Code schreiben müssen. Zusätzlich existiert ein umfangreiches Template-System, wodurch die Programmlogik von der HTML-Präsentation getrennt werden kann.

Django setzt das Modul **mod_python**, den **Apache**-Webserver sowie eine SQL-Datenbank voraus. Für FreeBSD gibt es einen Port, der alle Abhängigkeiten mit sinnvollen Optionen konfiguriert und installiert.

Beispiel 30-3. Django mit Apache2, mod_python3, und PostgreSQL installieren

```
# cd /usr/ports/www/py-django; make all install clean -DWITH_MOD_PYTHON3 -DWITH_POSTGRESQL
```

Nachdem Django (sowie die abhängigen Pakete) installiert ist, müssen Sie ein Projektverzeichnis erstellen. Danach konfigurieren Sie Apache so, dass der eingebettete Python-Interpreter spezifische URLs Ihrer Seiten aufruft.

Beispiel 30-4. Apache-Konfiguration für Django/mod_python

Sie müssen die Apache-Konfigurationsdatei `httpd.conf` anpassen, damit Apache Anfragen für bestimmte URLs an Ihre Internet-Applikation übergibt:

```
<Location "/">
    SetHandler python-program
    PythonPath "['/dir/to/your/django/packages/'] + sys.path"
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE mysite.settings
    PythonAutoReload On
    PythonDebug On
</Location>
```

30.7.6.2. Ruby on Rails

Bei *Ruby on Rails* handelt es sich um ein weiteres, als Open Source verfügbares Webframework. Es bietet einen kompletten Entwicklungsstack und erlaubt es Webentwicklern, umfangreiche und mächtige Applikationen in kurzer Zeit zu programmieren. Das Framework kann über die Ports-Sammlung installiert werden.

```
# cd /usr/ports/www/rubygem-rails; make all install clean
```

30.7.6.3. mod_perl

Die Kombination **Apache**/Perl vereinigt die Vorteile der Programmiersprache Perl und des **Apache** HTTP-Servers. Durch das Modul **mod_perl** ist es möglich, vollständig in Perl geschriebene **Apache**-Module zu erzeugen. Da der Perl-Interpreter in den Server eingebettet wird, müssen Sie weder einen externen Interpreter noch Perl zusätzlich aufrufen.

mod_perl ist in verschiedenen Versionen erhältlich. Bevor Sie **mod_perl** einsetzen,denken Sie bitte daran, dass **mod_perl** 1.0 nur mit **Apache** 1.3 und **mod_perl** 2.0 nur mit **Apache** 2.*X* zusammenarbeitet. **mod_perl** 1.0 kann über den Port `www/mod_perl`, eine statisch kompilierte Version hingegen über den Port `www/apache13-modperl` installiert werden. Für die Installation von **mod_perl** 2.0 schließlich verwenden Sie den Port `www/mod_perl2`.

30.7.6.4. mod_php

Geschrieben von Tom Rhodes.

Bei PHP, dem “Hypertext Preprocessor”, handelt es sich um eine vielseitig verwendbare Skriptsprache, die besonders für die Internetprogrammierung geeignet ist. PHP kann in HTML eingebettet werden und ähnelt von der Syntax her Sprachen wie C, Java und Perl. Das Hauptanliegen von PHP ist es, Internetprogrammierern die rasche Erstellung von dynamisch erzeugten Internetseiten zu ermöglichen.

Damit Ihr System PHP5 unterstützt, müssen Sie als Erstes den **Apache** Webserver über den Port `lang/php5` installieren.

Wenn Sie den Port `lang/php5` das erste Mal installieren, werden die verfügbaren Optionen (`OPTIONS`) automatisch angezeigt. Erscheint das Konfigurationsmenü bei Ihnen nicht, so liegt dies daran, dass Sie den Port `lang/php5` schon einmal auf Ihrem System installiert hatten. Es ist aber jederzeit möglich, dieses Menü aus dem Ports-Verzeichnis heraus über folgenden Befehl erneut aufzurufen:

```
# make config
```

In diesem Konfigurationsmenü müssen Sie die Option `APACHE` auswählen, damit **mod_php5** als ein vom **Apache**-Webserver ladbares Modul gebaut wird.

> **Anmerkung:** Viele Seiten verwenden nach wie vor (beispielsweise wegen der benötigten Kompatibilität zu bereits vorhandenen Web-Applikationen) PHP4. Ist dies bei Ihnen der Fall, so müssen Sie statt **mod_php5** **mod_php4** über den Port `lang/php4` installieren. Der Port `lang/php4` unterstützt viele der Konfigurations- und Laufzeitoptionen von `lang/php5`.

Dieser Port installiert und konfiguriert die Module, die für die Unterstützung von dynamischen PHP-Anwendungen benötigt werden. Stellen Sie danach sicher, dass Ihre `/usr/local/etc/apache/httpd.conf` die folgenden Abschnitte enthält:

```
LoadModule php5_module        libexec/apache/libphp5.so

AddModule mod_php5.c
    <IfModule mod_php5.c>
        DirectoryIndex index.php index.html
    </IfModule>
    <IfModule mod_php5.c>
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    </IfModule>
```

Nachdem dies erledigt ist, rufen Sie `apachectl` auf, um das PHP-Modul zu laden:

```
# apachectl graceful
```

Bei künftigen Upgrades von PHP wird `make config` nicht mehr benötigt, da die von Ihnen ursprünglich ausgewählten Optionen (`OPTIONS`) vom FreeBSD-Ports-Framework automatisch gespeichert werden.

Die PHP-Unterstützung von FreeBSD ist stark modular aufgebaut, daher verfügt eine Basisinstallation nur über wenige Funktionen. Eine Erweiterung um zusätzliche Funktionen ist allerdings sehr einfach über den Port `lang/php5-extensions` möglich. Der Port bietet Ihnen ein Auswahlmenü, über das Sie verschiedene PHP-Erweiterungen installieren können. Alternativ können Sie einzelne Erweiterungen aber weiterhin direkt über den jeweiligen Port installieren.

Um beispielsweise die Unterstützung des Datenbankservers **MySQL** in PHP5 zu aktivieren, installieren Sie den Port `databases/php5-mysql`.

Nachdem Sie eine Erweiterung installiert haben, müssen Sie den **Apache**-Server neu starten, damit die Erweiterung auch erkannt wird:

```
# apachectl graceful
```

Ab nun wird **MySQL** von **PHP** unterstützt.

30.8. FTP – File Transfer Protocol

Beigetragen von Murray Stokely.

30.8.1. Überblick

Das File Transfer Protocol (FTP) ermöglicht auf einfache Art und Weise den Dateiaustausch mit einem FTP-Server. Der FTP-Server **ftpd** ist bei FreeBSD bereits im Basisystem enthalten. Daher sind Konfiguration und Betrieb eines FTP-Servers unter FreeBSD relativ einfach.

30.8.2. Konfiguration

Der wichtigste Punkt ist hier die Entscheidung darüber, welche Benutzer auf Ihren FTP-Server zugreifen dürfen. Ein FreeBSD-System verfügt über diverse Systembenutzerkonten, um einzelnen Daemonen den Zugriff auf das System zu ermöglichen. Anonyme Benutzer sollten sich allerdings nicht über diese Benutzerkonten anmelden dürfen. Die Datei `/etc/ftpusers` enthält alle Benutzer, die vom FTP-Zugriff ausgeschlossen sind. In der Voreinstellung gilt dies auch die gerade erwähnten Systembenutzerkonten. Sie können über diese Datei weitere Benutzer vom FTP-Zugriff ausschließen.

Sie können den Zugriff für einige Benutzer einschränken, ohne FTP komplett zu verbieten. Dazu passen Sie `/etc/ftpchroot` entsprechend an. Diese Datei enthält Benutzer und Gruppen sowie die für sie geltenden FTP-Einschränkungen und wird in ftpchroot(5) ausführlich beschrieben.

Wenn Sie einen anonymen FTP-Zugriff auf Ihren Server ermöglichen wollen, müssen Sie den Benutzer `ftp` auf Ihrem FreeBSD-System anlegen. Danach können sich Benutzer mit dem Benutzernamen `ftp` oder `anonymous` auf Ihrem FTP-Server anmelden. Das Passwort ist dabei beliebig (allerdings wird dazu in der Regel eine E-Mail-Adresse verwendet). Meldet sich ein anonymer Benutzer an, aktiviert der FTP-Server chroot(2), um den Zugriff auf das Heimatverzeichnis des Benutzers `ftp` zu beschränken.

Es gibt zwei Textdateien, deren Inhalt Sie bei der Anmeldung an Ihrem FTP-Server anzeigen lassen können. Der Inhalt von `/etc/ftpwelcome` wird angezeigt, bevor der Login-Prompt erscheint. Nach einer erfolgreichen Anmeldung wird der Inhalt von `/etc/ftpmotd` angezeigt. Beachten Sie aber, dass es dabei um einen Pfad relativ zur Umgebung des anzumeldenden Benutzers handelt. Bei einer anonymen Anmeldung würde also die Datei `~ftp/etc/ftpmotd` angezeigt.

Nachdem Sie den FTP-Server konfiguriert haben, müssen Sie Ihn in `/etc/inetd.conf` aktivieren. Dazu müssen Sie lediglich das Kommentarsymbol “#” am Beginn der bereits vorhandenen **ftpd**-Zeile entfernen:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

Nachdem Sie diese Änderung durchgeführt haben, müssen Sie, wie in Beispiel 30-1 beschrieben, die **inetd**-Konfiguration neu einlesen. Lesen Sie bitte Abschnitt Abschnitt 30.2.2 des Handbuchs für weitere Informationen zur Aktivierung von **inetd** auf Ihren System.

Alternativ können Sie auch nur den **ftpd**-Server starten. In diesem Fall ist es ausreichend, die entsprechende Variable in der Datei `/etc/rc.conf` zu setzen:

```
ftpd_enable="YES"
```

Nachdem Sie diese Variable gesetzt haben, wird künftig beim Systemstart nur der FTP-Server gestartet. Alternativ können Sie den Server auch manuell starten, indem Sie als Benutzer `root` den folgenden Befehl ausführen:

```
# /etc/rc.d/ftpd start
```

Danach können Sie sich auf Ihrem FTP-Server anmelden:

```
% ftp localhost
```

30.8.3. Wartung

Der **ftpd**-Daemon verwendet syslog(3), um Protokolldateien zu erstellen. In der Voreinstellung werden alle FTP betreffenden Nachrichten in die Datei `/var/log/xferlog` geschrieben. Dies lässt sich aber durch das Einfügen der folgenden Zeile in `/etc/syslog.conf` ändern:

```
ftp.info      /var/log/xferlog
```

Beachten Sie, dass mit dem Betrieb eines anonymen FTP-Servers verschiedene Sicherheitsrisiken verbunden sind. Problematisch ist hier vor allem die Erlaubnis zum anonymen Upload von Dateien. Dadurch könnte Ihr Server zur Verbreitung von illegaler oder nicht lizensierter Software oder noch Schlimmeren missbraucht werden. Wollen Sie anonyme Uploads dennoch erlauben, sollten Sie die Zugriffsrechte so setzen, dass solche Dateien erst nach Ihrer Zustimmung von anderen Benutzern heruntergeladen werden können.

30.9. Mit Samba einen Datei- und Druckserver für Microsoft Windows-Clients einrichten

Beigetragen von Murray Stokely.

30.9.1. Überblick

Samba ist ein beliebtes Open Source-Softwarepaket, das es Ihnen ermöglicht, einen Datei- und Druckserver für Microsoft Windows-Clients einzurichten. Clients können sich dadurch mit einem FreeBSD-System verbinden und dessen Speicherplatz oder dessen Drucker verwenden. Dies genauso, als wenn es sich um lokale Drucker oder Festplatten handeln würde.

Samba sollte als Softwarepaket auf Ihren Installationsmedien vorhanden sein. Wenn Sie **Samba** noch nicht installiert haben, können Sie dies jederzeit über den Port oder das Paket `net/samba34` nachholen.

30.9.2. Konfiguration

Die Standardkonfigurationsdatei von **Samba** heißt `/usr/local/share/examples/samba34/smb.conf.default`. Diese Datei muss nach `/usr/local/etc/smb.conf` kopiert und angepasst werden, bevor **Samba** verwendet werden kann.

Die Datei `smb.conf` enthält Laufzeitinformationen für **Samba**, beispielsweise Druckerdefinitionen oder *filesystem shares*, also Bereiche des Dateisystems, die Sie mit Windows-Clients teilen wollen. Die Konfiguration der Datei `smb.conf` erfolgt webbasiert über das im **Samba**-Paket enthaltene Programm **swat**.

30.9.2.1. Das Samba Web Administration Tool (SWAT) verwenden

Das *Samba Web Administration Tool* (SWAT) wird als Daemon von **inetd** aktiviert. Daher müssen Sie den Kommentar vor der folgenden Zeile in `/etc/inetd.conf` entfernen, bevor Sie **swat** zur Konfiguration von **Samba** verwenden können:

```
swat   stream  tcp     nowait/400      root    /usr/local/sbin/swat    swat
```

Wie bereits in Beispiel 30-1 beschrieben, müssen Sie die **inetd**-Konfiguration neu einlesen, nachdem Sie diese Änderung durchgeführt haben.

Nachdem **swat** in der Datei `inetd.conf` aktiviert wurde, rufen Sie in Ihrem Internetbrowser die Adresse http://localhost:901 auf und melden sich mit dem `root`-Benutzerkonto an.

Nachdem Sie sich erfolgreich angemeldet haben, wird die Hauptkonfigurationseite von **Samba** geladen. Sie können nun die Dokumentation lesen, oder durch einen Klick auf die Globals-Karteikarte mit der Konfiguration beginnen. Die Einstellungen, die Sie hier vornehmen können, entsprechen denen des Abschnitts `[global]` von `/usr/local/etc/smb.conf`.

30.9.2.2. Globale Einstellungen

Unabhängig davon, ob Sie **swat** verwenden, oder `/usr/local/etc/smb.conf` direkt editieren, sollten Sie zuerst folgende Einstellungen anpassen:

`workgroup`

Der NT-Domänenname oder der Arbeitsgruppenname der Rechner, die auf den Server Zugriff haben sollen.

`netbios name`

Legt den NetBIOS-Namen fest, unter dem der **Samba**-Server bekannt ist. In der Regel handelt es sich dabei um den ersten Teil des DNS-Namens des Servers.

`server string`

Legt die Beschreibung fest, die angezeigt werden soll, wenn mit `net view` oder über andere Netzwerkprogramme Informationen über den Server angefordert werden.

30.9.2.3. Samba absichern

Zwei der wichtigsten Einstellungen in `/usr/local/etc/smb.conf` betreffen das zu verwendende Sicherheitsmodell sowie das Backend-Passwortformat für die Benutzer der Samba-Clients. Folgende Optionen sind dafür verantwortlich:

`security`

Die häufigsten Optionen sind `security = share` und `security = user`. Wenn Ihre Clients Benutzernamen verwenden, die den Benutzernamen auf Ihrem FreeBSD-Rechner entsprechen, dann sollten Sie die Einstellung *user level* verwenden. Dies ist auch die Standardeinstellung. Allerdings ist es dazu erforderlich, dass sich die Clients auf Ihrem Rechner anmelden, bevor sie auf gemeinsame Ressourcen zugreifen können.

In der Einstellung *share level* müssen sich Clients nicht unter Verwendung eines gültigen Logins auf Ihrem Rechner anmelden, bevor sie auf gemeinsame Ressourcen zugreifen können. In früheren **Samba**-Versionen war dies die Standardeinstellung.

```
passdb backend
```

Samba erlaubt verschiedene Backend-Authentifizierungsmodelle. Sie können Clients durch LDAP, NIS+, eine SQL-Datenbank oder eine Passwortdatei authentifizieren. In der Voreinstellung wird `smbpasswd` verwendet. Diese Methode wird im folgenden Abschnitt näher beschrieben.

Wenn Sie `smbpasswd` verwenden, müssen Sie die Datei `/usr/local/etc/samba/smbpasswd` erzeugen, damit **Samba** in der Lage ist, Clients zu authentifizieren. Wenn Sie auf Ihrem UNIX-Rechner vorhandenen Benutzern den Zugriff von einem Windows-Client aus ermöglichen wollen, verwenden Sie den folgenden Befehl:

```
# smbpasswd -a username
```

Anmerkung: Als Backend wird inzwischen `tdbsam` empfohlen. Mit dem folgenden Befehl legen Sie neue Benutzerkonten an:

```
# pdbedit -a -u username
```

Ausführliche Informationen zur Konfiguration von **Samba** finden Sie im Official Samba HOWTO (http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/). Sie sollten aber bereits nach dem Lesen dieses Abschnitts in der Lage sein, **Samba** zu starten.

30.9.3. Samba starten

Der Port `net/samba34` legt ein neues Startskript an, mit dem **Samba** gesteuert (also etwa gestartet oder beendet) werden kann. Um dieses Skript zu aktivieren, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
samba_enable="YES"
```

Alternativ können Sie auch die folgenden beiden Einträge verwenden:

```
nmbd_enable="YES"
```

```
smbd_enable="YES"
```

Anmerkung: Durch diese Einträge wird **Samba** beim Systemstart automatisch aktiviert.

Danach können Sie **Samba** jederzeit durch folgenden Befehl starten:

```
# /usr/local/etc/rc.d/samba start
Starting SAMBA: removing stale tdbs :
Starting nmbd.
Starting smbd.
```

Weitere Informationen zu den rc-Startskripten finden Sie im Abschnitt 12.7 des Handbuchs.

Samba verwendet drei Daemonen. Beachten Sie, dass sowohl **nmbd** als auch **smbd** durch das Skript `samba` gestartet werden. Wenn Sie die *winbind name resolution services* in `smb.conf` aktiviert haben, wird zusätzlich der **winbindd**-Daemon gestartet.

Sie können **Samba** jederzeit durch den folgenden Befehl beenden:

```
# /usr/local/etc/rc.d/samba stop
```

Samba ist ein komplexes Softwarepaket mit umfassenden Funktionen, die eine weitreichende Integration von Microsoft Windows-Netzwerken ermöglichen. Für eine Beschreibung dieser Zusatzfunktionen sollten Sie sich auf http://www.samba.org umsehen.

30.10. Die Uhrzeit mit NTP synchronisieren

Beigetragen von Tom Hukins.

30.10.1. Überblick

Da die interne Uhrzeit eines Computers nie ganz exakt ist, wurde mit NTP (*Network Time Protocol*) eine Möglichkeit geschaffen, die exakte Uhrzeit zu ermitteln und festzulegen.

Viele Internetdienste sind von einer exakten Uhrzeit abhängig. Ein Webserver könnte beispielsweise die Anforderung erhalten, eine Datei zu versenden, wenn sich diese in einer bestimmten Zeitspanne geändert hat. In einem lokalen Netzwerk ist es unbedingt notwendig, dass Rechner, die Dateien von einem gemeinsamen Dateiserver beziehen, ihre Uhrzeit synchronisieren, damit die Zeitstempel der Dateien konstistent bleiben. Dienste wie cron(8) führen Befehle zu einem bestimmten Zeitpunkt aus. Ist die Uhrzeit nicht korrekt, kann dies zu Problemen führen.

FreeBSD verwendet den ntpd(8)- NTP-Server, um die genaue Uhrzeit von anderen NTP-Servern abzufragen, die eigene Systemzeit zu setzen, oder um diese anderen Rechnern anzubieten.

30.10.2. Einen passenden NTP-Server auswählen

Um die Uhrzeit zu synchronisieren, müssen Sie sich mit einem NTP-Server verbinden. Ihr Netzwerkadministrator oder Ihr Internetprovider haben vielleicht schon einen NTP-Server eingerichtet. Lesen Sie deren Dokumentation, um dies zu überprüfen. Es gibt im Internet eine Liste mit frei zugänglichen NTP-Servern (http://support.ntp.org/bin/view/Servers/WebHome), aus der Sie sich einen in Ihrer Nähe gelegenen Server auswählen können. Beachten Sie aber auf jeden Fall die Nutzungsbedingungen des entsprechenden Servers, und fragen Sie um Erlaubnis, wenn dies nötig ist.

Die Auswahl von mehreren NTP-Servern kann sinnvoll sein, wenn ein Server ausfällt oder falsche Zeiten liefert. ntpd(8) verwendet die Antworten anderer Server, um zuverlässige Server zu bestimmen, die dann bevorzugt abgefragt werden.

30.10.3. NTP unter FreeBSD einrichten

30.10.3.1. NTP aktivieren

Wenn Sie Ihre Uhrzeit nur beim Systemstart synchronisieren wollen, können Sie ntpdate(8) verwenden. Für Desktoprechner, die regelmäßig neu gestartet werden und keine ständige Synchronisation benötigen, ist dies akzeptabel. In allen anderen Fällen sollten Sie jedoch ntpd(8) verwenden.

Die Ausführung von ntpdate(8) während des Systemstarts ist aber auch für Rechner, die ntpd(8) verwenden, sinnvoll. ntpd(8) passt die Systemzeit nur bei größeren Abweichungen an, während ntpdate(8) die Zeit immer synchronisiert, egal wie groß die Differenz zwischen Systemzeit und korrekter Zeit ist.

Um ntpdate(8) beim Systemstart zu aktivieren, fügen Sie den Eintrag `ntpdate_enable="YES"` in `/etc/rc.conf` ein. Außerdem müssen Sie alle Server, mit denen Sie sich synchronisieren wollen, sowie alle an ntpdate(8) zu übergebenden Optionen in den `ntpdate_flags` angeben.

30.10.3.2. NTP einrichten

Die Konfiguration von NTP erfolgt über die Datei `/etc/ntp.conf`, und wird in der Hilfeseite ntp.conf(5) beschrieben. Dazu ein einfaches Beispiel:

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

Die Option `server` legt die zu verwendenden Server fest, wobei jeder Server in einer eigenen Zeile steht. Wenn ein Server mit der Option `prefer` versehen ist, wie dies hier bei `ntplocal.example.com` der Fall ist, wird dieser Server bevorzugt verwendet. Eine Antwort von einem bevorzugten Server wird nur dann verworfen, wenn sie signifikant von denen anderer Server abweicht, ansonsten wird sie ohne Abfrage weiterer Server verwendet. Die Option `prefer` wird gewöhnlich nur für sehr zuverlässige und genaue Server verwendet, die über eine spezielle Hardware zur Zeitüberwachung verfügen.

Die Option `driftfile` legt fest, in welcher Datei die Abweichungen der Systemuhr protokolliert werden. ntpd(8) verwendet diese Datei, um die Systemzeit automatisch anzupassen, selbst wenn kurzzeitig kein NTP-Server zur Synchronisation verfügbar ist.

Weiterhin legt die Option `driftfile` fest, wo Informationen über frühere Antworten des von Ihnen verwendeten NTP-Servers gespeichert werden sollen. Diese Datei enthält NTP-interne Informationen, sie sollte daher von anderen Prozessen nicht verändert werden.

30.10.3.3. Den Zugang zu Ihrem NTP-Server beschränken

In der Voreinstellung ist Ihr NTP-Server für alle Rechner im Internet erreichbar. Über die Option `restrict` in der Datei `/etc/ntp.conf` können Sie den Zugang zu Ihrem Server beschränken.

Wenn Sie alle Rechner vom Zugriff auf Ihren NTP-Server ausschließen wollen, fügen Sie folgende Zeile in `/etc/ntp.conf` ein:

```
restrict default ignore
```

> **Anmerkung:** Durch diesen Eintrag verhindern Sie den Zugriff Ihres Servers auf alle auf Ihrem System konfigurierten Server. Müssen Sie Ihren NTP-Server mit einem externen NTP-Server synchronisieren, müssen Sie dies daher dezidiert zulassen. Lesen Sie in diesem Fall die Manualpage ntp.conf(5).

Wenn Sie nur Rechnern Ihres eigenen Netzwerks die Synchronisation mit Ihrem NTP-Server erlauben, gleichzeitig aber verhindern wollen, dass diese den NTP-Server konfigurieren oder als Server für andere Rechner dienen können, fügen Sie folgende Zeile ein:

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

Bei `192.168.1.0` handelt es sich um einen Rechner Ihres Netzwerks. `255.255.255.0` ist die Netzmaske Ihres Netzwerks.

`/etc/ntp.conf` kann verschiedene `restrict`-Optionen enthalten. Weiteres erfahren Sie im Abschnitt `Access Control Support` der Hilfeseite ntp.conf(5).

30.10.4. Den NTP-Server starten

Damit der NTP-Server beim Systemstart automatisch gestartet wird, fügen Sie den Eintrag `ntpd_enable="YES"` in `/etc/rc.conf` ein. Wenn Sie weitere Argumente an ntpd(8) übergeben wollen, passen Sie die Option `ntpd_flags` in der Datei `/etc/rc.conf` entsprechend an.

Um den NTP-Server ohne einen Systemneustart zu starten, rufen Sie `ntpd` mit den unter `ntpd_flags` in `/etc/rc.conf` festgelegten Parametern auf. Hierzu ein Beispiel:

```
# ntpd -p /var/run/ntpd.pid
```

30.10.5. ntpd mit einer Einwahlverbindung verwenden

ntpd(8) benötigt keine ständige Internetverbindung. Wenn Sie sich ins Internet einwählen, ist es sinnvoll, zu verhindern, dass NTP-Verkehr eine Verbindung aufbauen oder aufrechterhalten kann. Wenn Sie user-PPP verwenden, können Sie dies in den `filter`-Direktiven von `/etc/ppp/ppp.conf` festlegen. Sehen Sie sich dazu das folgende Beispiel ein:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Weitere Informationen finden Sie im Abschnitt `PACKET FILTERING` von ppp(8) sowie in den Beispielen unter `/usr/share/examples/ppp/`.

> **Anmerkung:** Einige Internetprovider blockieren Ports mit niedrigen Nummern. In solchen Fällen funktioniert NTP leider nicht, da Antworten eines NTP-Servers Ihren Rechner nicht erreichen werden.

30.10.6. Weitere Informationen

Weiterführende Dokumentation (im HTML-Format) zum NTP-Server finden Sie unter `/usr/share/doc/ntp/`.

30.11. Protokollierung von anderen Hosts mittels `syslogd`

Beigetragen von Tom Rhodes. Übersetzt von Benedict Reuschling.

Die Interaktion mit Systemprotokollen ist ein wichtiger Aspekt, sowohl was Sicherheit als auch Systemadministration anbelangt. Überwachen der Protokolldateien von mehreren Hosts kann sehr unhandlich werden, wenn diese Hosts über mittlere oder grosse Netze verteilt sind oder wenn sie Teile von unterschiedlichen Netzwerken sind. In diesen Fällen macht die Konfiguration der Protokollierung von anderen Hosts diesen Prozess wesentlich komfortabler.

Die zentralisierte Protokollierung auf einen bestimmten Protokollierungshost kann manche der administrativen Belastungen der Protokolldateiadministration reduzieren. Protokolldateiaggregation, -zusammenführung und -rotation kann an einer zentralen Stelle mit den FreeBSD-eigenen Werkzeugen wie syslogd(8) und newsyslog(8) konfiguriert werden. In der folgenden Beispielkonfiguration sammelt Host `A`, genannt `logserv.example.com`, Protokollinformationen für das lokale Netzwerk. Host `B`, genannt `logclient.example.com` wird seine Protokollinformationen an den Server weiterleiten. In realen Konfigurationen benötigen beide Hosts passende Vorwärts- und Umkehr-Einträge im DNS oder in `/etc/hosts`. Andernfalls werden die Daten vom Server abgelehnt.

30.11.1. Konfiguration des Protokollierungs-Servers

Protokollierungs-Server sind Maschinen, die konfiguriert sind, Protokollinformationen von anderen Hosts zu akzeptieren. In den meisten Fällen wird dies zur Vereinfachung der Konfiguration eingesetzt, in anderen Fällen ist es einfach nur ein Schritt in eine bessere Verwaltung. Was auch immer die Gründe sind, ein paar Anforderungen müssen vorher erfüllt sein.

Ein richtig konfigurierter Protokollierungs-Server muss minimal die folgenden Anforderungen erfüllen:

- Das Regelwerk der Firewall muss UDP auf Port 514 sowohl auf Client- als auch auf Serverseite erlauben;
- syslogd wurde so konfiguriert, dass es Nachrichten von anderen Clientrechnern akzeptiert;
- Der syslogd-Server und alle Clientrechner müssen gültige Einträge für sowohl Vorwärts- als auch Umkehr-DNS besitzen, oder in `/etc/hosts` korrekt eingetragen sein.

Um den Protokollierungs-Server zu konfigurieren, muss der Client in `/etc/syslog.conf` eingetragen sein und der Verbindungsweg der Protokollierung muss spezifiziert sein:

```
+logclient.example.com
*.*     /var/log/logclient.log
```

Anmerkung: Weitere Informationen zu den verschiedenen unterstützten und verfügbaren *Verbindungswegen* finden sich in der Manualpage syslog.conf(5).

Einmal hinzugefügt, werden alle Nachrichten über den `Verbindungsweg` in die zuvor angegebene Datei, `/var/log/logclient.log` protokolliert.

Der Server benötigt ausserdem die folgenden Zeilen in der `/etc/rc.conf`:

```
syslogd_enable="YES"
syslogd_flags="-a logclient.example.com -v -v"
```

Die erste Option aktiviert den `syslogd`-Dienst während des Systemstarts und die zweite Option erlaubt es, Daten von dem spezifizierten Client auf diesem Server zu akzeptieren. Die Verwendung von `-v -v` im letzten Teil erhöht die Anzahl von Protokollnachrichten. Dies ist sehr hilfreich für die Feineinstellung der Verbindungspfade, da Administratoren auf diese Weise erkennen, welche Arten von Nachrichten unter welchen Einstellungen protokolliert werden.

Mehrere `-a`-Optionen können angegeben werden, um die Protokollierung von mehreren Clients zu erlauben. IP-Adressen und ganze Netzblöcke können ebenfalls spezifiziert werden. Lesen Sie dazu die syslog(3)-Manualpage, um eine vollständige Liste von möglichen Optionen zu erhalten.

Zum Schluss muss noch die Protokolldatei erstellt werden. Auf welche Weise dies geschieht ist nicht wichtig, aber in den meisten Fällen funktioniert touch(1) grossartig, wie hier dargestellt:

```
# touch /var/log/logclient.log
```

Zu diesem Zeitpunkt sollte der `syslogd`-Dienst neu gestartet und überprüft werden:

```
# /etc/rc.d/syslogd restart
# pgrep syslog
```

Wenn eine PID zurückgegeben wird, wurde der Server erfolgreich neu gestartet und die Clientkonfiguration kann beginnen. Wenn der Server nicht neu gestartet wurde, suchen Sie im `/var/log/messages`-Protokoll nach eventuellen Fehlermeldungen.

30.11.2. Konfiguration des Protokollierungs-Clients

Ein Protokollierungs-Client ist eine Maschine, die Protokollinformationen an einen Protokollierungs-Server sendet, zusätzlich zu ihren lokalen Kopien.

Ähnlich wie Protokollierungs-Server müssen Clients auch ein paar minimale Anforderungen erfüllen:

- syslogd(8) muss so konfiguriert sein, dass es Nachrichten eines bestimmten Typs an einen Protokollierungs-Server schickt, welcher diese akzeptieren muss;
- Die Firewall muss UDP-Pakete durch Port 514 erlauben;
- Sowohl Vorwärts- als auch Umkehr-DNS muss konfiguriert sein oder es müssen passende Einträge in `/etc/hosts` vorhanden sein.

Die Clientkonfiguration ist ein bisschen entspannter, verglichen mit der des Servers. Der Clientrechner muss ebenfalls die folgenden Einträge in der `/etc/rc.conf` besitzen:

```
syslogd_enable="YES"
syslogd_flags="-s -v -v"
```

Wie zuvor aktivieren diese Einträge den `syslogd`-Dienst während des Systemstarts und erhöhen die Anzahl der Protokollnachrichten. Die Option `-s` verhindert, dass dieser Client Protokolle von anderen Hosts akzeptiert.

Verbindungspfade beschreiben den Systemteil, für den eine Nachricht generiert wird. Beispielsweise sind ftp und ipfw beides Verbindungspfade. Wenn Protokollnachrichten für diese beiden Dienste generiert werden, sind diese beiden Werkzeuge normalerweise in jeder Protokollnachricht enthalten. Verbindungspfade sind mit einer Priorität oder Stufe verbunden, die dazu verwendet wird, zu markieren, wie wichtig eine Nachricht im Protokoll ist. Die Häftigste ist `warning` und `info`. Bitte lesen Sie die syslog(3) Manualpage, um eine komplette Liste der verfügbaren Verbindungspfade und Prioritäten zu erhalten.

Der Protokollierungs-Server muss in der `/etc/syslog.conf` des Clients eingetragen sein. In diesem Beispiel wird das @-Symbol benutzt, um Protokolldaten an einen anderen Server zu senden. Der Eintrag sieht wie folgt aus:

```
*.*                @logserv.example.com
```

Einmal hinzugefügt, muss `syslogd` neu gestartet werden, damit diese Änderungen wirksam werden:

```
# /etc/rc.d/syslogd restart
```

Um zu testen, ob Protokollnachrichten über das Netzwerk gesendet werden, kann logger(1) auf dem Client benutzt werden, um eine Nachricht an `syslogd` zu schicken:

```
# logger "Test message from logclient"
```

Diese Nachricht sollte jetzt sowohl in `/var/log/messages` auf dem Client, als auch in `/var/log/logclient.log` auf dem Server vorhanden sein.

30.11.3. Fehlerbehebung beim Protokollierungs-Server

In bestimmten Fällen ist die Fehlerbehebung notwendig, wenn Nachrichten nicht auf dem Protokollierungs-Server empfangen werden. Es gibt mehrere Gründe dafür, jedoch treten am häufigsten Probleme bei der Netzwerkverbindung und beim DNS auf. Um diese Fälle zu überprüfen, stellen Sie sicher, dass beide Hosts in der Lage sind, sich gegenseitig über den Hostnamen zu erreichen, der in `/etc/rc.conf` angegeben ist. Wenn das funktioniert, ist möglicherweise eine Änderung der `syslogd_flags`-Option in `/etc/rc.conf` notwendig.

Im folgenden Beispiel ist `/var/log/logclient.log` leer und die `/var/log/messages`-Dateien enthalten keine Gründe für den Fehler. Um die Fehlerausgabe zu erhöhen, ändern Sie die `syslogd_flags`-Option so, dass diese wie in dem folgenden Beispiel aussieht und initiieren Sie dann einen Neustart:

```
syslogd_flags="-d -a logclien.example.com -v -v"
```

```
# /etc/rc.d/syslogd restart
```

Fehlerausgabedaten ähnlich der Folgenden werden sofort nach dem Neustart auf dem Bildschirm erscheinen:

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is /boot/kernel/k
Logging to FILE /var/log/messages
syslogd: kernel boot file is /boot/kernel/kernel
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
rejected in rule 0 due to name mismatch.
```

Es scheint klar zu sein, dass die Nachrichten aufgrund eines fehlerhaften Namens abgewiesen werden. Nach genauer Untersuchung der Konfiguration, kommt ein Tippfehler in der folgenden Zeile der `/etc/rc.conf` als Fehler in Betracht:

```
syslogd_flags="-d -a logclien.example.com -v -v"
```

Die Zeile sollte `logclient` und nicht `logclien` enthalten. Nachdem die entsprechenden Veränderungen gemacht wurden, ist ein Neustart fällig, mit den entsprechenden Ergebnissen:

```
# /etc/rc.d/syslogd restart
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is /boot/kernel/k
syslogd: kernel boot file is /boot/kernel/kernel
logmsg: pri 166, flags 17, from logserv.example.com,
msg Dec 10 20:55:02 <syslog.err> logserv.example.com syslogd: exiting on signal 2
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
accepted in rule 0.
logmsg: pri 15, flags 0, from logclient.example.com, msg Dec 11 02:01:28 trhodes: Test message 2
Logging to FILE /var/log/logclient.log
Logging to FILE /var/log/messages
```

Zu diesem Zeitpunkt werden die Nachrichten korrekt empfangen und in die richtige Datei geschrieben.

30.11.4. Sicherheitsbedenken

Wie mit jedem Netzwerkdienst, müssen Sicherheitsanforderungen in Betracht gezogen werden, bevor diese Konfiguration umgesetzt wird. Manchmal enthalten Protokolldateien sensitive Daten über aktivierte Dienste auf dem lokalen Rechner, Benutzerkonten und Konfigurationsdaten. Daten, die vom Client an den Server geschickt werden, sind weder verschlüsselt noch mit einem Passwort geschützt. Wenn ein Bedarf für Verschlüsselung besteht, ist es möglich, `security/stunnel` zu verwenden, welches die Daten über einen verschlüsselten Tunnel versendet.

Lokale Sicherheit ist ebenfalls ein Thema. Protokolldateien sind während der Verwendung oder nach ihrer Rotation nicht verschlüsselt. Lokale Benutzer versuchen vielleicht, auf diese Dateien zuzugreifen, um zusätzliche Einsichten in die Systemkonfiguration zu erlangen. In diesen Fällen ist es absolut notwendig, die richtigen Berechtigungen auf diesen Dateien zu setzen. Das newsyslog(8)-Werkzeug unterstützt das Setzen von Berechtigungen auf gerade erstellte oder rotierte Protokolldateien. Protokolldateien mit Zugriffsmodus `600` sollten verhindern, dass lokale Benutzer darin herumschnüffeln.

Kapitel 31. Firewalls

Beigetragen von Joseph J. Barbish. Nach SGML konvertiert und aktualisiert von Brad Davis. Übersetzt von Michael Bunzel, Johann Kois und Benjamin Lukas.

31.1. Einführung

Firewalls ermöglichen es, den ein- und ausgehenden Netzwerkverkehr Ihres Systems zu filtern. Dazu verwendet eine Firewall eine oder mehrere Gruppen von "Regeln", um ankommende Netzwerkpakete zu untersuchen und entweder durchzulassen oder zu blockieren. Die Regeln einer Firewall untersuchen charakteristische Eigenschaften von Datenpaketen, darunter den Protokolltyp, die Quell- und Zieladresse sowie den Quell- und Zielport.

Firewalls können die Sicherheit eines Rechners oder eines Netzwerks erhöhen, indem sie folgende Aufgaben übernehmen:

- Den Schutz der Anwendungen, Dienste und Rechner Ihres internen Netzwerks vor unerwünschtem Datenverkehr aus dem Internet.
- Die Beschränkung des Zugriffs von Rechnern des internen Netzwerk auf Rechner oder Dienste des externen Internets.
- Den Einsatz von Network Address Translation (NAT), die es Ihnen durch die Verwendung von privaten IP-Adressen ermöglicht, eine einzige gemeinsame Internetverbindung für mehrere Rechner zu nutzen (entweder über eine einzige Adresse oder über eine Gruppe von jeweils automatisch zugewiesenen öffentlichen IP-Adressen).

Nachdem Sie dieses Kapitel gelesen haben, werden Sie:

- Wissen, wie man korrekte Paketfilterregeln erstellt.
- Die Unterschiede zwischen den in FreeBSD eingebauten Firewalls kennen.
- Wissen, wie man die **PF**-Firewall von OpenBSD konfiguriert und einsetzt.
- **IPFILTER** konfigurieren und einsetzen können.
- Wissen, wie man **IPFW** konfiguriert und einsetzt.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Die grundlegenden Konzepte von FreeBSD und dem Internet verstehen.

31.2. Firewallkonzepte

Es gibt zwei grundlegende Arten, Regelgruppen für Firewalls zu erstellen: "einschließend" (*inclusive firewall*) sowie "ausschließend" (*exclusive Firewall*). Eine ausschließende Firewall lässt jeden Datenverkehr durch, der nicht durch eine Regel ausgeschlossen wurde. Eine einschließende Firewall macht das genaue Gegenteil. Sie lässt Datenverkehr nur dann durch, wenn er einer der definierten Regeln entspricht.

Eine inclusive Firewall bietet eine wesentlich bessere Kontrolle des ausgehenden Verkehrs, macht sie zur besseren Wahl für Systeme, die Services für das Internet anbieten. Sie kontrolliert auch den Verkehr vom Internet zu ihrem privaten Netzwerk. Jeder Verkehr, der keiner Regel entspricht wird geblockt und geloggt. Inclusive Firewalls sind

generell sicherer als exclusive Firewalls, da sie das Risiko, dass unerwünschter Verkehr hindurch geht, drastisch reduzieren.

> **Anmerkung:** Wenn nicht anders vermerkt, verwenden alle Konfigurationen und Beispielregelsets dieses Kapitels inclusive Firewalls.

Die Sicherheit einer Firewall kann durch den Einsatz einer “zustandsabhängigen Firewall” (*stateful firewall*) weiter erhöht werden. Dieser Typ einer Firewall überwacht alle durch die Firewall gehenden offenen Verbindungen und erlaubt nur schon bestehenden Verkehr oder Datenverkehr, der eine neue Verbindung öffnet. Der Nachteil einer zustandsabhängigen Firewall ist allerdings, dass sie anfällig für Denial of Service (DoS) -Attacken ist, wenn sehr schnell sehr viele neue Verbindungen erstellt werden. Bei den meisten Firewalls können Sie eine Kombination aus zustandsabhängigem und nicht zustandsabhängigem Verhalten verwenden, um eine für Ihre Bedürfnisse optimale Firewall einzurichten.

31.3. Firewallpakete

Das Basissystem von FreeBSD enthält bereits drei Firewallpakete: *IPFILTER* (auch als IPF bekannt), *IPFIREWALL* (auch als IPFW bezeichnet) sowie das von OpenBSD übernommene *PacketFilter* (das auch als PF bezeichnet wird). Zusätzlich verfügt FreeBSD über zwei eingebaute Pakete für das sogenannte *traffic shaping* (dabei handelt es sich die Steuerung des Bandbreitenverbrauchs): altq(4) sowie dummynet(4). Dummynet steht traditionell in enger Verbindung mit IPFW, während ALTQ gemeinsam mit PF eingesetzt wird. Traffic Shaping für IPFILTER ist derzeit mit IPFILTER für NAT sowie Filterung und mit IPFW und dummynet(4) *oder* durch die Kombination von PF mit ALTQ möglich. Gemeinsam ist allen Firewallpaketen (IPF, IPFW sowie PF), dass sie Regeln einsetzen, um den Transfer von Datenpaketen auf und von Ihrem System zu regeln. Unterschiedlich sind aber die Art und Weise, wie dies realisiert wird. Auch die für diese Regeln verwendete Syntax ist unterschiedlich.

FreeBSD überlässt es dem Anwender, das Firewallsystem zu wählen, dass seinen Anforderungen und Vorlieben am Besten entspricht. Keines der im Basissystem enthaltenen Firewallpakete wird dabei als “das beste” angesehen.

IPFILTER hat etwa den Vorteil, dass dessen zustandsabhängige Regeln relativ einfach in einer NAT-Umgebung implementiert werden können. Außerdem verfügt es über einen eigenen FTP-Proxy, der die Erstellung von sicheren Regeln für ausgehende FTP-Verbindungen vereinfacht.

Da alle Firewalls auf der Untersuchung der Werte ausgewählter Kontrollfelder von Datenpaketen basieren, ist es für die Erstellung von Firewallregeln notwendig, die Funktionsweise von TCP/IP zu verstehen. Außerdem muss man dazu wissen, was die Werte der einzelnen Kontrollfelder bedeuten und wie diese während einer Verbindung eingesetzt werden. Eine gute Erklärung dieser Thematik finden Sie unter http://www.ipprimer.com/overview.cfm.

31.4. Paket Filter (PF) von OpenBSD und ALTQ

Revised and updated by John Ferrell.

Im Juli 2003 wurde PF, die Standard-Firewall von OpenBSD, nach FreeBSD portiert und in die FreeBSD-Ports-Sammlung aufgenommen. 2004 war PF in FreeBSD 5.3 Teil des Basissystems. Bei PF handelt es sich um eine komplette, vollausgestattete Firewall, die optional auch ALTQ (Alternatives Queuing) unterstützt. ALTQ bietet Ihnen *Quality of Service* (QoS)-Bandbreitenformung.

Das OpenBSD-Projekt leistet bereits hervorragende Dokumentationsarbeit mit der PF FAQ (http://www.openbsd.org/faq/pf/). Aus diesem Grund konzentriert sich dieser Handbuchabschnitt nur auf diejenigen Besonderheiten von PF, die FreeBSD betreffen, sowie ein paar allgemeine Informationen hinsichtlich der Verwendung. Genauere Informationen zum Einsatz erhalten Sie in der PF FAQ (http://www.openbsd.org/faq/pf/).

Weitere Informationen zu PF für FreeBSD finden Sie unter http://pf4freebsd.love2party.net/.

31.4.1. Verwendung der PF-Kernelmodule

Um die PF Kernel Module zu laden, fügen Sie folgende Zeile in ihre `/etc/rc.conf` ein:

```
pf_enable="YES"
```

Danach starten Sie das Startup Script um die Module zu laden:

```
# /etc/rc.d/pf start
```

Das PF Modul wird nicht geladen, falls es die Ruleset Konfigurationsdatei nicht findet. Standardmässig befindet sich diese Datei in `/etc/pf.conf`. Falls das PF Ruleset sich an einem anderen Platz befindet, können Sie das durch Hinzufügen einer Zeile ähnlich der folgenden, in ihrer `/etc/rc.conf` ändern:

```
pf_rules="/path/to/pf.conf"
```

> **Anmerkung:** Ein Beispiel für die Datei `pf.conf` finden Sie im Verzeichnis `/usr/share/examples/pf/`.

Das PF-Modul kann auch manuell über die Kommandozeile geladen werden:

```
# kldload pf.ko
```

Protokollierungsfunktionen für PF werden durch das Modul `pflog.ko` zur Verfügung gestellt und können durch folgenden Eintrag in der `/etc/rc.conf` aktiviert werden:

```
pflog_enable="YES"
```

Danach starten Sie das Startup Script, um das Modul zu laden:

```
# /etc/rc.d/pflog start
```

Falls Sie noch weitere Features für PF benötigen, müssen Sie diese in den Kernel einbauen.

31.4.2. PF Kernel-Optionen

Es ist nicht zwingend nötig, dass Sie PF-Unterstützung in den FreeBSD-Kernel kompilieren. Sie werden dies tun müssen, um eine von PFs fortgeschritteneren Eigenschaften nutzen zu können, die nicht als Kernelmodul verfügbar ist. Genauer handelt es sich dabei um pfsync(4), ein Pseudo-Gerät, welches bestimmte Änderungen der PF-Zustandstabelle offenlegt. Es kann mit carp(4) kombiniert werden, um ausfallsichere Firewalls mit PF zu realisieren. Weitere Informationen zu CARP erhalten Sie in Abschnitt 32.13 des Handbuchs.

Die Kernelkonfigurationsoptionen von PF befinden sich in `/usr/src/sys/conf/NOTES` und sind im Folgenden wiedergegeben:

```
device pf
device pflog
device pfsync
```

Die Option `device pf` aktiviert die Unterstützung für die “Packet Filter”-Firewall (pf(4)).

Die Option `device pflog` aktiviert das optionale pflog(4)-Pseudonetzwerkgerät, das zum Protokollieren des Datenverkehrs über einen bpf(4)-Deskriptor dient. pflogd(8) ist in der Lage, diese Protokolldateien auf Ihre Platte zu speichern.

Die Option `device pfsync` aktiviert das optionale pfsync(4)-Pseudonetzwerkgerät für die Überwachung von “Statusänderungen”.

31.4.3. Verfügbare rc.conf-Optionen

Die folgenden rc.conf(5)-Einträge konfigurieren PF und pflog(4) beim Systemstart:

```
pf_enable="YES"                 # PF aktivieren (Modul, wenn nötig, aktivieren)
pf_rules="/etc/pf.conf"         # Datei mit Regeldefinitionen für pf
pf_flags=""                     # zusätzliche Parameter für den Start von pfctl
pflog_enable="YES"              # starte pflogd(8)
pflog_logfile="/var/log/pflog"  # wo soll pflogd die Protokolldatei speichern
pflog_flags=""                  # zusätzliche Parameter für den Start von pflogd
```

Wenn Sie ein lokales Netzwerk hinter dieser Firewall betreiben und Pakete für dessen Rechner weiterleiten oder NAT verwenden wollen, benötigen Sie zusätzlich die folgende Option:

```
gateway_enable="YES"            # LAN Gateway aktivieren
```

31.4.4. Filterregeln erstellen

PF liest seine konfigurierten Regeln aus pf.conf(5) (standardmässig `/etc/pf.conf`) und modifiziert, verwirft oder lässt Pakete passieren anhand der Regeln oder Definitionen, die in dieser Datei gespeichert sind. FreeBSD enthält dazu nach der Installation mehrere Beispieldateien, die in `/usr/share/examples/pf/` abgelegt sind. Für eine ausführliche Behandlung des PF-Regelwerks lesen Sie bitte die PF FAQ (http://www.openbsd.org/faq/pf/).

> **Warnung:** Beim Lesen der PF FAQ (http://www.openbsd.org/faq/pf/) wollten Sie darauf achten, dass verschiedene Versionen von FreeBSD auch unterschiedliche Versionen von PF enthalten. FreeBSD 8.*x* (und älter) FreeBSD-Versionen benutzen PF aus OpenBSD 4.1. FreeBSD 9.*x* (und neuer) benutzen hingegen PF aus OpenBSD 4.5.

Die FreeBSD packet filter mailing list (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf) ist eine erste Anlaufstelle für Fragen zur Konfiguration und dem Einsatz der PF Firewall. Vergessen Sie nicht, vorher die Mailinglistenarchive zu durchsuchen, bevor Sie dort eine Frage stellen!

31.4.5. Arbeiten mit PF

Benutzen Sie pfctl(8), um PF zu steuern. Unten finden Sie ein paar nützliche Befehle (lesen Sie auch die Manualpage zu pfctl(8), um alle verfügbaren Optionen nachzuschlagen):

Befehl	Zweck
`pfctl -e`	PF aktivieren
`pfctl -d`	PF deaktivieren
`pfctl -F all -f /etc/pf.conf`	Alle Filterregeln zurücksetzen (NAT, Filter, Zustand, Tabelle, etc.) und erneut aus der Datei `/etc/pf.conf` auslesen
`pfctl -s [ Regeln | NAT | Zustand ]`	Bericht über die Filterregeln, NAT-Regeln, oder Zustandstabellen
`pfctl -vnf /etc/pf.conf`	überprüft `/etc/pf.conf` auf Fehler, lädt aber das Regelwerk nicht neu

31.4.6. ALTQ aktivieren

ALTQ muss vor der Verwendung in den FreeBSD-Kernel kompiliert werden. Beachten Sie, dass ALTQ nicht von allen verfügbaren Netzwerkkartentreibern unterstützt wird. Sehen Sie daher zuerst in altq(4) nach, ob Ihre Netzwerkkarte diese Funktion unter Ihrer FreeBSD-Version unterstützt.

Die folgenden Kerneloptionen aktivieren ALTQ sowie alle Zusatzfunktionen:

```
options         ALTQ
options         ALTQ_CBQ        # Class Bases Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Scheduler (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
options         ALTQ_NOPCC      # Wird von SMP benötigt
```

`options ALTQ` aktiviert das ALTQ-Framework.

`options ALTQ_CBQ` aktiviert das *Class Based Queuing* (CBQ). CBQ erlaubt es, die Bandbreite einer Verbindung in verschiedene Klassen oder Warteschlangen zu unterteilen, um die Priorität von Datenpaketen basierend auf Filterregeln zu ändern.

`options ALTQ_RED` aktiviert *Random Early Detection* (RED). RED wird zur Vermeidung einer Netzwerkverstopfung verwendet. Dazu ermittelt RED die Größe der Warteschlange und vergleicht diesen Wert mit den minimalen und maximalen Grenzwerten der Warteschlange. Ist die Warteschlange größer als das erlaubte Maximum, werden alle neuen Pakete verworfen. Getreu seinem Namen verwirft RED Pakete unterschiedlicher Verbindungen nach dem Zufallsprinzip.

`options ALTQ_RIO` aktiviert *Random Early Detection In and Out*.

`options ALTQ_HFSC` aktiviert den *Hierarchical Fair Service Curve*-Paketplaner. Weitere Informationen zu HFSC finden Sie unter http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html.

`options ALTQ_PRIQ` aktiviert *Priority Queuing* (PRIQ). PRIQ lässt Verkehr einer Warteschlange mit höherer Priorität zuerst durch.

`options ALTQ_NOPCC` aktiviert die SMP Unterstützung von ALTQ. Diese Option ist nur auf SMP-System erforderlich.

31.5. Die IPFILTER-Firewall (IPF)

Geschrieben wurde IPFILTER von Darren Reed. IPFILTER ist vom Betriebssystem unabhängig: Es ist eine Open Source Anwendung, die auf die Betriebssysteme FreeBSD, NetBSD, OpenBSD, SunOS, HP/UX und Solaris portiert wurde. IPFILTER wird aktiv betreut und gepflegt. Es werden regelmäßig neue Versionen herausgegeben.

IPFILTER basiert auf einer kernelseitigen Firewall und einem NAT Mechanismus, der durch Anwenderprogramme betreut und gesteuert werden kann. Die Regeln der Firewall werden mit dem Programm ipf(8) gesetzt oder gelöscht. Für die Manipulation der NAT Regeln verwendet man ipnat(1). Mit ipfstat(8) werden Laufzeitstatistiken der kernelseitigen Anteile von IPFILTER aufgelistet. Und mit dem Programm ipmon(8) kann man die Aktionen von IPFILTER in die Protokolldateien des Systems speichern.

IPF funktionierte ursprünglich mit einer Regel-Prozess-Logik à la “die letzte Regel, die passt, entscheidet” und verwendete ausschließlich Regeln ohne feste Zustände. Inzwischen wurde die Regel-Prozess-Logik drastisch modernisiert: Es gibt eine `quick` und eine zustandsorientierte `keep-state` Option. Die offizielle Dokumentation beinhaltet leider nur die veralteten Parameter zur Regelerstellung - die neuen Funktionen werden nur als Zusatzoptionen aufgelistet, was ihre Vorteile beim Erstellen einer weit überlegenen und viel sichereren Firewall völlig untergräbt.

Die Anweisungen in diesem Kapitel basieren darauf, Regeln mit den Optionen `quick` und `keep-state` zu erstellen. Mit diesem Grundwissen wird man einen kompletten einschließenden Regelsatz erstellen können.

Für eine ausführliche Erläuterung der alten Methode zur Regelverarbeitung schauen Sie bitte auf http://www.obfuscation.org/ipf/ipf-howto.html#TOC_1 oder http://coombs.anu.edu.au/~avalon/ip-filter.html.

Antworten auf häufige Fragen finden Sie unter http://www.phildev.net/ipf/index.html.

Und ein durchsuchbares Archiv der Mailingliste zu IPFILTER gibt es unter http://marc.theaimsgroup.com/?l=ipfilter.

31.5.1. Aktivieren von IPF

FreeBSD enthält IPF in der Standardversion als ladbares Kernelmodul. Dieses Modul wird vom System automatisch geladen, wenn in der `rc.conf` der Eintrag `ipfilter_enable="YES"` angelegt wird. In dieser ursprünglichen Konfiguration ist die Protokollierung aktiv und die Option `default pass all` ("Pakete passieren lassen") als Standard gesetzt. Um die `block all` ("alles Blockieren") Option zu setzen, muss man nicht gleich einen neuen Kernel bauen - es reicht, `block all` als letzte Position des Regelsatzes aufzulisten.

31.5.2. Kernel-Optionen

Es ist nicht unbedingt notwendig, IPF durch die folgenden Optionen direkt in der Kernel einzubinden. Diese Möglichkeit der Verwendung von IPF wird hier mehr als Hintergrundwissen angeboten. Man sollte nur wissen, dass dadurch nicht mehr das Kernelmodul geladen wird - und dementsprechend auch nicht mehr entladen werden kann.

Die Beschreibung der einzelnen Optionen von IPF für die Verwendung in der Kernelkonfiguration finden Sie auch in der Datei `/usr/src/sys/conf/NOTES`.

```
options IPFILTER
```

```
options IPFILTER_LOG
options IPFILTER_DEFAULT_BLOCK
```

`options IPFILTER` aktiviert die Verwendung der “IPFILTER” Firewall.

`options IPFILTER_LOG` aktiviert den Logging-Mechanismus. Das bedeutet, dass jedes Paket geloggt wird, auf das eine Regel passt, die das Schlüsselwort `log` enthält. Dazu wird der Pseudo—Device `ipl` verwendet.

`options IPFILTER_DEFAULT_BLOCK` ändert das Verhalten der Firewall dahingehend, dass jedes Paket, dass nicht explizit von einer `pass` Regel Zugang erhält, abgewiesen, bzw. geblockt, wird.

Diese Einstellungen werden erst aktiv, wenn der Kernel, in den sie eingebunden wurden, kompiliert, installiert und gebootet wurde.

31.5.3. Optionen in rc.conf

Um IPF während des Bootvorgangs einzubinden, braucht man lediglich die folgenden Zeilen der Datei `/etc/rc.conf` anzufügen:

```
ipfilter_enable="YES"             # Startet IPF
ipfilter_rules="/etc/ipf.rules"   # liest den Regelsatz aus einer Datei
ipmon_enable="YES"                # Startet das IP-Monitor Log
ipmon_flags="-Ds"                 # D = Als Da:mon starten
                                  # s = Protokollierung via syslog
                                  # v = Protokollierung von tcp window, ack, seq
                                  # n = Namen statt IP & port ausgeben
```

Falls sich hinter der Firewall ein lokales Netzwerk befindet, das den reservierten privaten Adressbereich verwendet, müssen die folgenden Zeilen zur Aktivierung von NAT ebenfalls in `/etc/rc.conf` eingetragen werden:

```
gateway_enable="YES"              # Aktivierung des LAN-Gateways
ipnat_enable="YES"                # Startet die ipnat Funktion
ipnat_rules="/etc/ipnat.rules"    # Liest die ipnat-Regeldefinitionen aus einer Datei
```

31.5.4. Der Befehl ipf

Mit dem Befehl ipf(8) liest man die Datei, die den Regelsatz enthält ein. Mit dem folgenden Befehl können Sie Ihre eigenen, für Ihr System maßgeschneiderten Regeln einlesen und so in einem Schritt alle Regeln der laufenden Firewall ersetzen:

```
# ipf -Fa -f /etc/ipf.rules
```

`-Fa` bedeutet, dass alle intern gespeicherten Tabellen mit Regeln gelöscht werden.

`-f` gibt die Datei an, aus der die neuen Regeln gelesen werden sollen.

Mit diesen beiden Optionen erhalten Sie die Möglichkeit, Änderungen an der Datei mit Ihrem Regelsatz vorzunehmen und gleich die Firewall mit den neuen Regeln zu bestücken, ohne den Rechner neu starten zu müssen. Da dieser Vorgang beliebig wiederholt werden kann, ist es ein sehr bequemer Weg, neue Regeln einzuarbeiten und zu testen.

Um mehr über diese und weitere Optionen von ipf(8) zu erfahren, konsultieren Sie bitte die Manpage.

ipf(8) erwartet, dass es sich bei der Datei mit dem Regelsatz um eine Standard-Textdatei handelt. Eine Datei, die ein Skript oder Variablen enthält, wird nicht verarbeitet.

Es gibt allerdings doch einen Weg, IPF Regeln mit Hilfe von Skripten und Variablen zu erstellen. Weitere Informationen dazu finden Sie unter Abschnitt 31.5.9.

31.5.5. IPFSTAT

Das normale Verhalten von ipfstat(8) ist, die Zusammenfassung der angefallenen Statistiken, die als Resultat der Anwendung von nutzerspezifischen Regeln auf ein- und ausgehende Pakete seit dem letzten Start der Firewall oder seit dem letzten Zurücksetzen der Zähler auf Null durch das Kommando `ipf -Z` angesammelt wurden, abzurufen und anzuzeigen.

Für weiterführende Informationen schauen Sie bitte auf die Manpage von ipfstat(8)!

Die Ausgabe von ipfstat(8), wenn keine Parameter übergeben wurden, sieht etwa so aus:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
input packets logged: blocked 99286 passed 0
output packets logged: blocked 0 passed 0
packets logged: input 0 output 0
log failures: input 3898 output 0
fragment state(in): kept 0 lost 0
fragment state(out): kept 0 lost 0
packet state(in): kept 169364 lost 0
packet state(out): kept 431395 lost 0
ICMP replies: 0 TCP RSTs sent: 0
Result cache hits(in): 1215208 (out): 1098963
IN Pullups succeeded: 2 failed: 0
OUT Pullups succeeded: 0 failed: 0
Fastroute successes: 0 failures: 0
TCP cksum fails(in): 0 (out): 0
Packet log flags set: (0)
```

Wenn die Option `-i` für “eingehend” oder `-o` für “ausgehend” übergeben wird, liefert das Kommando eine entsprechende Liste von Filter-Regeln, die gerade installiert sind und vom Kernel verwendet werden.

`ipfstat -in` zeigt alle aktive Regeln für eingehende Verbindungen zusammen mit ihren Nummern.

`ipfstat -on` erledigt dasselbe für die ausgehenden Verbindungen.

Die Ausgabe sieht in etwa folgendermaßen aus:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

`ipfstat -ih` zeigt die Tabelle der aktiven Regeln für eingehende Verbindungen zusammen mit der Anzahl, wie oft jeder einzelnen Regel entsprochen wurde.

`ipfstat -oh` zeigt das Gleiche für die ausgehenden Verbindungen.

Hier wird die Ausgabe so oder so ähnlich aussehen:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Einer der wichtigsten Funktionen von `ipfstat` wird über die Option `-t` bereitgestellt. Mit ihr wird eine Statustabelle vergleichbar der Prozess-Tabelle von top(1) ausgegeben. Mit dieser Funktion erhalten Sie im Falle eines Angriffs die Möglichkeit, die angreifenden Pakete zu identifizieren, abzufangen und auszuwerten. Weitere Unteroptionen eröffnen, die IP-Adresse, den Port oder das Protokoll, geteilt nach Herkunft und Ziel, auszuwählen und dann in Echtzeit zu beobachten. Lesen Sie dazu bitte auch die Manpage von ipfstat(8).

31.5.6. IPMON

Damit der Befehl `ipmon` korrekt arbeiten kann, muss die Option `IPFILTER_LOG` in die Kernelkonfiguration eingearbeitet werden. Das Kommando selbst arbeitet in zwei verschiedenen Modi. Für den nativen Modus startet man `ipmon` auf der Kommandozeile ohne die Option `-D`.

Der Hintergrundmodus (`daemon mode`) dient der Erstellung eines stetigen Systemprotokolls, so dass Einträge vergangener Ereignisse inspiziert werden können. So sollen FreeBSD und IPFILTER entsprechend ihrer Konfiguration zusammen arbeiten. FreeBSD kann mit einem eingebauten Mechanismus Systemprotokolle turnusmäßig abspeichern. Aus diesem Grund sollte man besser syslogd(8) verwenden anstatt die Protokollinformationen in eine Datei zu schreiben, wie es als Standard vorgesehen ist. In der Standard-`rc.conf`-Datei (im Ordner `/etc/defaults/`) wird dem Eintrag `ipmon_flags` die Option `-Ds` übergeben:

```
ipmon_flags="-Ds" # D = Als Da:mon starten
# s = Protokollierung via syslog
# v = Protokollierung von tcp window, ack, seq
# n = Namen statt IP & port ausgeben
```

Die Vorteile des Protokollierens liegen auf der Hand: Sie versetzen den Administrator in die Lage, nach einem Vorfall Informationen abzurufen, etwa welche Pakete aussortiert wurden, welche Adressen diese Pakete gesendet haben oder wohin sie gesendet werden sollten. Alles in allem erhält er ein sehr gutes Werkzeug zum Aufspüren von Angreifern.

Jedoch, auch wenn die Protokollierung aktiviert ist, wird IPF keine einzige Regel zum Protokollieren von alleine entwerfen und umsetzen. Der Administrator der Firewall entscheidet, welche Regeln in seinem Regelsatz mitgeschrieben werden sollen und er muss dementsprechend das Schlüsselword `log` in dieser Regel angeben. Normalerweise werden nur Treffer auf abweisende Regeln protokolliert.

Es ist üblich, als letzte Regel eine alles blockierende Regel mit dem Schlüsselwort `log` in den Regelsatz einzutragen. Dadurch erkennt man alle Pakete, die keiner Regel im Regelsatz entsprachen.

31.5.7. IPMON Logging

Syslogd verwendet seine eigene Methode zum Sortieren der gesammtelten Protokolldaten - spezielle Gruppierungen namens “facility” und “level”. IPMON verwendet im `daemon`-Modus als “facility” den Wert `security`. Die folgenden “level” können für eine genauere Trennung der Protokolldaten verwendet werden:

```
LOG_INFO - Alle zu protokollierenden Pakete
LOG_NOTICE - Protokollierte Pakete, die passieren durften
LOG_WARNING - Protokollierte Pakete, die blockiert wurden
```

```
LOG_ERR - Protokollierte Pakete, deren Headerdaten nicht komplett vorlagen
```

Damit IPFILTER angewiesen werden kann, alle Protokolldaten in die Datei `/var/log/ipfilter.log` zu schreiben, muss diese erst erstellt werden. Folgendes Kommando übernimmt diese Aufgabe:

```
# touch /var/log/ipfilter.log
```

Die Funktionen von syslogd(8) werden durch Definition in der Datei `/etc/syslog.conf` gesteuert. In dieser Datei kann sehr weitläfig eingestellt werden, wie **syslog** mit den Systemnachrichten umgehen soll, die ihm von Anwendungen wie IPF übergeben werden.

Fügen Sie folgende Definition in die Datei `/etc/syslog.conf` ein, um die Protokollierung für IPF via `syslog` zu aktivieren:

```
security.* /var/log/ipfilter.log
```

`security.*` bedeutet, dass alle Nachrichten der Klasse `security.*` am angegebenen Ort (hier eine Datei) geschrieben werden sollen.

Um Änderungen an der Datei `/etc/syslog.conf` zu aktivieren müssen Sie den Rechner neu starten, oder den Befehl

```
# /etc/rc.d/syslogd reload
```

ausführen.

Vergessen Sie nicht, `/etc/newsyslog.conf` anzupassen, damit die neuen Protokolldateien, die eben konfiguriert wurden, auch in den Rotationsturnus eingefügt werden!

31.5.8. Die Formatierung der Logdatei

Nachrichten, die durch `ipmon` erzeugt werden, bestehen aus durch Leerstellen getrennten Datenfeldern. Folgende Felder sind in allen Nachrichten enthalten:

1. Das Datum der Paketerstellung.
2. Die Uhrzeit der Paketerstellung in der Form `HH:MM:SS.F`, mit Stunden, Minuten, Sekunden und Sekundenbruchteilen, wobei letztere mehrere Stellen lang sein können.
3. Der Name der Schnittstelle, die das Paket verarbeitet hat, bspw. `dc0`.
4. Die Gruppe und die Nummer der angewandten Regel, bspw. `@0:17`.
5. Die ausgeführte Aktion: p für `passed` (zugelassen), b für blockiert, S für `short packet` (unvollständiger Header), n für `no match` (gar keine Regel wurde berührt) und L für Log-Regel. Die Reihe, in der die Flags angezeigt werden ist: S, p, b, n, L. Ein groß geschriebenes P oder B bedeutet, dass das Paket aufgrund einer globalen Einstellung protokolliert wurde und nicht wegen einer einzelnen Regel.
6. Die Adressen. Diese bestehen aus drei Feldern: Der Quelladresse mit Port (getrennt durch ein Komma), dem Symbol “->” und der Zieladresse. Also bspw. `209.53.15.22,80 -> 198.64.221.18,1722`.
7. `PR` gefolgt vom Namen eines Netzwerk-Protokolls oder dessen Nummer. Bspw. `PR tcp`.
8. `len` gefolgt von der Länge des Headers und der Gesamtlänge des Paketes, beispielsweise `len 20 40`.

Wenn es sich um ein TCP-Paket handelt, wird ein weiteres Feld, beginnend mit einem Querstrich und gefolgt von Buchstaben, die den gesetzten Flags entsprechen, angezeigt. Lesen Sie bitte die Manpage ipmon(8) für eine Liste der Buchstaben und deren Bedeutungen.

Falls das Paket ein ICMP-Paket ist, werden zwei Felder am Ende hinzugefügt - das erstere ist immer “ICMP”, das zweite enthält die ICMP-Nachricht und den Nachrichtentyp, getrennt durch einen Schrägstrich. `ICMP 3/3` steht beispielsweise für “Port nicht erreichbar”.

31.5.9. Die Erstellung eines Regelsatzes mit Variablen

Erfahrenere IPF Anwender erstellen sich eine Datei, die die Regeln enthält und gestalten diese als ein Skript, in dem Variablen verwendet werden. Der wichtigste Vorteil besteht darin, dass man lediglich den Wert der Variablen anpassen muss und diese, sobald das Skript gestartet wird, durch die entsprechenden Werte ersetzt und die Regeln entsprechend formuliert werden. In Skripten kann man so häufig verwendete Werte einfach als Variable in mehreren Regeln zuweisen. Am folgenden Beispiel soll das verdeutlicht werden.

Die Syntax dieses Skriptes ist kompatibel mit den Shells sh(1), csh(1) und tcsh(1).

Variablen beginnen mit einem Dollar-Zeichen: `$Variablenname`. Im Beispiel unten steht `$oif` für die Variable, in der der Name der Schnittstelle abgelegt wird, über die der Verkehr nach außen erfolgt.

In Variablenzuweisungen fehlt das beginnende $-Zeichen. Alleine der Name der Variable wird angegeben, gefolgt von einem Gleichheitszeichen, und dem Wert, der der Variablen zugewiesen werden soll. Dieser muss in doppelten Anführungszeichen (`""`) stehen. Also folgt eine Zuweisung dem Schema `Variablenname = "Wert"`.

```
############# Start of IPF rules script ########################

oif="dc0"            # Name der Internet-Schnittstelle
odns="192.0.2.11"    # IP des DNS-Servers unseres ISPs
myip="192.0.2.7"     # die statische IP, die uns der ISP zugeteilt hat
ks="keep state"
fks="flags S keep state"

# Sie haben die Wahl, aus diesem Skript eine eigene
# /etc/ipf.rules erstellen zu lassen oder es einfach
# direkt als Skript laufen zu lassen.
#
# Entfernen Sie dazu das eine Kommentarzeichen
# und kommentieren Sie die andere Zeile aus!
#
# 1) Diese Zeile verwenden Sie zur Erstellung von /etc/ipf.rules
#cat > /etc/ipf.rules << EOF
#
# 2) Diese Zeile, wenn Sie direkt mit dem Skript arbeiten wollen
/sbin/ipf -Fa -f - << EOF

# Erlaubnis ausgehenden Verkehrs an den Nameserver des ISPs
pass out quick on $oif proto tcp from any to $odns port = 53 $fks
pass out quick on $oif proto udp from any to $odns port = 53 $ks

# Erlaubnis ausgehenden unsicheren www-Verkehrs
pass out quick on $oif proto tcp from $myip to any port = 80 $fks
```

```
# Erlaubnis ausgehenden sicheren www-Verkehrs https via TLS SSL
pass out quick on $oif proto tcp from $myip to any port = 443 $fks
EOF
################## End of IPF rules script ######################
```

Das ist schon alles. Die Regeln selbst sind im Beispiel nicht so wichtig - achten Sie auf die Anwendung der Variablenzuweisung am Anfang und die Verwendung der Variablen im Skript. Falls das obige Beispiel in einer Datei namens `/etc/ipf.rules.script` gespeichert wurde, können die Regeln mit folgenden Kommando neu geladen werden:

```
# sh /etc/ipf.rules.script
```

Es gibt ein Problem mit Regelsatz-Dateien, die Variablen verwenden: IPF kann mit Variablen nichts anfangen - und kann derartige Skripte nicht direkt einlesen.

Unser kleines Skript kann daher nur auf eine der beiden folgenden Weisen verwendet werden:

- Entfernen Sie das Kommentarzeichen der Zeile, die mit `cat` beginnt. Kommentieren Sie die Zeile aus, die mit `/sbin/ipf` beginnt. Schreiben Sie die Zeile `ipfilter_enable="YES"` in die Datei `/etc/rc.conf` und rufen Sie dann das Skript auf, um `/etc/ipf.rules` zu erstellen oder zu erneuern.
- Deaktivieren Sie IPFILTER in den Systemstart-Skripten, indem Sie die Zeile `ipfilter_enable="NO"` in die Datei `/etc/rc.conf` eintragen (was auch der Standard-Einstellung entspricht).

  Fügen Sie ein Skript ähnlich dem folgenden in Ihr Verzeichnis `/usr/local/etc/rc.d/`. Es sinnvoll, dem Skript einen offensichtlichen Namen zu geben, wie etwa `ipf.loadrules.sh`. Die Endung `.sh` ist dabei verbindlich.

  ```
  #!/bin/sh
  sh /etc/ipf.rules.script
  ```

  Die Zugriffsrechte für die Datei, die das Skript enthält, müssen für den Eigentümer `root` auf Lesen, Schreiben und Ausführen gesetzt werden.

  ```
  # chmod 700 /usr/local/etc/rc.d/ipf.loadrules.sh
  ```

Wenn nun Ihr System startet, werden Ihre IPF-Regeln geladen.

31.5.10. IPF Regelsätze

Ein Regelsatz ist eine Gruppe von IPF-Regeln, die anhand der Werte eines Netzwerkpaketes entscheiden, ob dieses Paket durchgelassen oder blockiert wird. Der Austausch von Paketen erfolgt immer zweiseitig in Form einer sogenannten Session. Der Regelsatz der Firewall verarbeitet sowohl die eingehenden Pakete aus dem öffentlichen Internet als auch die Pakete, die vom System als Antwort auf die Ersteren gesendet werden. Jeder Dienst, der via TCP/IP arbeitet, zum Beispiel `telnet`, `www` oder `mail`, ist vordefiniert durch sein Protokoll und seinen privilegierten Port, an dem er auf Anfragen wartet und reagieren kann. Pakete, die gezielt einen Dienst ansprechen sollen, werden von einem unprivilegierten Port des Senders an einen konkreten privilegierten Port des Zielsystems geschickt. Alle genannten Parameter (Ports, Adressen usw.) können als Auswahlkriterien zum erstellen von Regeln eingesetzt werden, die Dienste erlauben oder blockieren.

IPF wurde ursprünglich mit einer Regel-Prozess-Logik geschrieben, die ausschließlich statusfreie Regeln zuließ und nach dem Prinzip “die letzte Regel, die passt, entscheidet” arbeitete. Mit der Zeit erhielt IPF eine `quick` Option sowie `keep-state` Option für die Anwendung von zustandsorientierten Regeln, was die Regel-Prozess-Logik signifikant modernisierte.

Die Anweisungen in diesem Kapitel basieren auf der Verwendung von Regeln, die diese beiden neuen Optionen verarbeiten. Dies ist das Framework zur Entwicklung eines Firewallregelsatzes.

> **Warnung:** Wenn Sie mit einer Firewall arbeiten, seien Sie *sehr vorsichtig*. Durch wenige Einstellungen können Sie sich aus Ihrem System *aussperren*. Wenn Sie auf der sicheren Seite sein wollen, führen Sie die Firewall-Konfiguration direkt am entsprechenden Gerät aus und nicht über eine Netzwerkverbindung wie bspw. **ssh**.

31.5.11. IPF Regel-Syntax

Die Syntax zur Erstellung der Regeln, die hier vorgestellt wird, ist dahingehend vereinfacht worden, dass sie ausschliesslich auf den modernen Regelkontext, mit statusbehafteten Regeln und einer “die erste Regel, die passt, gewinnt”-Logik, zurückgreift. Um alles über die veraltete Syntax zu erfahren, lesen Sie bitte die Man-Page von ipf(8).

Ein #-Zeichen markiert den Beginn eines Kommentars. Es darf nach nach einer Regel stehen oder als erstes Zeichen einer Zeile. Leere Zeilen werden von der Regel-Prozess-Logik ignoriert.

Regeln enthalten Schlüsselwörter. Diese Schlüsselwörter müssen in einer bestimmten Reihenfolge von links nach rechts in einer Zeile erscheinen. Als solche identifizierte Schlüsselwörter werden fett wiedergegeben. Einige Schlüsselwörter haben Unteroptionen, die wiederum selbst Schlüsselwörter sein und ebenfalls weiter Unteroptionen einschließen können.

`ACTION IN-OUT OPTIONS SELECTION STATEFUL PROTO SRC_ADDR,DST_ADDR OBJECT PORT_NUM TCP_FLAG STATEFUL`

`ACTION` = block | pass

`IN-OUT` = in | out

`OPTIONS` = log | quick | on interface-name

`SELECTION` = proto value | source/destination IP | port = number | flags flag-value

`PROTO` = tcp/udp | udp | tcp | icmp

`SRC_ADD,DST_ADDR` = all | from object to object

`OBJECT` = IP address | any

`PORT_NUM` = port number

`TCP_FLAG` = S

`STATEFUL` = keep state

31.5.11.1. ACTION

Die “ACTION” bestimmt, was mit dem Paket passieren soll, wenn der Rest der Regel zutrifft. Dieser Teil muss für jede Regel angegeben werden.

Das Schlüsselwort `block` gibt an, dass das Paket verfallen soll, wenn die Auswahlparameter zutreffen.

Das Schlüsselwort `pass` gibt an, dass das Paket durch die Firewall durchgelassen werden soll, wenn die Auswahlparameter zutreffen.

31.5.11.2. IN-OUT

Ebenfalls verbindlich ist die Angabe, welchen Teil der Verbindung, Ein- oder Ausgang, die Regel beeinflussen soll. Das nächste Schlüsselwort muss daher entweder `in`, für eingehend, oder `out`, für ausgehend, lauten - oder die Regel wird aufgrund eines Syntaxfehlers nicht umgesetzt.

`in` bedeutet, dass diese Regel auf eingehende Pakete angewendet wird, die gerade an der dem öffentlichen Internet zugewandten Schnittstelle empfangen wurden.

`out` bedeutet, das diese Regel auf ausgehende Pakete angewendet wird, also Pakete die gerade gesendet werden und deren Zieladresse im öffentlichen Internet liegt.

31.5.11.3. OPTIONS

> **Anmerkung:** Die Optionen müssen in der hier aufgeführten Reihenfolge verwendet werden.

`log` bestimmt, dass die Kopfdaten des Paketes an die Systemschnittstelle ipl(4) geschrieben werden sollen. Genaueres dazu weiter unten im Abschnitt LOGGING.

`quick` bestimmt, dass, *wenn* die Auswahlkriterien der Regel auf das Paket zutreffen, keine weiteren Regeln ausgewertet werden. So vermeidet man das Abarbeiten des gesamten Regelsatzes. Diese Option ist eine verbindliche Vorraussetzung der modernen Regel-Prozess-Logik.

`on` bestimmt den Namen der Schnittstelle, der als Auswahlkriterium hinzugefügt werden soll. Die Namen aller verfügbaren Schnittstellen werden durch den Befehl ifconfig(8) angezeigt. wenn man diese Option verwendet, passt die Regeln nur auf Pakete, die durch diese Schnittstelle empfangen (`in`) oder gesendet (`out`) wurden. Für die modernisierte Regel-Prozess-Logik ist die Verwendung dieser Option verbindlich.

Wenn ein Paket protokolliert wird, werden die Kopfdaten in die Pseudo-Schnittstelle ipl(4) geschrieben. Folgende Parameter können zusätzlich übergeben werden, müssen dazu aber direkt nach dem Schlüsselwort `log` und in gleicher Reihenfolge stehen:

`body` bestimmt, dass die ersten 128 Bytes des Paketinhaltes zusätzlich zu den Kopfdaten protokolliert werden.

`first` trifft nur zu, wenn das Schlüsselwort `log` zusammen mit `keep-state` verwendet wird. Es bestimmt, dass nur das auslösende Paket protokolliert wird und nicht jedes weitere Paket, dass von der gespeicherten Status-Regel betroffen ist.

31.5.11.4. SELECTION

Die Schlüsselwörter, die in diesem Abschnitt vorgestellt werden, dienen zur Beschreibung von Attributen, anhand derer geprüft und entschieden wird, ob eine Regel zutrifft oder nicht. Es gibt ein Schlüsselwort, und das hat mehrere Optionen, von denen eine ausgewählt werden muss. Die folgenden allgemeinen Attribute können beliebig zum Erstellen einer Regel verwendet werden, allerdings nur in der vorgestellten Reihenfolge:

31.5.11.5. PROTO

`proto` ist das Schlüsselwort für das im Paket angewendete Protokoll. Als Option ein Protokoll aus Auswahlkriterium übergeben. Diese Option ist verbindlich, wenn man die moderne Regel-Prozess-Logik verwendet.

`tcp/udp | udp | tcp | icmp` oder irgendein Protokollname, der in der Datei `/etc/protocols` zu finden ist, kann übergeben werden. Außerdem kann das Schlüsselwort `tcp/udp` verwendet werden, wenn sowohl TCP als auch UDP von der Regel betroffen sein sollen. Dieses Schlüsselwort wurde eingeführt, um Duplikate sonst identischer Regeln zu vermeiden.

31.5.11.6. SRC_ADDR/DST_ADDR

Das Schlüsselwort `all` ist ein Synonym für “from any to any” ohne weitere Auswahlkriterien.

`from src to dst`: Die Schlüsselwörter `from` und `to` dienen zur Angabe von Quelle und Ziel in Form von IP-Adressen oder -Bereichen. Innerhalb einer Regel muss immer beides angegeben werden. Statt einer Adresse kann auch das Schlüsselwort `any` übergeben werden, das für jede beliebige IP-Adresse steht. Zum Beispiel: `from any to any` oder `from 0.0.0.0/0 to any` oder `from any to 0.0.0.0/0` oder `from 0.0.0.0 to any` oder `from any to 0.0.0.0` bedeuten alle das Gleiche.

IP-Bereiche können nur in der CIDR-Notation angegeben werden. Der Port `net-mgmt/ipcalc` hilft Ihnen bei der Berechnung der richtigen Angaben. Weiterführende Informationen zu CIDR finden Sie auf der Webseite von `ipcalc` (http://www.rfc-editor.org/rfc/rfc1519.txt).

31.5.11.7. PORT

Wenn ein Port als Auswahlkriterium übergeben wurde, bei Quelle und/oder Ziel, wird er nur bei TCP und UDP Paketen verwendet. Angegeben werden kann entweder die Portnummer oder der Dienstname aus `/etc/services`. Die Verwendung der Portoption mit dem `to`-Objekt ist verbindlich für die Verwendung der modernisierten Regel-Prozess-Logik. Ein Beispiel für die Filterung Paketen von allen Quell-IPs mit beliebiger Portnummer auf beliebige Ziel-IPs mit der Portnummer 80 (dem `www`-Port): `from any to any port = 80`.

Einfache Portvergleiche können auf verschiedenen Wegen erfolgen. Mehrere Vergleichsoperatoren stehen dafür zur Verfügung. Genauso können Bereiche angegeben werden.

port "=" | "!=" | "<" | ">" | "<=" | ">=" | "eq" | "ne" | "lt" | "gt" | "le" | "ge".

Um einen Bereich anzugeben: port "<>" | "><"

> **Warnung:** Genau wie die Trefferspezifikation für Quelle und Ziel sind auch die beiden folgenden Parameter obligatorisch bei der Verwendung der modernen Regel-Prozess-Logik.

31.5.11.8. TCP_FLAG

Flags spielen nur beim Filtern von TCP eine Rolle. Die Buchstaben entsprechen jeweils einem möglichen Flag, dass in den Kopfdaten der TCP-Pakete geprueft werden soll.

Die moderne Regel-Prozess-Logik verwendet den Parameter `flags S` um eine Anfrage zum Start einer TCP-Session zu identifizieren.

31.5.11.9. STATEFUL

`keep state` zeigt bei einer Passage-Regel an, dass für alle Pakete, die die Selektion erfolgreich durchlaufen, `Stateful Filtering` eingerichtet werden soll.

Anmerkung: Diese Option ist obligatorisch für die Verwendung der modernen Prozess-Regel-Logik.

31.5.12. Stateful Filtering

Stateful filtering treats traffic as a bi-directional exchange of packets comprising a session conversation. When activated, keep-state dynamically generates internal rules for each anticipated packet being exchanged during the bi-directional session conversation. It has sufficient matching capabilities to determine if the session conversation between the originating sender and the destination are following the valid procedure of bi-directional packet exchange. Any packets that do not properly fit the session conversation template are automatically rejected as impostors.

Keep state will also allow ICMP packets related to a TCP or UDP session through. So if you get ICMP type 3 code 4 in response to some web surfing allowed out by a keep state rule, they will be automatically allowed in. Any packet that IPF can be certain is part of an active session, even if it is a different protocol, will be let in.

What happens is:

Packets destined to go out through the interface connected to the public Internet are first checked against the dynamic state table. If the packet matches the next expected packet comprising an active session conversation, then it exits the firewall and the state of the session conversation flow is updated in the dynamic state table. Packets that do not belong to an already active session, are simply checked against the outbound ruleset.

Packets coming in from the interface connected to the public Internet are first checked against the dynamic state table. If the packet matches the next expected packet comprising an active session conversation, then it exits the firewall and the state of the session conversation flow is updated in the dynamic state table. Packets that do not belong to an already active session, are simply checked against the inbound ruleset.

When the conversation completes it is removed from the dynamic state table.

Stateful filtering allows you to focus on blocking/passing new sessions. If the new session is passed, all its subsequent packets will be allowed through automatically and any impostors automatically rejected. If a new session is blocked, none of its subsequent packets will be allowed through. Stateful filtering has technically advanced matching abilities capable of defending against the flood of different attack methods currently employed by attackers.

31.5.13. Inclusive Ruleset Example

The following ruleset is an example of how to code a very secure inclusive type of firewall. An inclusive firewall only allows services matching `pass` rules through, and blocks all others by default. Firewalls intended to protect other machines, also called “network firewalls”, should have at least two interfaces, which are generally configured to trust one side (the LAN) and not the other (the public Internet). Alternatively, a firewall might be configured to protect only the system it is running on—this is called a “host based firewall”, and is particularly appropriate for servers on an untrusted network.

All UNIX flavored systems including FreeBSD are designed to use interface `lo0` and IP address `127.0.0.1` for internal communication within the operating system. The firewall rules must contain rules to allow free unmolested movement of these special internally used packets.

The interface which faces the public Internet is the one to place the rules that authorize and control access of the outbound and inbound connections. This can be your user PPP `tun0` interface or your NIC that is connected to your DSL or cable modem.

In cases where one or more NICs are cabled to private network segments, those interfaces may require rules to allow packets originating from those LAN interfaces transit to each other and/or to the outside (Internet).

The rules should be organized into three major sections: first trusted interfaces, then the public interface outbound, and last the public untrusted interface inbound.

The rules in each of the public interface sections should have the most frequently matched rules placed before less commonly matched rules, with the last rule in the section blocking and logging all packets on that interface and direction.

The Outbound section in the following ruleset only contains `pass` rules which contain selection values that uniquely identify the service that is authorized for public Internet access. All the rules have the `quick`, `on`, `proto`, `port`, and `keep state` options set. The `proto tcp` rules have the `flag` option included to identify the session start request as the triggering packet to activate the stateful facility.

The Inbound section has all the blocking of undesirable packets first, for two different reasons. The first is that malicious packets may be partial matches for legitimate traffic. These packets have to be discarded rather than allowed in, based on their partial matches against `allow` rules. The second reason is that known and uninteresting rejects may be blocked silently, rather than being caught and logged by the last rules in the section. The final rule in each section, blocks and logs all packets and can be used to create the legal evidence needed to prosecute the people who are attacking your system.

Another thing that should be taken care of, is to ensure there is no response returned for any of the undesirable traffic. Invalid packets should just get dropped and vanish. This way the attacker has no knowledge if his packets have reached your system. The less the attackers can learn about your system, the more time they must invest before actually doing something bad. Rules that include a `log first` option, will only log the event the first time they are triggered. This option is included in the sample `nmap OS fingerprint` rule. The `security/nmap` utility is commonly used by attackers who attempt to identify the operating system of your server.

Any time there are logged messages on a rule with the `log first` option, an `ipfstat -hio` command should be executed to evaluate how many times the rule has actually matched. Large number of matches usually indicate that the system is being flooded (i.e.: under attack).

The `/etc/services` file may be used to lookup unknown port numbers. Alternatively, visit http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers and do a port number lookup to find the purpose of a particular port number.

Check out this link for port numbers used by Trojans http://www.sans.org/security-resources/idfaq/oddports.php.

The following ruleset creates a complete and very secure `inclusive` type of firewall ruleset that has been tested on production systems. It can be easily modified for your own system. Just comment out any `pass` rules for services that should not be authorized.

To avoid logging unwanted messages, just add a `block` rule in the inbound section.

The `dc0` interface name has to be changed in every rule to the real interface name of the NIC card that connects your system to the public Internet. For user PPP it would be `tun0`.

Add the following statements to `/etc/ipf.rules`:

```
######################################################################
# No restrictions on Inside LAN Interface for private network
# Not needed unless you have LAN
######################################################################

#pass out quick on xl0 all
#pass in quick on xl0 all

######################################################################
# No restrictions on Loopback Interface
######################################################################
pass in quick on lo0 all
pass out quick on lo0 all

######################################################################
# Interface facing Public Internet (Outbound Section)
# Match session start requests originating from behind the
# firewall on the private network
# or from this gateway server destined for the public Internet.
######################################################################

# Allow out access to my ISP's Domain name server.
# xxx must be the IP address of your ISP's DNS.
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
pass out quick on dc0 proto tcp from any to xxx port = 53 flags S keep state
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Allow out access to my ISP's DHCP server for cable or DSL networks.
# This rule is not needed for 'user ppp' type connection to the
# public Internet, so you can delete this whole group.
# Use the following rule and check log for IP address.
# Then put IP address in commented out rule & delete first rule
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state


# Allow out non-secure standard www function
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Allow out secure www function https over TLS SSL
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Allow out send & get email function
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Allow out Time
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Allow out nntp news
```

```
pass out quick on dc0 proto tcp from any to any port = 119 flags S keep state

# Allow out gateway & LAN users' non-secure FTP ( both passive & active modes)
# This function uses the IPNAT built in FTP proxy function coded in
# the nat rules file to make this single rule function correctly.
# If you want to use the pkg_add command to install application packages
# on your gateway system you need this rule.
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Allow out ssh/sftp/scp (telnet/rlogin/FTP replacements)
# This function is using SSH (secure shell)
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Allow out insecure Telnet
pass out quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Allow out FreeBSD CVSup
pass out quick on dc0 proto tcp from any to any port = 5999 flags S keep state

# Allow out ping to public Internet
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Allow out whois from LAN to public Internet
pass out quick on dc0 proto tcp from any to any port = 43 flags S keep state

# Block and log only the first occurrence of everything
# else that's trying to get out.
# This rule implements the default block
block out log first quick on dc0 all

#################################################################
# Interface facing Public Internet (Inbound Section)
# Match packets originating from the public Internet
# destined for this gateway server or the private network.
#################################################################

# Block all inbound traffic from non-routable or reserved address spaces
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 private IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 private IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 private IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #reserved for docs
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Class D & E multicast

##### Block a bunch of different nasty things. ############
# That I do not want to see in the log

# Block frags
block in quick on dc0 all with frags
```

```
# Block short tcp packets
block in quick on dc0 proto tcp all with short

# block source routed packets
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr

# Block nmap OS fingerprint attempts
# Log first occurrence of these so I can get their IP address
block in log first quick on dc0 proto tcp from any to any flags FUP

# Block anything with special options
block in quick on dc0 all with ipopts

# Block public pings
block in quick on dc0 proto icmp all icmp-type 8

# Block ident
block in quick on dc0 proto tcp from any to any port = 113

# Block all Netbios service. 137=name, 138=datagram, 139=session
# Netbios is MS/Windows sharing services.
# Block MS/Windows hosts2 name server requests 81
block in log first quick on dc0 proto tcp/udp from any to any port = 137
block in log first quick on dc0 proto tcp/udp from any to any port = 138
block in log first quick on dc0 proto tcp/udp from any to any port = 139
block in log first quick on dc0 proto tcp/udp from any to any port = 81

# Allow traffic in from ISP's DHCP server. This rule must contain
# the IP address of your ISP's DHCP server as it's the only
# authorized source to send this packet type. Only necessary for
# cable or DSL configurations. This rule is not needed for
# 'user ppp' type connection to the public Internet.
# This is the same IP address you captured and
# used in the outbound section.
pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Allow in standard www function because I have apache server
pass in quick on dc0 proto tcp from any to any port = 80 flags S keep state

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID/PW passed over public Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
#pass in quick on dc0 proto tcp from any to any port = 23 flags S keep state

# Allow in secure FTP, Telnet, and SCP from public Internet
# This function is using SSH (secure shell)
pass in quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Block and log only first occurrence of all remaining traffic
# coming into the firewall. The logging of only the first
# occurrence avoids filling up disk with Denial of Service logs.
# This rule implements the default block.
```

```
block in log first quick on dc0 all
################### End of rules file ######################################
```

31.5.14. NAT

NAT stands for *Network Address Translation*. To those familiar with Linux, this concept is called IP Masquerading; NAT and IP Masquerading are the same thing. One of the many things the IPF NAT function enables is the ability to have a private Local Area Network (LAN) behind the firewall sharing a single ISP assigned IP address on the public Internet.

You may ask why would someone want to do this. ISPs normally assign a dynamic IP address to their non-commercial users. Dynamic means that the IP address can be different each time you dial in and log on to your ISP, or for cable and DSL modem users, when the modem is power cycled. This dynamic IP address is used to identify your system to the public Internet.

Now lets say you have five PCs at home and each one needs Internet access. You would have to pay your ISP for an individual Internet account for each PC and have five phone lines.

With NAT only a single account is needed with your ISP. The other four PCs may then be cabled to a switch and the switch to the NIC in your FreeBSD system which is going to service your LAN as a gateway. NAT will automatically translate the private LAN IP address for each separate PC on the LAN to the single public IP address as it exits the firewall bound for the public Internet. It also does the reverse translation for returning packets.

There is a special range of IP addresses reserved for NATed private LANs. According to RFC 1918, the following IP ranges may be used for private nets which will never be routed directly to the public Internet:

Start IP `10.0.0.0`	-	Ending IP `10.255.255.255`
Start IP `172.16.0.0`	-	Ending IP `172.31.255.255`
Start IP `192.168.0.0`	-	Ending IP `192.168.255.255`

31.5.15. IPNAT

NAT rules are loaded by using the `ipnat` command. Typically the NAT rules are stored in `/etc/ipnat.rules`. See ipnat(1) for details.

When changing the NAT rules after NAT has been started, make your changes to the file containing the NAT rules, then run the `ipnat` command with the `-CF` flags to delete the internal in use NAT rules and flush the contents of the translation table of all active entries.

To reload the NAT rules issue a command like this:

```
# ipnat -CF -f /etc/ipnat.rules
```

To display some statistics about your NAT, use this command:

```
# ipnat -s
```

To list the NAT table’s current mappings, use this command:

```
# ipnat -l
```

To turn verbose mode on, and display information relating to rule processing and active rules/table entries:

```
# ipnat -v
```

31.5.16. IPNAT Rules

NAT rules are very flexible and can accomplish many different things to fit the needs of commercial and home users.

The rule syntax presented here has been simplified to what is most commonly used in a non-commercial environment. For a complete rule syntax description see the ipnat(5) manual page.

The syntax for a NAT rule looks something like this:

```
map IF LAN_IP_RANGE -> PUBLIC_ADDRESS
```

The keyword `map` starts the rule.

Replace *`IF`* with the external interface.

The *`LAN_IP_RANGE`* is what your internal clients use for IP Addressing, usually this is something like `192.168.1.0/24`.

The *`PUBLIC_ADDRESS`* can either be the external IP address or the special keyword `0/32`, which means to use the IP address assigned to *`IF`*.

31.5.17. How NAT works

A packet arrives at the firewall from the LAN with a public destination. It passes through the outbound filter rules, NAT gets its turn at the packet and applies its rules top down, first matching rule wins. NAT tests each of its rules against the packet's interface name and source IP address. When a packet's interface name matches a NAT rule then the source IP address (i.e.: private LAN IP address) of the packet is checked to see if it falls within the IP address range specified to the left of the arrow symbol on the NAT rule. On a match the packet has its source IP address rewritten with the public IP address obtained by the `0/32` keyword. NAT posts an entry in its internal NAT table so when the packet returns from the public Internet it can be mapped back to its original private IP address and then passed to the filter rules for processing.

31.5.18. Enabling IPNAT

To enable IPNAT add these statements to `/etc/rc.conf`.

To enable your machine to route traffic between interfaces:

```
gateway_enable="YES"
```

To start IPNAT automatically each time:

```
ipnat_enable="YES"
```

To specify where to load the IPNAT rules from:

```
ipnat_rules="/etc/ipnat.rules"
```

31.5.19. NAT for a very large LAN

For networks that have large numbers of PC's on the LAN or networks with more than a single LAN, the process of funneling all those private IP addresses into a single public IP address becomes a resource problem that may cause problems with the same port numbers being used many times across many NATed LAN PC's, causing collisions. There are two ways to relieve this resource problem.

31.5.19.1. Assigning Ports to Use

A normal NAT rule would look like:

```
map dc0 192.168.1.0/24 -> 0/32
```

In the above rule the packet's source port is unchanged as the packet passes through IPNAT. By adding the `portmap` keyword, IPNAT can be directed to only use source ports in the specified range. For example the following rule will tell IPNAT to modify the source port to be within the range shown:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000
```

Additionally we can make things even easier by using the `auto` keyword to tell IPNAT to determine by itself which ports are available to use:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto
```

31.5.19.2. Using a Pool of Public Addresses

In very large LANs there comes a point where there are just too many LAN addresses to fit into a single public address. If a block of public IP addresses is available, these addresses can be used as a "pool", and IPNAT may pick one of the public IP addresses as packet-addresses are mapped on their way out.

For example, instead of mapping all packets through a single public IP address, as in:

```
map dc0 192.168.1.0/24 -> 204.134.75.1
```

A range of public IP addresses can be specified either with a netmask:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/255.255.255.0
```

or using CIDR notation:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/24
```

31.5.20. Port Redirection

A very common practice is to have a web server, email server, database server and DNS server each segregated to a different PC on the LAN. In this case the traffic from these servers still have to be NATed, but there has to be some way to direct the inbound traffic to the correct LAN PCs. IPNAT has the redirection facilities of NAT to solve this problem. For example, assuming a web server operating on LAN address `10.0.10.25` and using a single public IP address of `20.20.20.5` the rule would be coded as follows:

```
rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

or:

```
rdr dc0 0.0.0.0/0 port 80 -> 10.0.10.25 port 80
```

or for a LAN DNS Server on LAN address of `10.0.10.33` that needs to receive public DNS requests:

```
rdr dc0 20.20.20.5/32 port 53 -> 10.0.10.33 port 53 udp
```

31.5.21. FTP and NAT

FTP is a dinosaur left over from the time before the Internet as it is known today, when research universities were leased lined together and FTP was used to share files among research Scientists. This was a time when data security was not a consideration. Over the years the FTP protocol became buried into the backbone of the emerging Internet and its username and password being sent in clear text was never changed to address new security concerns. FTP has two flavors, it can run in active mode or passive mode. The difference is in how the data channel is acquired. Passive mode is more secure as the data channel is acquired by the ordinal ftp session requester. For a real good explanation of FTP and the different modes see http://www.slacksite.com/other/ftp.html.

31.5.21.1. IPNAT Rules

IPNAT has a special built in FTP proxy option which can be specified on the NAT map rule. It can monitor all outbound packet traffic for FTP active or passive start session requests and dynamically create temporary filter rules containing only the port number really in use for the data channel. This eliminates the security risk FTP normally exposes the firewall to from having large ranges of high order port numbers open.

This rule will handle all the traffic for the internal LAN:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
```

This rule handles the FTP traffic from the gateway:

```
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
```

This rule handles all non-FTP traffic from the internal LAN:

```
map dc0 10.0.10.0/29 -> 0/32
```

The FTP map rule goes before our regular map rule. All packets are tested against the first rule from the top. Matches on interface name, then private LAN source IP address, and then is it a FTP packet. If all that matches then the special FTP proxy creates temp filter rules to let the FTP session packets pass in and out, in addition to also NATing the FTP packets. All LAN packets that are not FTP do not match the first rule and fall through to the third rule and are tested, matching on interface and source IP, then are NATed.

31.5.21.2. IPNAT FTP Filter Rules

Only one filter rule is needed for FTP if the NAT FTP proxy is used.

Without the FTP Proxy, the following three rules will be needed:

```
# Allow out LAN PC client FTP to public Internet
# Active and passive modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state

# Allow out passive mode data channel high order port numbers
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state

# Active mode let data channel in from FTP server
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

31.6. IPFW

Die *IPFIREWALL* (IPFW) ist eine vom FreeBSD Project gesponserte Software-Firewall. Sie wurde und wird freiwillig von Mitgliedern des FreeBSD Projects geschrieben und gewartet. Mit zustandslosen Regeln und einer Grammatik für Regeln implementiert sie eine sogenannte “Einfache Zustandsgesteuerte Logik”.

Die Standardinstallation von IPFW enthält einen beispielhaften Regelsatz (`/etc/rc.firewall` und `/etc/rc.firewall6`). Dieser ist eher einfach gehalten; es ist nicht zu erwarten, dass dieser ohne Modifikationen angewandt werden kann. Dieses Beispiel nutzt keine zustandsorientierte Filterung, von der allerdings die meisten Installationen profitieren sollten. Deshalb wird sich dieser Abschnitt auch nicht auf diese Beispiele stützen.

Die zustandslose IPFW Regel-Syntax ist durch ihre technisch ausgefeilten Selektions-Fähigkeiten, die über das Niveau der gebrächlichen Firewall-Installationsprogramme weit hinausgehen, sehr mächtig. IPFW richtet sich an professionelle oder technisch versierte Nutzer mit weitergehenden Anforderungen an die Paket-Auswahl. Um die Ausdrucksstärke der IPFW zu nutzen, ist sehr detailliertes Wissen über die Art und Weise, wie verschiedene Protokolle ihre jeweilige Paket-Header-Information erzeugen und nutzen, erforderlich. Im Rahmen dieses Abschnitts ist es nicht möglich, auf alle diese Punkte detailliert einzugehen.

IPFW besteht aus sieben Komponenten: Hauptbestandteil ist der Kernel Firewall Filter, ein Regel-Prozessor mit integrierter Paket-Buchführung. Außerdem enthalten ist eine Komponente zur Protokollierung der Aktivitäten der Firewall (also ein Logfunktion). Weiters besteht die IPFW aus einer Regel zum Umleiten des Datenverkehrs (`divert`), die auch Network Address Translation (NAT) unterstützt. Die restlichen Bestandteile dienen verschiedenen fortgeschrittenen Zwecken. Der *Traffic Shaper* dummynet(4) gestattet es beispielsweise, den Datenverkehr zu lenken, während die `fwd`-Regel zum Weiterleiten von Datenpaketen dient. Komplettiert wird IPFW durch Funktionen zum Überbrücken von Netzwerkgrenzen (*Bridge*-Funktion) sowie *ipstealth*, das es gestattet, bridging-Funktionen durchzuführen, ohne dabei das TTL-Feld im IP-Paket zu erhöhen. IPFW unterstützt IPv4 und IPv6.

31.6.1. IPFW aktivieren

IPFW ist in der FreeBSD-Installation standardmäßig als ein zur Laufzeit ladbares Kernelmodul enthalten, das vom System automatisch geladen wird, wenn in der Datei `rc.conf` die Option `firewall_enable="YES"` gesetzt wird. Es ist daher in der Regel nicht notwendig, IPFW statisch in den Kernel zu kompilieren. Es sei denn, man benötigt die NAT-Funktionalität.

Während des Systemstart wird bei gesetzter Option `firewall_enable="YES"` (in der Datei `rc.conf`) folgende Nachricht ausgegeben:

```
ipfw2 initialized, divert disabled, rule-based forwarding disabled, default to deny, logging disa
```

Das Kernelmodul hat eine Protokollierungsfunktion. Um diese zu aktivieren und einen Schwellwert für die Protokollierung zu definieren, ist es erforderlich, folgende Ausdrücke der `/etc/sysctl.conf` hinzuzufügen:

```
net.inet.ip.fw.verbose=1
net.inet.ip.fw.verbose_limit=5
```

31.6.2. Kerneloptionen

IPFW muss nicht durch einkompilieren bestimmter, im folgenden konkretisierter Optionen in den Kernel aktiviert werden, es sei denn, man benötigt NAT-Funktionalität. Die erforderlichen Optionen werden deshalb hier lediglich als Hintergrundinformation aufgeführt.

```
options IPFIREWALL
```

Diese Option aktiviert IPFW als Bestandteil des Kernels.

```
options IPFIREWALL_VERBOSE
```

Diese Option aktiviert die Funktion, alle Pakete, die durch IPFW verarbeitet werden und bei denen das Schlüsselwort `log` gesetzt ist, zu protokollieren.

```
options IPFIREWALL_VERBOSE_LIMIT=5
```

Diese Option limitiert die Anzahl der durch syslogd(8) protokollierten Pakete auf das angegebene Maximum. Sie wird in feindlichen Umgebungen verwandt, in denen die Protokollierung der Firewall-Aktivität erwünscht ist. Dadurch wird ein möglicher Denial-of-Service-Angriff durch Überflutung von syslogd(8) verhindert.

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
```

Diese Option erlaubt allen Paketen, die Firewall zu passieren. Diese Einstellung kann beispielsweise bei der ersten Konfiguration der Firewall hilfreich sein.

```
options IPDIVERT
```

Dies aktiviert die Nutzung der NAT-Funktionalität.

> **Anmerkung:** Die Firewall wird alle eingehenden oder ausgehenden Pakete blockieren, wenn entweder die Kernel-Option `IPFIREWALL_DEFAULT_TO_ACCEPT` fehlt oder aber keine Regel, die die betreffenden Verbindungen explizit gestattet, existiert. Dies enstpricht im Wesentlichen der Einstellung “default to deny”

31.6.3. Optionen in `/etc/rc.conf`

Der Eintrag

```
firewall_enable="YES"
```

aktiviert die Firewall während des Systemstarts.

Die Auswahl einer für FreeBSD verfügbaren Firewall erfolgt durch einen entsprechenden Eintrag in der Datei `/etc/rc.firewall`, durch den der Firewalltyp festgelegt wird.

```
firewall_type="open"
```

Konkret sind folgende Einträge erlaubt:

- `open` — gestattet jeglichen Datenverkehr
- `client` — schützt nur die jeweilige Maschine (Client/Mandant)
- `simple` — schützt das gesamte Netzwerk
- `closed` — unterbindet jeglichen IP-Datenverkehr mit Ausnahme des Verkehrs über die Loopback-Schnittstelle.
- `UNKNOWN` — deaktiviert das Laden von Firewallregeln
- *`filename`* — absoluter Pfad zu einer Datei, in der die Firewallregeln definiert sind

Angepasste Regeln für ipfw(8) können auf zwei verschiedene Arten geladen werden. Einerseits kann man durch die Variable `firewall_type` den absoluten Pfad der Datei angeben, welche die *Firewallregeln* (ohne weitere Optionen) für ipfw(8) enthält. Ein einfaches Beispiel für einen Regelsatz, der jeglichen eingehenden und ausgehenden Datenverkehr blockiert, könnte beispielsweise so aussehen:

```
add deny in add deny out
```

Andererseits ist es möglich, den Wert der `firewall_type`-Variable mit dem absoluten Pfad einer Datei zu belegen, die (als ausführbares Skript) die ipfw(8)-Kommandos enthält, die beim Booten ausgeführt werden sollen. Ein gültiges Skript (das die gleiche Funktion hat wie die Zeile im letzten Beispiel) könnte beispielsweise so aussehen:

```
#!/bin/sh

ipfw -q flush

ipfw add deny in
ipfw add deny out
```

> **Anmerkung:** Wenn die Variable `firewall_type` entweder auf `client` oder `simple` gesetzt ist, sollten die Standardregeln in der Datei `/etc/rc.firewall` geprüft und an die Konfiguration der gegebenen Maschine angepasst werden. Beachten Sie dabei bitte, dass die Beispiele dieses Kapitels davon ausgehen, dass das `firewall_script` auf `/etc/ipfw.rules` gesetzt ist.

Das Logging wird durch folgenden Eintrag aktiviert:

```
firewall_logging="YES"
```

> **Warnung:** Die Variable `firewall_logging` definiert lediglich die sysctl-Variable als `net.inet.ip.fw.verbose = 1` (lesen Sie dazu bitte auch den Abschnitt Abschnitt 31.6.1 des Handbuchs). Es gibt keine `rc.conf`-Variable, mit der man Protokollierungsschwellen setzen könnte. Dies kann lediglich über sysctl(8) geschehen, wobei Sie in der Datei `/etc/sysctl.conf` nur Werte > 1 angeben sollten:
>
> ```
> net.inet.ip.fw.verbose_limit=5
> ```

Sollte Ihre Maschinen als Gateway fungieren (also mittels natd(8) *Network Address Translation* (NAT) durchführen), finden Sie in Abschnitt Abschnitt 32.9 weitere Optionen für die `/etc/rc.conf`.

31.6.4. Der Befehl IPFW

Mit ipfw(8) ist es möglich, im laufenden Betrieb einzelne Regeln hinzuzufügen oder zu entfernen. Problematisch ist allerdings, dass diese Änderungen verloren gehen, wenn das System neu gestartet wird. Daher ist es empfehlenswert, eigene Regeln in einer Datei zu definieren und diese zu laden, um die Regeln der Firewall im laufenden Betrieb anzupassen.

ipfw(8) ist jedoch hilfreich, um die Regeln der laufenden Firewall in der Konsole auszugeben. IPFW erzeugt dynamisch einen Zähler, der jedes Paket, auf das eine Regel zutrifft, zählt. Dadurch wird es möglich, die Funktion einer Regel zu überprüfen.

Eine sequentielle Liste aller Regeln erhalten Sie mit:

```
# ipfw list
```

Eine Liste aller Regeln inklusive des letzten Treffers erhalten Sie durch den folgenden Befehl:

```
# ipfw -t list
```

Um eine Liste aller Regeln inklusive der Anzahl der Pakete, die von einer Regel gefiltert wurden, zu erhalten, geben Sie den folgenden Befehl ein:

```
# ipfw -a list
```

Eine Liste, die zusätzlich allen dynamischen Regeln enthält, erhalten Sie mit:

```
# ipfw -d list
```

Um diese Liste um alle “abgelaufenen” Regeln zu erweitern, ädern Sie diesen Befehl wie folgt ab:

```
# ipfw -d -e list
```

Alle Zähler auf Null zurücksetzen:

```
# ipfw zero
```

Es ist auch möglich, einen spezifischen Zähler auszuwählen und zurückzusetzen:

```
# ipfw zero NUM
```

31.6.5. IPFW-Regeln

Ein Regelwerk ist eine Menge von IPFW-Regeln, die in Abhängigkeit von bestimmten Paketeigenschaften Pakete entweder passieren lassen oder abweisen. Der zustandshafte bidirektionale Transfer von Paketen zwischen Rechnern wird als Sitzung bezeichnet. Das Regelwerk der Firewall verarbeitet sowohl ankommende Pakete (aus dem öffentlichen Internet) als auch Pakete, deren Ursprung in einer Antwort des Systems auf empfangene Pakete liegt.

Jeder TCP/IP-Dienst (wie telnet, www, mail) ist durch sein Protokoll und durch den priveligierten (eingehenden) Port definiert. An einen spezifischen Dienst adressierte Pakete kommen von einer Quelladresse und einem unprivilegierten (high order) Port. Sie adressieren den spezifischen Port des Dienstes an der Zieladresse. Alle weiter oben aufgeführten Parameter (also Ports und Adressen) können als Selektionskriterium zur Erzeugung von Regeln genutzt werden, die ein Passieren der Firewall für oder ein Blockieren von Diensten bewirken.

Wenn ein Paket die Firewall "betritt", also von der Firewall geprüft und verarbeitet wird, wird die erste Regel des Regelwerkes auf das Paket angewandt. Auf diese Weise wird in aufsteigender Reihenfolge der Regelnummer mit allen weiteren Regeln verfahren. Falls die Selektionsparameter einer Regel auf ein Paket zutreffen, wird das Aktionsfeld der Regel ausgeführt und die Prüfung des Pakets beendet, nachfolgende Regeln werden also nicht mehr geprüft. Diese Suchmethode wird als "erster Treffer gewinnt" bezeichnet. Falls keine Regel auf das betreffende Paket zutrifft, wird die obligatorische IPFW-Rückfallregel (also Regel 65535) angewendet und das Paket wird ohne Rückantwort verworfen.

> **Anmerkung:** Die Prüfung der Regeln wird nach Treffern von mit `count`, `skipto` und `tee` parametrisierten Regeln ungeachtet des "erster Treffer gewinnt"-Prinzips weiter fortgeführt.

Die Anweisungen basieren auf der Nutzung von Regeln mit den zustandsgesteuerten Optionen `keep`, `state`, `limit`, `in` und `out`. Diese bilden die Basis für die Spezifikation von Firewallregeln.

> **Warnung:** Bei der Arbeit mit Firewallregeln ist Vorsicht geboten. Es ist sehr einfach, sich selbst auszuschließen.

31.6.5.1. Syntax der Firewallregeln

Mit der in diesem Abschnitt dargestellten Syntax der Regeln kann ein Standardregelsatz für eine "einschließende" Firewall erstellt werden. Für eine vollständige Beschreibung der Regelsyntax lesen Sie bitte die Manualpage ipfw(8)

Regelausdrücke werden "von links nach rechts" ausgewertet. Schlüsselwörter werden in fetter Schrift dargestellt. Manche Schlüsselworte beinhalten Unteroptionen, die wiederum selbst aus Schlüsselworten samt Optionen bestehen können.

Kommentare sind mit einen führenden Doppelkreuz (#) ausgezeichnet. Sie können am Ende einer Regel oder in einzelnen, separaten Zeilen stehen. Leerzeilen werden ignoriert.

```
CMD RULE_NUMBER ACTION LOGGING SELECTION STATEFUL
```

31.6.5.1.1. CMD

Jede neue Regel benötigt das Präfix `add`, um die Regel der internen Tabelle hinzuzufügen.

31.6.5.1.2. RULE_NUMBER

Zu jeder Regel gehört eine Regelnummer zwischen 1 und 65535.

31.6.5.1.3. ACTION

Eine Regel kann mit einer der vier folgenden Aktionen verbunden sein, die ausgeführt werden, wenn ein Paket den Selektionskriterien der Regel entspricht.

`allow | accept | pass | permit`

Alle diese Aktionen bewirken das Gleiche: Pakete, die den Selektionskriterien der Regel entsprechen, verlassen den Regelprüfungsabschnitt der Firewall und die Regelprüfung wird beendet.

`check-state`

Diese Aktion prüft das Paket gegen die Regeln aus den dynamischen Regeltabellen. Trifft ein Selektionskriterium zu, wird die zur dynamischen Regel gehörende Aktion ausgeführt. Anderenfalls wird gegen die nächste Regel geprüft. Die `check-state`-Regel selbst hat kein Selektionskriterium. Sollte eine `check-state`-Regel im Regelwerk fehlen, wird gegen die erste `keep-state`- oder `limit`-Regel in den dynamischen Regeln geprüft.

`deny | drop`

Beide Schlüsselworte bewirken dieselbe Aktion: Ein Paket, dass die Selektionskriterien der Regel erfüllt, wird verworfen und die Regelprüfung wird beendet.

31.6.5.1.4. Protokollierung

`log` oder *`logamount`*

Erfüllt ein Paket die Selektionskriterien mit dem Schlüsselwort `log`, wird dies von syslogd(8) mit der Annotation SECURITY protokolliert. Dies erfolgt allerdings nur, wenn die Anzahl der protokollierten Pakete der betreffenden Regel die im `logamount`-Parameter definierte Schwelle nicht übersteigt. Ist der Parameter `logamount` nicht definiert, wird diese Grenze aus der `sysctl`-Variable `net.inet.ip.fw.verbose_limit` ermittelt. Ist einer dieser beiden Werte auf “Null” gesetzt, wird unbegrenzt protokolliert. Wurde hingegen ein definierter Schwellenwert erreicht, wird die Protokollierung deaktiviert. Um sie zu reaktivieren, können Sie entweder den Protokoll- oder den Paketzähler rücksetzen (und zwar über den Befehl `ipfw reset log`).

> **Anmerkung:** Die Protokollierung findet statt, nachdem alle Paketselektionskriterien geprüft und bevor die daraus folgende, endgültige Aktion (`accept` oder `deny`) auf das Paket ausgeführt wird. Die Entscheidung, welche Regel protokolliert werden soll, bleibt Ihnen überlassen.

31.6.5.1.5. Selektion

Die in diesem Abschnitt beschriebenen Schlüsselwörter beschreiben die Attribute eines Pakets, durch die bestimmt wird, ob eine Regel auf ein Paket zutrifft. Die folgenden Attribute dienen der Bestimmung des Protokolls und müssen in der angegebenen Reihenfolge verwendet werden.

`udp | tcp | icmp`

Weitere in `/etc/protocols` angegebene Protokolle werden ebenfalls erkannt und können daher verwendet werden, um das Protokoll zu definieren, gegen das Pakete geprüft werden. Die Angabe des Protokolls ist verpflichtend.

`from src to dst`

Die Schlüsselwörter `from` und `to` beziehen sich auf IP-Adressen und definieren sowohl Ursprungs- als auch Zieladresse einer Datenverbindung. Firewallregeln müssen Parameter für den Ursprung *und* das Ziel enthalten. Das

Schlüsselwort `any` steht für beliebige IP-Adressen. Bei `me` handelt es sich um ein spezielles Schlüsselwort, das alle IP-Adressen beschreibt, die einer bestimmten Netzwerkschnittstelle Ihres Systems (auf dem die Firewall läuft) zugeordnet sind. Beispiele hierfür sind `from me to any`, `from any to me`, `from 0.0.0.0/0 to any` , `from any to 0.0.0.0/0`, `from 0.0.0.0 to any`, `from any to 0.0.0.0` oder `from me to 0.0.0.0`. IP-Adressen werden entweder in CIDR-Notation oder durch Punkte getrennt mit Suffixen (`192.168.2.101/24`) für die Netzmaske oder als einzelne numerische, durch Punkte getrennte Adressen (`192.168.2.101`) angegeben. Die dafür notwendigen Berechnungen erleichtert der Port `net-mgmt/ipcalc`. Weiterführende Informationen finden sich auf http://jodies.de/ipcalc.

`port number`

Bei der Verarbeitung von Protokollen wie TCP oder UDP, die Portnummern verwenden, muss die Portnummer des betreffenden Dienstes angegeben werden. Anstelle der Portnummer kann auch der in der Datei `/etc/services` definierte Name des Dienstes angegeben werden.

`in | out`

Diese Schlüsselwörter beziehen sich auf die Richtung des Datenverkehrs. Jede Regel *muss* eines dieser beiden Schlüsselwörter enthalten.

`via IF`

Eine Regel mit dem Schlüsselwort `via IF` betrifft nur Pakete, die über die angegebene Schnittstellte geroutet werden (ersetzen Sie `IF` durch den Namen Ihrer Netzwerkschnittstelle). Die Angabe des Schlüsselwortes `via` bewirkt, dass die Netzwerkschnittstelle in die Regelprüfung aufgenommen wird.

`setup`

Dieses obligatorische Schlüsselwort bezeichnet die Anforderung des Sitzungsstarts für TCP-Pakete.

`keep-state`

Dieses obligatorische Schlüsselwort bewirkt, dass die Firewall eine dynamische Regel erzeugt, die bidirektionalen Datenverkehr zwischen Ursprungs- und Zieladresse sowie Ursprungs- und Zielport prüft, der das gleiche Protokoll verwendet.

`limit {src-addr | src-port | dst-addr | dst-port}`

Wird das Schlüsselwort `limit` verwendet, sind nur `N` durch diese Regel definierte Verbindungen erlaubt. Es können dabei ein oder mehrere Ursprungs- und Zieladressen sowie ein oder mehrere Ports angegeben werden. Die Schlüsselwörter `limit` und `keep-state` können nicht in derselben Regel verwendet werden. Die Option `limit` bewirkt dieselbe Zustandsteuerung wie die Option `keep-state`, erweitert diese jedoch um eigene Regeln.

31.6.5.2. Optionen für zustandsgesteuerte Regeln

Eine zustandsgesteuerte Filterung behandelt Datenverkehr als einen bidirektionalen Austausch von Datenpaketen (die eine sogenannte Konversation innerhalb einer Sitzung darstellen). Sie ist in der Lage, zu bestimmen, ob die Konversation von originärem Sender und Empfänger gültigen Prozeduren des bidirektionalen Pakettausches entspricht. Pakete, die dem Muster von Konversationen in Sitzungen nicht folgen, werden automatisch als “Betrüger” abgelehnt.

Die `check-state`-Option wird verwendet, wo genau innerhalb des IPFW-Regelwerks die Prüfung dynamischer Regeln stattfinden soll. Erfüllt ein Datenpaket die Selektionskriterien der Regel, verlässt das Paket die Firewall. Gleichzeitig wird eine neue dynamische Regel erzeugt, die für das nächste Paket der bidirektionalen Konversation in

der Sitzung vorgesehen ist. Falls ein Paket die (dyanmische) Regel nicht erfüllt, wird es gegen die nächste Regel im Regelwerk geprüft.

Dynamische Regeln sind für einem sogenannten *SYN-flood*-Angriff anfällig, bei dem eine riesige Anzahl "schwebender" dynamischer Regelprüfungungsinstanzen erzeugt wird. Um einem solchen Angriff zu begegnen, wurde in FreeBSD die neue Option `limit` geschaffen. Diese Option begrenzt die Anzahl der gleichzeitig möglichen Sitzungen und/oder Konversationen. Es handelt sich dabei um einen Zähler, der die Anzahl von Instanzen dynamischer Regelprüfungen in Abhängigkeit von einer eindeutigen Urspungs- und Quelladresskombination zählt. Übersteigt der Zähler den durch `limit` definierten Schwellenwert, wird das Paket verworfen.

31.6.5.3. Protokollierung von Firewall-Nachrichten

Die Vorteile einer Protokollierung sind offensichtlich. Sie ermöglicht nach Aktivierung von Regeln zu untersuchen, welche Pakete verworfen wurden, von wo diese stammen und für welche Systeme sie bestimmt waren. Diese Informationen sind sehr nützlich bei der Erkennung eventueller Angriffe sowie bei deren Abwehr.

IPFW protokolliert nur jene Regeln, für die ein Administrator dies explizit aktiviert. Ein Aktivieren der Protolllfunktion führt also nicht dazu, dass automatisch alle Regeln protokolliert werden. Vielmehr entscheidet der Administrator der Firewall, welche Regeln protokolliert werden sollen. Dazu wird die Option `log` für diese Regeln aktiviert. Im Regelfall werden nur `deny`-Regeln protokolliert, beispielsweise die `deny`-Regel für eintreffende ICMP-Nachrichten. Üblicherweise wird die "ipfw default deny everything"-Regel doppelt angelegt. Einmal mit und einmal ohne aktivierte Option `log`. Dadurch erhält man eine Auflistung aller Pakete, auf die keine Regel zutraf.

Protokollierung ist allerdings ein zweischneidiges Schwert, bei mangelnder Vorsicht wird man mit einer enormen Flut von Protokollierungsdaten förmlich *überschwemmt* und belastet zusätzlich die Festplatte des Systems durch rasch wachsende Protokolldateien. DoS-Angriffe, die auf diese Art und Weise Festplatten an die Kapazitätsgrenze treiben, gehören zu den ältesten Angriffen überhaupt. Außerdem werden Protokollnachrichten nicht nur an syslogd(8) geschickt, sondern auch auf einem root-Terminal angezeigt.

Die Kerneloption `IPFIREWALL_VERBOSE_LIMIT=5` begrenzt die Anzahl gleicher Nachrichten an syslogd(8) für eine gegebene Regel auf fünf Nachrichten. Ist diese Option im Kernel aktiviert, wird nach Erreichen der festgelegten Anzahl die Protokollierung einer (sich unmittelbar hintereinander wiederholenden) Nachricht auf den angegebenen Schwellenwert begrenzt, da beispielsweise die Speicherung von 200 gleichen Protokollnachrichten durch syslogd(8) sinnlos ist. Daher werden durch diesen nur füf derartige Nachrichten protokolliert. Alle weiteren derartigen Nachrichten werden nur gezählt und deren Gesamtzahl wird schließlich von syslogd(8) durch folgenden Ausdruck ausgegeben:

```
last message repeated 45 times
```

Alle protokollierten Nachrichten für Datenpakete werden in der Voreinstellung in die Datei `/var/log/security` (die in der Datei `/etc/syslog.conf` definiert wird), geschrieben.

31.6.5.4. Skripte zur Regeldefinition erstellen

Die meisten fortgeschrittenen IPFW-Nutzer erzeugen eine Datei, die die Regeln für die Firewall enthält, um diese als Skript ausführen zu können. Der Hauptvorteil einer derartigen Konfiguration ist es, dass dadurch mehrere Regeln gleichzeitig geändert und (re-)aktiviert werden können, ohne dass dazu das System neu gestartet werden muss. Dies ist auch beim Testen von Regeländerungen sehr hilfreich. Weil es sich bei der Datei, in der die Regeln gespeichert sind, um ein Skript handelt, ist es auch möglich, häufig verwendete Werte/Befehle durch Aliase zu ersetzen und diese so in mehreren Regeln zu nutzen. Diese Funktion wird im folgenden Beispiel näher vorgestellt.

Die Syntax des folgenden Skripts entspricht der Syntax von sh(1), csh(1) sowie tcsh(1). Felder, die symbolisch substituiert werden, haben das Präfix $ (das Dollarzeichen). Symbolische Felder haben dieses $-Praefix nicht. Der Wert, mit dem das symbolische Feld belegt wird, muss in “doppelten Anführungszeichen” eingeschlossen sein.

Beginnen Sie Ihre Regeldatei wie folgt:

```
############### start of example ipfw rules script #############
#
ipfw -q -f flush       # Delete all rules
# Set defaults
oif="tun0"             # out interface
odns="192.0.2.11"      # ISP’s DNS server IP address
cmd="ipfw -q add "     # build rule prefix
ks="keep-state"        # just too lazy to key this each time
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
################### End of example ipfw rules script ############
```

Die Regeln in diesem Beispiel sind nicht wichtig. Wichtig ist es, zu zeigen, wie die symbolische Substitution innerhalb der Regeln verwendet wird.

Wurde dieses Beispiel in der Datei `/etc/ipfw.rules` gespeichert, so können alle Regeln durch die Ausführung des folgenden Befehls neu geladen werden:

```
# sh /etc/ipfw.rules
```

Statt `/etc/ipfw.rules` können Sie auch einen beliebigen anderen Namen und/oder Speicherort verwenden.

Alternativ könnten Sie die einzelnen Befehle dieses Skripts auch manuell starten:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

31.6.5.5. Zustandsgesteuertes Regelwerk

Das folgende Regelwerk (ohne NAT-Funktionalität) ist ein Beispiel dafür, wie man eine sehr sichere “einschließende” Firewall aufsetzen kann. Eine einschließende Firewall erlaubt es nur Diensten, für die explizite Regeln existieren, die Firewall zu passieren. Alle anderen Dienste und Pakete werden hingegen blockiert. Firewalls, die ganze Netzwerksegmente schützen sollen, benötigen mindestens zwei Netzwerkschnittstellen, für die jeweils eigene Regeln definiert werden müssen, damit die Firewall ordnungsgemäß funktioniert.

Alle unixoiden Betriebssysteme (aber auch solche, die Konzepte aus UNIX implementieren), darunter auch FreeBSD, verwenden die Schnittstelle `lo0` mit der IP-Adresse `127.0.0.1` zur internen Kommunikation mit dem

Betriebssystem. Die Firewall muss so eingestellt sein, dass sie den Datenverkehr dieser speziellen (und nur intern genutzten) Pakete ungehindert durchlässt.

Die Regeln, die den Zugriff auf eingehene und ausgehende Verbindungen regeln, autorisieren und kontrollieren, müssen mit der für die Verbindung zum öffentlichen Internet verantwortlichen Schnittstelle assoziiert werden. Bei dieser Schnittstelle kann es sich beispielsweise um PPP/`tun0` oder die Netzwerkkarte handelt, über, die mit Ihrem DSL- oder Kabelmodem verbunden ist.

Falls mehr als eine Netzwerkkarte mit einem privaten Netzwerk (hinter der Firewall) verbunden sind, müssen die Firewallregeln für alle diese Schnittstellen entstammenden Datenpakete freien und ungehinderten Datenverkehr erlauben.

Es ist sinnvoll, die Regeln in drei Abschnitte aufzuteilen. Der erste Abschnitt enthält die freien, von der Firewall nicht zu überwachenden Netzwerkschnittstellen. Danach folgen die öffentlichen, für den ausgehenden Verkehr verantwortlichen Schnittstellen. Zuletzt kommen dann die Schnittstellen, die für den eingehenden Datenverkehr verantwortlich sind.

Innerhalb der einzelnen Abschnitte ist es sinnvoll, die am häufigsten verwendeten Regeln vor den seltener verwendeten Regel zu platzieren. Jeder Abschnitt sollte mit einer letzten Regel (die alle Pakete, auf die keine Regel zutraf, verwirft) abgeschlossen werden.

Der Abschnitt für den ausgehenden Datenverkehr des folgenden Beispiels enthät nur `allow`)-Regeln, in denen der Dienst, dem der Zugriff auf das öffentliche Internet gewährt wird, eindeutig definiert ist. Alle Regeln verwenden die Optionen `proto`, `port`, `in/out`, `via` sowie `keep state` kodiert. Die Regeln mit `proto tcp` verwenden zusätzlich die Option `setup`, damit die initiale, eine Sitzung beginnende Anfrage identifiziert werden kann, damit die die Zustandsttabelle gefüllt werden kann.

Der Abschnitt für den eingehenden Datenverkehr beginnt mit allen Regeln, die zur Blockierung unerwünschten Datenverkehrs benötigt werden. Für diese Vorgehensweise gibt es zwei Gründe: Zum einen könnten bösartige Pakete legtitimen Datenverker so sehr ähneln, dass sie die Bedingungen von `allow`-Regeln erfüllen und daher die Firewall passieren dürfen. Daher sollten derartige Pakete direkt verworden werden. Zum anderen sollten unerwünschte Pakete mit bekannten (und somit uninteressanten Mustern) sofort ohne Rückmeldung blockiert werden, anstatt erst von der letzten, generischen Regel blockiert (und, was noch wichtiger ist, auch noch protokolliert). Die letzte Regel jedes Abschnittes blockiert und protokolliert; sie kann daher dazu verwendet werden, vor Gericht haltbare Beweise zu erhalten, damit sie gegen Personen vorgehen können, die versuchen, Ihre Systeme anzugreifen.

Achten Sie darauf, dass Sie keine Netwerkantworten für geblockte Pakete senden. Diese müssen ohne Rückmeldung verworfen werden, damit ein Angreifer keine Informationen darüber erhält, ob seine Datenpakete Ihr System erreicht hat. Je weniger Information ein Angreifer über Ihr System erhält, desto sicherer ist Ihr System. Datenpakete an Ports, die nicht bekannten Diensten zugeordnet werden können, können über die Datei `/etc/services` identifiziert werden. Alternativ kann eine Anfrage an http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers Klarheit über die Aufgabe/Funktion einer bestimmten Portnummer bringen. Auf der Seite http://www.sans.org/security-resources/idfaq/oddports.php kann man Information über bekannte Trojaner und von diesen verwendete Portnummern erhalten.

31.6.5.6. Ein Beispiel für einschließende Regeln

Das folgende Regelwerk (ohne NAT-Funktionalität) beschreibt ein vollständiges, einschließendes Regelwerk. Dieses Regelwerk kann direkt auf Ihren eigenen Systemen eingesetzt werden, wenn alle `pass`-Regeln für von Ihnen nicht benötigten Dienste auskommentiert werden. Falls Sie keine Protokollierung benötigen, können Sie diese im Abschnitt für den eingehenden Datenverkehr durch eine `deny` deaktivieren. Die im Beispiel verwendete

Netzwerkschnittstelle `dc0` müssen Sie durch die auf Ihrem System für ausgehenden Datenverkehr vorgesehenen Netzwerkschnittstelle ersetzen. Im Falle von benutzergesteuertem PPPs wäre dies beispielsweise `tun0`.

Alle Regeln folgen einem bestimmten Muster.

- Alle Ausdrücke, die eine Anfrage zum Beginn einer zustandsgesteuerten darstellen, beinhalten den Ausdruck `keep-state`.
- Alle Dienste aus dem öffentlichen Internet beinhalten die Option `limit`, um gegebenenfalls *flooding* zu unterbinden.
- Alle Regeln bezeichnen die Richtung durch der Ausdrücke `in` oder `out`.
- Alle Regeln legen die verwendete Netzwerkschnittstelle die Ausdrücke `via` und *`interface-name`* fest.

Die folgenden Regeln werden in der Datei `/etc/ipfw.rules` definiert.

```
############### Start of IPFW rules file ###############################
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
pif="dc0"     # public interface name of NIC
              # facing the public Internet

########################################################################
# No restrictions on Inside LAN Interface for private network
# Not needed unless you have LAN.
# Change xl0 to your LAN NIC interface name
########################################################################
#$cmd 00005 allow all from any to any via xl0

########################################################################
# No restrictions on Loopback Interface
########################################################################
$cmd 00010 allow all from any to any via lo0

########################################################################
# Allow the packet through if it has previous been added to the
# the "dynamic" rules table by a allow keep-state statement.
########################################################################
$cmd 00015 check-state

########################################################################
# Interface facing Public Internet (Outbound Section)
# Interrogate session start requests originating from behind the
# firewall on the private network or from this gateway server
# destined for the public Internet.
########################################################################

# Allow out access to my ISP's Domain name server.
# x.x.x.x must be the IP address of your ISP.s DNS
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
```

```
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Allow out access to my ISP’s DHCP server for cable/DSL configurations.
# This rule is not needed for .user ppp. connection to the public Internet.
# so you can delete this whole group.
# Use the following rule and check log for IP address.
# Then put IP address in commented out rule & delete first rule
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Allow out non-secure standard www function
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state

# Allow out secure www function https over TLS SSL
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state

# Allow out send & get email function
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state

# Allow out FBSD (make install & CVSUP) functions
# Basically give user root "GOD" privileges.
$cmd 00240 allow tcp from me to any out via $pif setup keep-state uid root

# Allow out ping
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Allow out Time
$cmd 00260 allow tcp from any to any 37 out via $pif setup keep-state

# Allow out nntp news (i.e. news groups)
$cmd 00270 allow tcp from any to any 119 out via $pif setup keep-state

# Allow out secure FTP, Telnet, and SCP
# This function is using SSH (secure shell)
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state

# Allow out whois
$cmd 00290 allow tcp from any to any 43 out via $pif setup keep-state

# deny and log everything else that.s trying to get out.
# This rule enforces the block all by default logic.
$cmd 00299 deny log all from any to any out via $pif

######################################################################
# Interface facing Public Internet (Inbound Section)
# Check packets originating from the public Internet
# destined for this gateway server or the private network.
######################################################################

# Deny all inbound traffic from non-routable reserved address spaces
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif  #RFC 1918 private IP
```

```
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif     #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif          #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif            #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif   #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif         #Class D & E multicast

# Deny public pings
$cmd 00310 deny icmp from any to any in via $pif

# Deny ident
$cmd 00315 deny tcp from any to any 113 in via $pif

# Deny all Netbios service. 137=name, 138=datagram, 139=session
# Netbios is MS/Windows sharing services.
# Block MS/Windows hosts2 name server requests 81
$cmd 00320 deny tcp from any to any 137 in via $pif
$cmd 00321 deny tcp from any to any 138 in via $pif
$cmd 00322 deny tcp from any to any 139 in via $pif
$cmd 00323 deny tcp from any to any 81 in via $pif

# Deny any late arriving packets
$cmd 00330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
$cmd 00332 deny tcp from any to any established in via $pif

# Allow traffic in from ISP’s DHCP server. This rule must contain
# the IP address of your ISP.s DHCP server as it.s the only
# authorized source to send this packet type.
# Only necessary for cable or DSL configurations.
# This rule is not needed for .user ppp. type connection to
# the public Internet. This is the same IP address you captured
# and used in the outbound section.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Allow in standard www function because I have apache server
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow in secure FTP, Telnet, and SCP from public Internet
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID & PW are passed over public
# Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
$cmd 00420 allow tcp from any to me 23 in via $pif setup limit src-addr 2

# Reject & Log all incoming connections from the outside
$cmd 00499 deny log all from any to any in via $pif
```

```
# Everything else is denied by default
# deny and log all packets that fell through to see what they are
$cmd 00999 deny log all from any to any
################ End of IPFW rules file ################################
```

31.6.5.7. Ein Beispiel für zustandshafte NAT-Regeln

Es müssen einige zusätzliche Konfigurationseinstellungen vorgenommen werden, um die die NAT-Funktion von IPFW zu nutzen. Die Kernelquellen müssen mit der Option `IPDIVERT` (im IPFIREWALL-Abschnitt der Kernelkonfigurationsdatei) neu gebaut werden, um den benötigten angepassten Kernel zu erzeugen.

Zusätzlich werden folgende Optionen in der `/etc/rc.conf` benötigt:

```
natd_enable="YES"                   # Enable NATD function
natd_interface="rl0"                # interface name of public Internet NIC
natd_flags="-dynamic -m"            # -m = preserve port numbers if possible
```

Zustandshafte Regeln bei aktiviertem `divert natd` (*Network Address Translation*) verkomplizieren die Formulierung des Regelwerkes beträchtlich. Damit Ihre Firewall funktioniert, kommt es insbesondere auf die Position der Ausdrücke `check-state` sowie `divert natd` an. Sie können nicht länger einen einfachen, kaskadierenden Ablauf verwenden (also einen Regelsatz, bei dem einfach auf eine Regel nach der anderen geprüft wird. Vielmehr wird der neue Aktionstyp `skipto` benötigt. Dies erfordert, dass jede Regel über eine eindeutige Nummer verfügt, um so eindeutige Sprungziele zu erhalten.

Im Folgenden wird anhand eines umkommentierten Beispiels der Paketfluss durch das Regelwerk verdeutlicht.

Die Verarbeitung beginnt mit der ersten Regel (also am Anfang der Regeldatei. Sie setzt sich Regel für Regel weiter fort, bis das Ende der Datei erreicht ist oder eine Regel für das Paket einen Treffer erzielt und das Paket so die Firewall verlassen kann. Achten Sie besonders auf die Position der Regeln mit den Nummern `100, 101, 450, 500` sowie `510`. Diese Regeln steuern die Adressumsetzung ausgehender und eingehender Pakete, so dass deren entsprechende Einträge in der Zustandstabelle immer die private LAN-Adressen abbilden. Zusätzlich werden in allen Regeln die Richtung des Pakets (eingehend oder ausgehend) so die vom Paket zu verwendende Netzwerkschnittstelle definiert. Ausgehende Anfragen, die eine Sitzung starten, rufen immer `skipto rule 500`, damit NAT verwendet werden kann.

Nehmen wir nun an, dass ein Nutzer einen Webbrowser verwendet, um eine Internetseite aufzurufen. Derartige Anfragen werden in der Regel über Port 80 geleitet. Die zugehörigen Pakete werden durch die Firewall verarbeitet. Regel 100 trifft nicht zu, denn das Paket geht nach außen, nicht nach innen. Regel 101 trifft ebenfalls nicht zu, denn es handelt sich um das erste Paket. Folglich wird die Sitzung erst initiiert und kann somit noch nicht in der Zustandstabelle enthalten sein kann. Die erste Regel, die zutrifft, ist Regel 125. Das Paket will das lokale Netzwerk über die Schnittstelle zum öffentlichen Internet (das heißt nach außen) verlassen, es hat aber noch die Quelladresse des privaten lokalen Netzwerks. Da Regel 125 zutrifft, werden zwei Aktionen ausgeführt: Die Option `keep-state` bewirkt, dass das Paket in der internen Tabelle für zustandshafte, dynamische Regeln registriert wird. Danach wird der Aktionsteil der Regel ausgeführt. Dieser ist Bestandteil der Informationen, die in die in der Tabelle für dynamische Regeln aufgenommen wird und lautet `skipto rule 500`. Die Regel 500 führt NATs auf die IP-Adresse des Paketes durch. Danach verlässt das Paket das LAN nach außen in Richtung des öffentlichen Internets. Dieser letzte Teil ist für funktionierendes NAT von entscheidender Bedeutung. Nachdem dieses Paket am Bestimmungsort angekommen ist, wird dort eine Antwort generiert und zurückgeschickt. Dieses Paket wird auf die gleiche Art und Weise durch das gegebene Regelwerk verarbeitet. Dieses Mal trifft Regel 100 auf das Paket zu, damit wird die Bestimmungsadresse auf die zugehörige (lokale) LAN-Adresse (rück-)abgebildet. Danach wird es von der `check-state`-Regel verarbeitet, die Zustandstabelle erkennt, dass eine zugehörige aktive Sitzung vorliegt

und das Paket wird freigegeben und in das LAN geleitet. Es wird innerhalb des LANs von dem PC, der die zugehörige Sitzung hält, empfangen, der ein neues Paket absendet und ein weiteres Datensegment vom entfernten Server anfordert. Dieses Mal wird bei der Prüfung der `check-state`-Regel ein nach außen gehender zugehöriger Eintrag in der Zustandstabelle gefunden und die entsprechende Aktion (also `skipto 500`) wird ausgeführt. Das Paket springt zu Regel 500 und wird durch diese Regel für das öffentliche Internet freigegeben.

Innerhalb des durch die Firewall geschützten Netzwerks werden alle eingehenden Pakete, die zu einer existierenden Sitzung gehören, durch die Regel `check-state` sowie entsprechend platzierte `divert natd`-Regeln verarbeitet. Die notwendige Arbeit beschränkt sich darauf, alle "schlechten" Pakete zu blockieren und nur authorisierten Diensten zugehörige Pakete durchzulassen. In Umkehrung des bisherigen Beispiels nehmen wir nun, dass auf dem Rechner, auf dem die Firewall läuft, auch ein Apache Webserver läuft, auf den von außen, also aus dem öffentlichen Internet, zugegriffen werden kann. Das erste von außen eintreffende Paket (das auch eine neue Sitzung startet) erfüllt Regel 100. Die Zieladresse des Paketes wird daher auf die LAN-Adresse des Firewallrechners abgebildet. Das Paket wird dann weiter auf alle in der Firewall definierten Regeln geprüft und trifft schließlich auf Regel 425. Durch diese Regel werden zwei Aktionen ausgelösst: Erstens wird aus dem Paket eine dynamische Regel generiert und in die Zustandstabelle geschrieben. Zusätzlich wird jedoch die Anzahl neuer Sitzungsanfragen (von der gleichen Quell-IP-Adresse) auf `2` begrenzt, um so DoS-Angriffe auf Dienste, die auf diesem Port laufen, zu verhindern. Die Aktion dieser Regel ist `allow`, daher wird das Paket freigegeben und in das LAN weitergeleitet. Das als Antwort generierte Paket wird durch die `check-state`-Regel als zu einer Sitzung gehörend erkannt. Damit wird es der Regel 500 zugeführt, NAT wird durchgeführt und über die Schnittstelle zum öffentlichen Internet nach außen geroutet.

Beispiel 1 für einen Regelsatz:

```
#!/bin/sh
cmd="ipfw -q add"
skip="skipto 500"
pif=rl0
ks="keep-state"
good_tcpo="22,25,37,43,53,80,443,110,119"

ipfw -q -f flush

$cmd 002 allow all from any to any via xl0  # exclude LAN traffic
$cmd 003 allow all from any to any via lo0  # exclude loopback traffic

$cmd 100 divert natd ip from any to any in via $pif
$cmd 101 check-state

# Authorized outbound packets
$cmd 120 $skip udp from any to xx.168.240.2 53 out via $pif $ks
$cmd 121 $skip udp from any to xx.168.240.5 53 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
$cmd 135 $skip udp from any to any 123 out via $pif $ks


# Deny all inbound traffic from non-routable reserved address spaces
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
```

```
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #reserved for docs
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast

# Authorized inbound packets
$cmd 400 allow udp from xx.70.207.54 to any 68 in $ks
$cmd 420 allow tcp from any to me 80 in via $pif setup limit src-addr 1


$cmd 450 deny log ip from any to any

# This is skipto location for outbound stateful rules
$cmd 500 divert natd ip from any to any out via $pif
$cmd 510 allow ip from any to any

######################### end of rules  ##################
```

Das folgende Beispiel ist praktisch identisch mit dem ersten Regelsatz. Allerdings wurden die Regel umfassend kommentiert und umgeschrieben, damit sie für weniger erfahrene Benutzer leichter verständlich werden.

Beispiel 2 für einen Regelsatz:

```
#!/bin/sh
################ Start of IPFW rules file ###############################
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
skip="skipto 800"
pif="rl0"     # public interface name of NIC
              # facing the public Internet

######################################################################
# No restrictions on Inside LAN Interface for private network
# Change xl0 to your LAN NIC interface name
######################################################################
$cmd 005 allow all from any to any via xl0

######################################################################
# No restrictions on Loopback Interface
######################################################################
$cmd 010 allow all from any to any via lo0

######################################################################
# check if packet is inbound and nat address if it is
######################################################################
$cmd 014 divert natd ip from any to any in via $pif

######################################################################
# Allow the packet through if it has previous been added to the
# the "dynamic" rules table by a allow keep-state statement.
######################################################################
```

```
$cmd 015 check-state

#####################################################################
# Interface facing Public Internet (Outbound Section)
# Check session start requests originating from behind the
# firewall on the private network or from this gateway server
# destined for the public Internet.
#####################################################################

# Allow out access to my ISP's Domain name server.
# x.x.x.x must be the IP address of your ISP's DNS
# Dup these lines if your ISP has more than one DNS server
# Get the IP addresses from /etc/resolv.conf file
$cmd 020 $skip tcp from any to x.x.x.x 53 out via $pif setup keep-state


# Allow out access to my ISP's DHCP server for cable/DSL configurations.
$cmd 030 $skip udp from any to x.x.x.x 67 out via $pif keep-state

# Allow out non-secure standard www function
$cmd 040 $skip tcp from any to any 80 out via $pif setup keep-state

# Allow out secure www function https over TLS SSL
$cmd 050 $skip tcp from any to any 443 out via $pif setup keep-state

# Allow out send & get email function
$cmd 060 $skip tcp from any to any 25 out via $pif setup keep-state
$cmd 061 $skip tcp from any to any 110 out via $pif setup keep-state

# Allow out FreeBSD (make install & CVSUP) functions
# Basically give user root "GOD" privileges.
$cmd 070 $skip tcp from me to any out via $pif setup keep-state uid root

# Allow out ping
$cmd 080 $skip icmp from any to any out via $pif keep-state

# Allow out Time
$cmd 090 $skip tcp from any to any 37 out via $pif setup keep-state

# Allow out nntp news (i.e. news groups)
$cmd 100 $skip tcp from any to any 119 out via $pif setup keep-state

# Allow out secure FTP, Telnet, and SCP
# This function is using SSH (secure shell)
$cmd 110 $skip tcp from any to any 22 out via $pif setup keep-state

# Allow out whois
$cmd 120 $skip tcp from any to any 43 out via $pif setup keep-state

# Allow ntp time server
$cmd 130 $skip udp from any to any 123 out via $pif keep-state

#####################################################################
```

```
# Interface facing Public Internet (Inbound Section)
# Check packets originating from the public Internet
# destined for this gateway server or the private network.
#####################################################################

# Deny all inbound traffic from non-routable reserved address spaces
$cmd 300 deny all from 192.168.0.0/16  to any in via $pif  #RFC 1918 private IP
$cmd 301 deny all from 172.16.0.0/12   to any in via $pif  #RFC 1918 private IP
$cmd 302 deny all from 10.0.0.0/8      to any in via $pif  #RFC 1918 private IP
$cmd 303 deny all from 127.0.0.0/8     to any in via $pif  #loopback
$cmd 304 deny all from 0.0.0.0/8       to any in via $pif  #loopback
$cmd 305 deny all from 169.254.0.0/16  to any in via $pif  #DHCP auto-config
$cmd 306 deny all from 192.0.2.0/24    to any in via $pif  #reserved for docs
$cmd 307 deny all from 204.152.64.0/23 to any in via $pif  #Sun cluster
$cmd 308 deny all from 224.0.0.0/3     to any in via $pif  #Class D & E multicast

# Deny ident
$cmd 315 deny tcp from any to any 113 in via $pif

# Deny all Netbios service. 137=name, 138=datagram, 139=session
# Netbios is MS/Windows sharing services.
# Block MS/Windows hosts2 name server requests 81
$cmd 320 deny tcp from any to any 137 in via $pif
$cmd 321 deny tcp from any to any 138 in via $pif
$cmd 322 deny tcp from any to any 139 in via $pif
$cmd 323 deny tcp from any to any 81  in via $pif

# Deny any late arriving packets
$cmd 330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
$cmd 332 deny tcp from any to any established in via $pif

# Allow traffic in from ISP's DHCP server. This rule must contain
# the IP address of your ISP's DHCP server as it's the only
# authorized source to send this packet type.
# Only necessary for cable or DSL configurations.
# This rule is not needed for 'user ppp' type connection to
# the public Internet. This is the same IP address you captured
# and used in the outbound section.
$cmd 360 allow udp from x.x.x.x to any 68 in via $pif keep-state

# Allow in standard www function because I have Apache server
$cmd 370 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow in secure FTP, Telnet, and SCP from public Internet
$cmd 380 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Allow in non-secure Telnet session from public Internet
# labeled non-secure because ID & PW are passed over public
# Internet as clear text.
# Delete this sample group if you do not have telnet server enabled.
$cmd 390 allow tcp from any to me 23 in via $pif setup limit src-addr 2
```

```
# Reject & Log all unauthorized incoming connections from the public Internet
$cmd 400 deny log all from any to any in via $pif

# Reject & Log all unauthorized out going connections to the public Internet
$cmd 450 deny log all from any to any out via $pif

# This is skipto location for outbound stateful rules
$cmd 800 divert natd ip from any to any out via $pif
$cmd 801 allow ip from any to any

# Everything else is denied by default
# deny and log all packets that fell through to see what they are
$cmd 999 deny log all from any to any
################ End of IPFW rules file ################################
```

Kapitel 32. Weiterführende Netzwerkthemen

Übersetzt von Johann Kois.

32.1. Übersicht

Dieses Kapitel beschreibt verschiedene weiterführende Netzwerkthemen.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Die Grundlagen von Gateways und Routen kennen.
- Bluetooth- sowie drahtlose, der Norm IEEE® 802.11 entsprechende, Geräte mit FreeBSD verwenden können.
- Eine Bridge unter FreeBSD einrichten können.
- Einen plattenlosen Rechner über das Netzwerk starten können.
- Wissen, wie man NAT (Network Address Translation) einrichtet.
- Zwei Computer über PLIP verbinden können.
- IPv6 auf einem FreeBSD-Rechner einrichten können.
- ATM einrichten können.
- CARP, das Common Address Redundancy Protocol, unter FreeBSD einsetzen können.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Grundlagen der `/etc/rc`-Skripte verstanden haben.
- Mit der grundlegenden Netzwerkterminologie vertraut sein.
- Einen neuen FreeBSD-Kernel konfigurieren und installieren können (Kapitel 9).
- Wissen, wie man zusätzliche Softwarepakete von Drittherstellern installiert (Kapitel 5).

32.2. Gateways und Routen

Beigetragen von Coranth Gryphon.

Damit ein Rechner einen anderen über ein Netzwerk finden kann, muss ein Mechanismus vorhanden sein, der beschreibt, wie man von einem Rechner zum anderen gelangt. Dieser Vorgang wird als *Routing* bezeichnet. Eine “Route” besteht aus einem definierten Adressenpaar: Einem “Ziel” und einem “Gateway”. Dieses Paar zeigt an, dass Sie über das *Gateway* zum *Ziel* gelangen wollen. Es gibt drei Arten von Zielen: Einzelne Rechner (Hosts), Subnetze und das “Standard”ziel. Die “Standardroute” wird verwendet, wenn keine andere Route zutrifft. Wir werden Standardrouten später etwas genauer behandeln. Außerdem gibt es drei Arten von Gateways: Einzelne Rechner (Hosts), Schnittstellen (Interfaces, auch als “Links” bezeichnet), sowie Ethernet Hardware-Adressen (MAC-Adressen).

32.2.1. Ein Beispiel

Um die verschiedenen Aspekte des Routings zu veranschaulichen, verwenden wir folgende Ausgaben von `netstat`:

```
% netstat -r
Routing tables

Destination              Gateway            Flags       Refs     Use     Netif Expire

default                  outside-gw         UGSc        37       418     ppp0
localhost                localhost          UH          0        181     lo0
test0                    0:e0:b5:36:cf:4f   UHLW        5        63288   ed0    77
10.20.30.255             link#1             UHLW        1        2421
example.com              link#1             UC          0        0
host1                    0:e0:a8:37:8:1e    UHLW        3        4601    lo0
host2                    0:e0:a8:37:8:1e    UHLW        0        5       lo0 =>
host2.example.com        link#1             UC          0        0
224                      link#1             UC          0        0
```

Die ersten zwei Zeilen geben die Standardroute (die wir im nächsten Abschnitt behandeln), sowie die `localhost` Route an.

Das in der Routingtabelle für `localhost` festgelegte Interface (`Netif`-Spalte) `lo0`, ist auch als loopback-Gerät (Prüfschleife) bekannt. Das heißt, dass der ganze Datenverkehr für dieses Ziel intern (innerhalb des Gerätes) bleibt, anstatt ihn über ein Netzwerk (LAN) zu versenden, da das Ziel dem Start entspricht.

Der nächste auffällige Punkt sind die mit `0:e0:` beginnenden Adressen. Es handelt sich dabei um Ethernet Hardwareadressen, die auch als MAC-Adressen bekannt sind. FreeBSD identifiziert Rechner im lokalen Netz automatisch (im Beispiel `test0`) und fügt eine direkte Route zu diesem Rechner hinzu. Dies passiert über die Ethernet-Schnittstelle `ed0`. Außerdem existiert ein Timeout (in der Spalte `Expire`) für diese Art von Routen, der verwendet wird, wenn dieser Rechner in einem definierten Zeitraum nicht reagiert. Wenn dies passiert, wird die Route zu diesem Rechner automatisch gelöscht. Rechner im lokalen Netz werden durch einen als RIP (Routing Information Protocol) bezeichneten Mechanismus identifiziert, der den kürzesten Weg zu den jeweiligen Rechnern bestimmt.

FreeBSD fügt außerdem Subnetzrouten für das lokale Subnetz hinzu (`10.20.30.255` ist die Broadcast-Adresse für das Subnetz `10.20.30`, `example.com` ist der zu diesem Subnetz gehörige Domainname). Das Ziel `link#1` bezieht sich auf die erste Ethernet-Karte im Rechner. Sie können auch feststellen, dass keine zusätzlichen Schnittstellen angegeben sind.

Routen für Rechner im lokalen Netz und lokale Subnetze werden automatisch durch den **routed** Daemon konfiguriert. Ist dieser nicht gestartet, sind nur statisch definierte (explizit eingegebene) Routen vorhanden.

Die Zeile `host1` bezieht sich auf unseren Rechner, der durch seine Ethernetadresse bekannt ist. Da unser Rechner der Sender ist, verwendet FreeBSD automatisch das Loopback-Gerät (`lo0`), anstatt den Datenverkehr über die Ethernetschnittstelle zu senden.

Die zwei `host2` Zeilen sind ein Beispiel dafür, was passiert, wenn wir ein ifconfig(8) Alias verwenden (Lesen Sie dazu den Abschnitt über Ethernet, wenn Sie wissen wollen, warum wir das tun sollten.). Das Symbol `=>` (nach der `lo0`-Schnittstelle) sagt aus, dass wir nicht nur das Loopbackgerät verwenden (da sich die Adresse auf den lokalen Rechner bezieht), sondern dass es sich zusätzlich auch um ein Alias handelt. Solche Routen sind nur auf Rechnern vorhanden, die den Alias bereitstellen; alle anderen Rechner im lokalen Netz haben für solche Routen nur eine einfache `link#1` Zeile.

Die letzte Zeile (Zielsubnetz `224`) behandelt das Multicasting, das wir in einem anderen Abschnitt besprechen werden.

Schließlich gibt es für Routen noch verschiedene Attribute, die Sie in der Spalte `Flags` finden. Nachfolgend finden Sie eine kurze Übersicht von einigen dieser Flags und ihrer Bedeutung:

U	Up: Die Route ist aktiv.
H	Host: Das Ziel der Route ist ein einzelner Rechner (Host).
G	Gateway: Alle Daten, die an dieses Ziel gesendet werden, werden von diesem System an ihr jeweiliges Ziel weitergeleitet.
S	Static: Diese Route wurde manuell konfiguriert, das heißt sie wurde *nicht* automatisch vom System erzeugt.
C	Clone: Erzeugt eine neue Route, basierend auf der Route für den Rechner, mit dem wir uns verbinden. Diese Routenart wird normalerweise für lokale Netzwerke verwendet.
W	WasCloned: Eine Route, die automatisch konfiguriert wurde. Sie basiert auf einer lokalen Netzwerkroute (Clone).
L	Link: Die Route beinhaltet einen Verweis auf eine Ethernetkarte (MAC-Adresse).

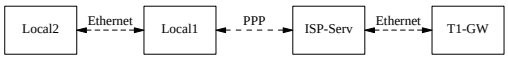
32.2.2. Standardrouten

Wenn sich der lokale Rechner mit einem entfernten Rechner verbinden will, wird die Routingtabelle überprüft, um festzustellen, ob bereits ein bekannter Pfad vorhanden ist. Gehört dieser entfernte Rechner zu einem Subnetz, dessen Pfad uns bereits bekannt ist (*Cloned route*), dann versucht der lokale Rechner über diese Schnittstelle eine Verbindung herzustellen.

Wenn alle bekannten Pfade nicht funktionieren, hat der lokale Rechner eine letzte Möglichkeit: Die Standardroute (Defaultroute). Bei dieser Route handelt es sich um eine spezielle Gateway-Route (gewöhnlich die einzige im System vorhandene), die im Flags-Feld immer mit `C` gekennzeichnet ist. Für Rechner im lokalen Netzwerk ist dieses Gateway auf *welcher Rechner auch immer eine Verbindung nach außen hat* gesetzt (entweder über eine PPP-Verbindung, DSL, ein Kabelmodem, T1 oder eine beliebige andere Netzwerkverbindung).

Wenn Sie die Standardroute für einen Rechner konfigurieren, der selbst als Gateway zur Außenwelt funktioniert, wird die Standardroute zum Gateway-Rechner Ihres Internetanbieter (ISP) gesetzt.

Sehen wir uns ein Beispiel für Standardrouten an. So sieht eine übliche Konfiguration aus:



Die Rechner `Local1` und `Local2` befinden sich auf Ihrer Seite. `Local1` ist mit einem ISP über eine PPP-Verbindung verbunden. Dieser PPP-Server ist über ein lokales Netzwerk mit einem anderen Gateway-Rechner verbunden, der über eine Schnittstelle die Verbindung des ISP zum Internet herstellt.

Die Standardrouten für Ihre Maschinen lauten:

Host	Standard Gateway	Schnittstelle
Local2	Local1	Ethernet
Local1	T1-GW	PPP

Eine häufig gestellte Frage lautet: “Warum (oder wie) sollten wir `T1-GW` als Standard-Gateway für `Local1` setzen, statt den (direkt verbundenen) ISP-Server zu verwenden?”.

Bedenken Sie, dass die PPP-Schnittstelle für die Verbindung eine Adresse des lokalen Netzes des ISP verwendet. Daher werden Routen für alle anderen Rechner im lokalen Netz des ISP automatisch erzeugt. Daraus folgt, dass Sie bereits wissen, wie Sie `T1-GW` erreichen können! Es ist also unnötig, einen Zwischenschritt über den ISP-Server zu machen.

Es ist üblich, die Adresse `X.X.X.1` als Gateway-Adresse für ihr lokales Netzwerk zu verwenden. Für unser Beispiel bedeutet dies Folgendes: Wenn Ihr lokaler Klasse-C-Adressraum `10.20.30` ist und Ihr ISP `10.9.9` verwendet, sehen die Standardrouten so aus:

Rechner (Host)	Standardroute
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

Sie können die Standardroute ganz einfach in der Datei `/etc/rc.conf` festlegen. In unserem Beispiel wurde auf dem Rechner `Local2` folgende Zeile in `/etc/rc.conf` eingefügt:

```
defaultrouter="10.20.30.1"
```

Die Standardroute kann über route(8) auch direkt gesetzt werden:

```
# route add default 10.20.30.1
```

Weitere Informationen zum Bearbeiten von Netzwerkroutingtabellen finden Sie in route(8).

32.2.3. Rechner mit zwei Heimatnetzen

Es gibt noch eine Konfigurationsmöglichkeit, die wir besprechen sollten, und zwar Rechner, die sich in zwei Netzwerken befinden. Technisch gesehen, zählt jeder als Gateway arbeitende Rechner zu den Rechnern mit zwei Heimatnetzen (im obigen Beispiel unter Verwendung einer PPP-Verbindung). In der Praxis meint man damit allerdings nur Rechner, die sich in zwei lokalen Netzen befinden.

Entweder verfügt der Rechner über zwei Ethernetkarten und jede dieser Karten hat eine Adresse in einem separaten Subnetz, oder der Rechner hat nur eine Ethernetkarte und verwendet ifconfig(8) Aliasing. Die erste Möglichkeit wird verwendet, wenn zwei physikalisch getrennte Ethernet-Netzwerke vorhanden sind, die zweite, wenn es nur ein physikalisches Ethernet-Netzwerk gibt, das aber aus zwei logisch getrennten Subnetzen besteht.

In beiden Fällen werden Routingtabellen erstellt, damit jedes Subnetz weiß, dass dieser Rechner als Gateway zum anderen Subnetz arbeitet (*inbound route*). Diese Konfiguration (der Gateway-Rechner arbeitet als Router zwischen den Subnetzen) wird häufig verwendet, wenn es darum geht, Paketfilterung oder eine Firewall (in eine oder beide Richtungen) zu implementieren.

Soll dieser Rechner Pakete zwischen den beiden Schnittstellen weiterleiten, müssen Sie diese Funktion manuell konfigurieren und aktivieren. Lesen Sie den nächsten Abschnitt, wenn Sie weitere Informationen zu diesem Thema benötigen.

32.2.4. Einen Router konfigurieren

Ein Netzwerkrouter ist einfach ein System, das Pakete von einer Schnittstelle zur anderen weiterleitet. Internetstandards und gute Ingenieurspraxis sorgten dafür, dass diese Funktion in FreeBSD in der Voreinstellung deaktiviert ist. Sie können diese Funktion aktivieren, indem Sie in rc.conf(5) folgende Änderung durchführen:

```
gateway_enable="YES"          # Auf YES setzen, wenn der Rechner als Gateway arbeiten soll
```

Diese Option setzt die sysctl(8)-Variable `net.inet.ip.forwarding` auf `1`. Wenn Sie das Routing kurzzeitig unterbrechen wollen, können Sie die Variable auf `0` setzen.

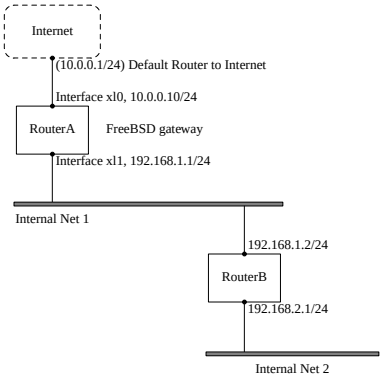
Ihr neuer Router benötigt nun noch Routen, um zu wissen, wohin er den Verkehr senden soll. Haben Sie ein (sehr) einfaches Netzwerk, können Sie statische Routen verwenden. FreeBSD verfügt über den Standard BSD-Routing-Daemon routed(8), der RIP (sowohl Version 1 als auch Version 2) und IRDP versteht. BGP v4, OSPF v2 und andere Protokolle werden von `net/zebra` unterstützt. Es stehen auch kommerzielle Produkte wie **gated** zur Verfügung.

32.2.5. Statische Routen einrichten

Beigetragen von Al Hoang.

32.2.5.1. Manuelle Konfiguration

Nehmen wir an, dass wir über folgendes Netzwerk verfügen:



`RouterA`, ein FreeBSD-Rechner, dient als Router für den Zugriff auf das Internet. Die Standardroute ist auf `10.0.0.1` gesetzt, damit ein Zugriff auf das Internet möglich wird. Wir nehmen nun an, dass `RouterB` bereits konfiguriert ist und daher weiß, wie er andere Rechner erreichen kann. Dazu wird die Standardroute von `RouterB` auf `192.168.1.1` gesetzt, da dieser Rechner als Gateway fungiert.

Sieht man sich die Routingtabelle für `RouterA` an, erhält man folgende Ausgabe:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0/24          link#1             UC          0        0    xl0
```

```
192.168.1/24       link#2             UC          0        0    xl1
```

Mit dieser Routingtabelle kann `RouterA` unser internes Netz 2 nicht erreichen, da keine Route zum Rechner `192.168.2.0/24` vorhanden ist. Um dies zu korrigieren, kann die Route manuell gesetzt werden. Durch den folgenden Befehl wird das interne Netz 2 in die Routingtabelle des Rechners `RouterA` aufgenommen, indem `192.168.1.2` als nächster Zwischenschritt verwenden wird:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Ab sofort kann `RouterA` alle Rechner des Netzwerks `192.168.2.0/24` erreichen.

32.2.5.2. Routen dauerhaft einrichten

Das obige Beispiel ist für die Konfiguration einer statischen Route auf einem laufenden System geeignet. Diese Information geht jedoch verloren, wenn der FreeBSD-Rechner neu gestartet werden muss. Um dies zu verhindern, wird diese Route in `/etc/rc.conf` eingetragen:

```
# Add Internal Net 2 as a static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

Die Variable `static_routes` enthält eine Reihe von Strings, die durch Leerzeichen getrennt sind. Jeder String bezieht sich auf den Namen einer Route. In unserem Beispiel hat `static_routes` *`internalnet2`* als einzigen String. Zusätzlich verwendet man die Konfigurationsvariable `route_`*`internalnet2`*, in der alle sonstigen an route(8) zu übergebenden Parameter festgelegt werden. In obigen Beispiel hätte man folgenden Befehl verwendet:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Daher wird `"-net 192.168.2.0/24 192.168.1.2"` als Parameter der Variable `route_` angegeben.

Wie bereits erwähnt, können bei `static_routes` auch mehrere Strings angegeben werden. Dadurch lassen sich mehrere statische Routen anlegen. Durch folgende Zeilen werden auf einem imaginären Rechner statische Routen zu den Netzwerken `192.168.0.0/24` sowie `192.168.1.0/24` definiert:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

32.2.6. Verteilung von Routing-Informationen

Wir haben bereits darüber gesprochen, wie wir unsere Routen zur Außenwelt definieren, aber nicht darüber, wie die Außenwelt uns finden kann.

Wir wissen bereits, dass Routing-Tabellen so erstellt werden können, dass sämtlicher Verkehr für einen bestimmten Adressraum (in unserem Beispiel ein Klasse-C-Subnetz) zu einem bestimmten Rechner in diesem Netzwerk gesendet wird, der die eingehenden Pakete im Subnetz verteilt.

Wenn Sie einen Adressraum für Ihre Seite zugewiesen bekommen, richtet Ihr Diensteanbieter seine Routingtabellen so ein, dass der ganze Verkehr für Ihr Subnetz entlang Ihrer PPP-Verbindung zu Ihrer Seite gesendet wird. Aber woher wissen die Seiten in der Außenwelt, dass sie die Daten an Ihren ISP senden sollen?

Es gibt ein System (ähnlich dem verbreiteten DNS), das alle zugewiesenen Adressräume verwaltet und ihre Verbindung zum Internet-Backbone definiert und dokumentiert. Der “Backbone” ist das Netz aus Hauptverbindungen, die den Internetverkehr in der ganzen Welt transportieren und verteilen. Jeder Backbone-Rechner verfügt über eine Kopie von Haupttabellen, die den Verkehr für ein bestimmtes Netzwerk hierarchisch vom Backbone über eine Kette von Diensteanbietern bis hin zu Ihrer Seite leiten.

Es ist die Aufgabe Ihres Diensteanbieters, den Backbone-Seiten mitzuteilen, dass sie mit Ihrer Seite verbunden wurden. Durch diese Mitteilung der Route ist nun auch der Weg zu Ihnen bekannt. Dieser Vorgang wird als *Bekanntmachung von Routen* (*routing propagation*) bezeichnet.

32.2.7. Problembehebung

Manchmal kommt es zu Problemen bei der Bekanntmachung von Routen, und einige Seiten sind nicht in der Lage, Sie zu erreichen. Vielleicht der nützlichste Befehl, um festzustellen, wo das Routing nicht funktioniert, ist traceroute(8). Er ist außerdem sehr nützlich, wenn Sie einen entfernten Rechner nicht erreichen können (lesen Sie dazu auch ping(8)).

traceroute(8) wird mit dem zu erreichenden Rechner (Host) ausgeführt. Angezeigt werden die Gateway-Rechner entlang des Verbindungspfades. Schließlich wird der Zielrechner erreicht oder es kommt zu einem Verbindungsabbruch (beispielsweise durch Nichterreichbarkeit eines Gateway-Rechners).

Weitere Informationen finden Sie in traceroute(8).

32.2.8. Multicast-Routing

FreeBSD unterstützt sowohl Multicast-Anwendungen als auch Multicast-Routing. Multicast-Anwendungen müssen nicht konfiguriert werden, sie laufen einfach. Multicast-Routing muss in der Kernelkonfiguration aktiviert werden:

```
options MROUTING
```

Zusätzlich muss mrouted(8), der Multicast-Routing-Daemon, über die Datei `/etc/mrouted.conf` eingerichtet werden, um Tunnel und DVMRP zu aktivieren. Weitere Informationen zu diesem Thema finden Sie in mrouted(8).

> **Anmerkung:** mrouted(8), der Multicast Routing Daemon, verwendet das DVMRP Multicast Routing Protocol, das inzwischen in den meisten Multicast-Installationen durch pim(4) ersetzt wurde. mrouted(8) sowie die damit in Verbindung stehenden Werkzeuge map-mbone(8) und mrinfo(8)können über die FreeBSD-Ports-Sammlung (genauer den Port `net/mrouted`) installiert werden.

32.3. Drahtlose Netzwerke

Loader, Marc Fonvieille und Murray Stokely.

32.3.1. Grundlagen

Die meisten drahtlosen Netzwerke basieren auf dem Standard IEEE 802.11. Sie bestehen aus Stationen, die in der Regel im 2,4 GHz- oder im 5 GHz-Band miteinander kommunizieren. Es ist aber auch möglich, dass regional andere

Frequenzen, beispielsweise im 2,3 GHz- oder 4,9 GHz-Band, verwendet werden.

802.11-Netzwerke können auf zwei verschiedene Arten aufgebaut sein: Im *Infrastruktur-Modus* agiert eine Station als Master, mit dem sich alle anderen Stationen verbinden. Die Summe aller Stationen wird als BSS (Basic Service Set), die Master-Station hingegen als Access Point (AP) bezeichnet. In einem BSS läuft jedwede Kommunikation über den Access Point. Die zweite Form drahtloser Netzwerke sind die sogenannten *Ad-hoc-Netzwerke* (auch als IBSS bezeichnet), in denen es keinen Access Point gibt und in denen die Stationen direkt miteinander kommunizieren.

Die ersten 802.11-Netzwerke arbeiteten im 2,4 GHz-Band und nutzten dazu Protokolle der IEEE-Standards 802.11 sowie 802.11b. Diese Standards legen unter anderem Betriebsfrequenzen sowie Merkmale des MAC-Layers (wie Frames und Transmissionsraten) fest. Später kam der Standard 802.11a hinzu, der im 5 GHz-Band, im Gegensatz zu den ersten beiden Standards aber mit unterschiedlichen Signalmechanismen und höheren Transmissionsraten arbeitet. Der neueste Standard 802.11g implementiert die Signal- und Transmissionsmechanismen von 802.11a im 2,4 GHz-Band, ist dabei aber abwärtskompatibel zu 802.11b-Netzwerken.

Unabhängig von den zugrundeliegenden Transportmechanismen verfügen 802.11-Netzwerke über diverse Sicherheitsmechanismen. Der ursprüngliche 802.11-Standard definierte lediglich ein einfaches Sicherheitsprotokoll namens WEP. Dieses Protokoll verwendet einen fixen (gemeinsam verwendeten) Schlüssel sowie die RC4-Kryptografie-Chiffre, um Daten verschlüsselt über das drahtlose Netzwerk zu senden. Alle Stationen des Netzwerks müssen sich auf den gleichen fixen Schlüssel einigen, um miteinander kommunizieren zu können. Dieses Schema ist sehr leicht zu knacken und wird deshalb heute kaum mehr eingesetzt. Aktuelle Sicherheitsmechanismen bauen auf dem Standard IEEE 802.11i auf, der neue kryptografische Schlüssel (Chiffren), ein neues Protokoll für die Anmeldung von Stationen an einem Access Point, sowie Mechanismen zum Austausch von Schlüsseln als Vorbereitung der Kommunikation zwischen verschiedenen Geräten festlegt. Kryptografische Schlüssel werden regelmäßig getauscht. Außerdem gibt es Mechanismen, um Einbruchsversuche zu entdecken (und Gegenmaßnahmen ergreifen zu können). Ein weiteres häufig verwendetes Sicherheitsprotokoll ist WPA. Dabei handelt es sich um einen Vorläufer von 802.11i, der von einem Industriekonsortium als Zwischenlösung bis zur endgültigen Verabschiedung von 802.11i entwickelt wurde. WPA definiert eine Untergruppe der Anforderungen des 802.11i-Standards und ist für den Einsatz in älterer Hardware vorgesehen. WPA benötigt nur den (auf dem ursprünglichen WEP-Code basierenden) TKIP-Chiffre. 802.11i erlaubt zwar auch die Verwendung von TKIP, fordert aber zusätzlich eine stärkere Chiffre (AES-CCM) für die Datenverschlüsselung. (AES war für WPA nicht vorgesehen, weil man es als zu rechenintensiv für den Einsatz in älteren Geräten ansah.)

Neben den weiter oben erwähnten Standards ist auch der Standard 802.11e von großer Bedeutung. Dieser definiert Protokolle zur Übertragung von Multimedia-Anwendungen wie das Streaming von Videodateien oder Voice-over-IP (VoIP) in einem 802.11-Netzwerk. Analog zu 802.11i verfügt auch 802.11e über eine vorläufige Spezifikation namens WMM (ursprünglich WME), die von einem Industriekonsortium als Untergruppe von 802.11e spezifiziert wurde, um Multimedia-Anwendungen bereits vor der endgültigen Verabschiedung des 802.11e-Standards implementieren zu können. 802.11e sowie WME/WMM erlauben eine Prioritätenvergabe beim Datentransfer im einem drahtlosen Netzwerk. Möglich wird dies durch den Einsatz von Quality of Service-Protokollen (QoS) und erweiterten Medienzugriffsprotokollen. Werden diese Protokolle korrekt implementiert, erlauben sie daher hohe Datenübertragungsraten und einen priorisierten Datenfluss.

FreeBSD unterstützt die Standards 802.11a, 802.11b, sowie 802.11g. Ebenfalls unterstützt werden WPA sowie die Sicherheitsprotokolle gemäß 802.11i (dies sowohl für 11a, 11b als auch 11g). QoS und Verkehrspriorisierung, die von den WME/WMM-Protokollen benötigt werden, werden ebenfalls (allerdings nicht für alle drahtlosen Geräte) unterstützt.

32.3.2. Basiskonfiguration

32.3.2.1. Kernelkonfiguration

Um ein drahtloses Netzwerk zu nutzen, benötigen Sie eine drahtlose Netzwerkkarte und einen Kernel, der drahtlose Netzwerke unterstützt. Der FreeBSD-Kernel unterstützt den Einsatz von Kernelmodulen. Daher müssen Sie nur die Unterstützung für die von Ihnen verwendeten Geräte aktivieren.

Als Erstes benötigen Sie ein drahtloses Gerät. Die meisten drahtlosen Geräte verwenden Bauteile von Atheros und werden deshalb vom ath(4)-Treiber unterstützt. Um diesen Treiber zu verwenden, nehmen Sie die folgende Zeile in die Datei `/boot/loader.conf` auf:

```
if_ath_load="YES"
```

Der Atheros-Treiber besteht aus drei Teilen: dem Treiber selbst (ath(4)), dem Hardware-Support-Layer für die chip-spezifischen Funktionen (ath_hal(4)) sowie einem Algorithmus zur Auswahl der korrekten Frame-Übertragungsrate (ath_rate_sample). Wenn Sie die Unterstützung für diesen Treiber als Kernelmodul laden, kümmert sich dieses automatisch um diese Aufgaben. Verwenden Sie ein Nicht-Atheros-Gerät, so müssen Sie hingegen das für dieses Gerät geeignete Modul laden, beispielsweise

```
if_wi_load="YES"
```

für Geräte, die auf Bauteilen von Intersil Prism basieren und daher den Treiber wi(4) voraussetzen.

> **Anmerkung:** In den folgenden Abschnitten wird der ath(4)-Treiber verwendet. Verwenden Sie ein anderes Gerät, müssen Sie diesen Wert daher an Ihre Konfiguration anpassen. Eine Liste aller verfügbaren Treiber und unterstützten drahtlosen Geräte finden sich in den FreeBSD Hardware Notes. Diese sind für verschiedene Releases und Architekturen auf der Seite Release Information (http://www.FreeBSD.org/releases/index.html) der FreeBSD Homepage. Gibt es keinen nativen FreeBSD-Treiber für Ihr drahtloses Gerät, können Sie möglicherweise mit NDIS einen Windows-Treiber verwenden.

Unter FreeBSD 7.X benötigen Sie zusätlich zum korrekten Treiber auch die Unterstützung für 802.11-Netzwerke. Für den ath(4)-Treiber werden dazu mindestens die Module wlan(4), `wlan_scan_ap` sowie `wlan_scan_sta` benötigt. Das wlan(4)-Kernelmodul wird automatisch mit dem Treiber des drahtlosen Geräts geladen, die beiden anderen Module werden jeweils durch einen Eintrag in der Datei `/boot/loader.conf` beim Systemstart geladen:

```
wlan_scan_ap_load="YES"
wlan_scan_sta_load="YES"
```

Ab FreeBSD 8.0 sind diese Module Teil des wlan(4)-Treibers und werden bei Bedarf automatisch geladen.

Zusätzlich benötigen Sie noch Module zur Verschlüsselung ihres drahtlosen Netzwerks. Diese werden normalerweise dynamisch vom wlan(4)-Modul geladen. Im folgenden Beispiel erfolgt allerdings eine manuelle Konfiguration. Folgende Module sind verfügbar: wlan_wep(4), wlan_ccmp(4) sowie wlan_tkip(4). Sowohl wlan_ccmp(4) als auch wlan_tkip(4) werden nur benötigt, wenn Sie WPA und/oder die Sicherheitsprotokolle von 802.11i verwenden wollen. Wollen Sie Ihr Netzwerk hingegen ohne Verschlüsselung betreiben, benötigen Sie nicht einmal die wlan_wep(4)-Unterstützung. Um alle drei Module beim Systemstart zu laden, fügen Sie folgende Zeilen in die Datei `/boot/loader.conf` ein:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
```

```
wlan_tkip_load="YES"
```

Um diese neuen Einträge in der Datei `/boot/loader.conf` zu aktivieren, müssen Sie Ihr FreeBSD-System neu starten. Alternativ können Sie die Kernelmodule aber auch manuell mit kldload(8) laden.

> **Anmerkung:** Wollen Sie keine Kernelmodule verwenden, können Sie die benötigten Treiber auch in Ihren Kernel kompilieren. Daz nehmen Sie folgende Zeilen in Ihre Kernelkonfigurationsdatei auf:
>
> ```
> device wlan              # 802.11 support
> device wlan_wep          # 802.11 WEP support
> device wlan_ccmp         # 802.11 CCMP support
> device wlan_tkip         # 802.11 TKIP support
> device wlan_amrr         # AMRR transmit rate control algorithm
> device ath               # Atheros pci/cardbus NIC's
> device ath_hal           # pci/cardbus chip support
> options AH_SUPPORT_AR5416 # enable AR5416 tx/rx descriptors
> device ath_rate_sample   # SampleRate tx rate control for ath
> ```
>
> Verwenden Sie FreeBSD 7.X, müssen Sie auch die beiden Module `wlan_scan_ap` und `wlan_scan_sta` in den Kernel aufnehmen (unter FreeBSD 8.X ist dies hingegen nicht mehr notwendig):
>
> ```
> device wlan_scan_ap      # 802.11 AP mode scanning
> device wlan_scan_sta     # 802.11 STA mode scanning
> ```
>
> Danach bauen Sie den neuen Kernel und starten Ihr FreeBSD-System neu.

Während des Systemstarts sollten nun einige Informationen ähnlich den folgenden über das von Ihnen verwendete drahtlose Gerät ausgegeben werden:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```

32.3.3. Infrastruktur-Modus

Drahtlose Netzwerke werden in der Regel im Infrastruktur-Modus (auch BSS-Modus genannt) betrieben. Dazu werden mehrere drahtlose Access Points zu einem gemeinsamen drahtlosen Netzwerk verbunden. Jedes dieser drahtlosen Netzwerke hat einen eigenen Namen, der als *SSID* bezeichnet wird. Alle Clients eines drahtlosen Netzwerks verbinden sich in diesem Modus mit einem Access Point.

32.3.3.1. FreeBSD-Clients

32.3.3.1.1. Einen Access Point finden

Um nach drahtlosen Netzwerken zu suchen verwenden Sie `ifconfig`. Dieser Scanvorgang kann einige Zei in Anspruch nehmen, da dazu jede verfügbare Frequenz auf verfügbare Access Points hin überprüft werden muss. Um die Suche zu starten, müssen Sie als Super-User angemeldet sein:

```
# ifconfig wlan0 create wlandev ath0
```

```
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
dlinkap         00:13:46:49:41:76   11   54M -90:96   100 EPS  WPA WME
freebsdap       00:11:95:c3:0d:ac    1   54M -83:96   100 EPS  WPA
```

Anmerkung: Ihre Netzwerkkarte muss in den Status `up` versetzt werden, bevor Sie den ersten Scanvorgang starten können. Für spätere Scans ist dies aber nicht mehr erforderlich.

Anmerkung: Unter FreeBSD 7.X wird der Gerätetreiber, beispielsweise *`ath0`*, direkt verwendet, anstatt auf das allgemeine Gerät *`wlan0`* zuzugreifen. Verwenden Sie also FreeBSD 7.X, müssen Sie die beiden Befehle im vorigen Beispiel durch den folgenden Befehl ersetzen:

```
# ifconfig ath0 up scan
```

Dies gilt auch für alle weiteren Ausführungen in diesem Kapitel. Unter FreeBSD 7.X müssen analog alle Befehle und Konfigurationsdateien/Zeilen entsprechend angepasst werden.

Als Ergebnis erhalten Sie eine Liste mit allen gefundenen BSS/IBSS-Netzwerken. Zusätzlich zur `SSID` (dem Namen des Netzwerks) wird auch die `BSSID` ausgegeben. Dabei handelt es sich um MAC-Adresse des Access Points. Das Feld `CAPS` gibt den Typ des Netzwerks sowie die Fähigkeiten der Stationen innerhalb des Netzwerks an:

`E`

Extended Service Set (ESS). Zeigt an, dass die Station Teil eines Infrastruktur-Netzwerks ist (und nicht eines IBSS/Ad-hoc-Netzwerks).

`I`

IBSS/Ad-hoc-Netzwerk. Die Station ist Teil eines Ad-hoc-Netzwerks (und nicht eines ESS-Netzwerks).

`P`

Privacy. Alle Datenframes, die innerhalb des BSS ausgetauscht werden, sind verschlüsselt. Dieses BSS verwendet dazu kryptografische Verfahren wie WEP, TKIP oder AES-CCMP.

`S`

Short Preamble. Das Netzwerk verwendet eine kurze Präambel (definiert in 802.11b High Rate/DSSS PHY). Eine kurze Präambel verwendet ein 56 Bit langes Sync-Feld (im Gegensatz zu einer langen Präambel, die ein 128 Bit langes Sync-Feld verwendet).

`s`

Short slot time. Das 802.11g-Netzwerk verwendet eine kurze Slotzeit, da es in diesem Netzwerk keine veralteten (802.11b) Geräte gibt.

Um eine Liste der bekannten Netzwerke auszugeben, verwenden Sie den folgenden Befehl:

```
# ifconfig wlan0 list scan
```

Diese Liste kann entweder automatisch durch das drahtlose Gerät oder manuell durch eine `scan`-Aufforderung aktualisiert werden. Veraltete Informationen werden dabei automatisch entfernt.

32.3.3.1.2. Basiseinstellungen

Dieser Abschnitt beschreibt, wie Sie ein einfaches drahtloses Netzerk ohne Verschlüsselung unter FreeBSD einrichten. Nachdem Sie sich mit den Informationen dieses Abschnitts vertraut gemacht haben, sollten Sie Ihr drahtloses Netzwerk mit WPA verschlüsseln.

Das Einrichten eines drahtlosen Netzwerks erfolgt in drei Schritten: Der Auswahl eines Access Points, der Anmeldung Ihrer Station sowie der Konfiguration Ihrer IP-Adresse.

32.3.3.1.2.1. Einen Access Point auswählen

Im Normalfall wird sich Ihre Station automatisch mit einem der zur Verfügung stehenden Access Points verbinden. Sie müssen dazu lediglich Ihr drahtloses Gerät aktivieren. Alternativ können Sie auch einen Eintrag ähnlich dem folgenden in `/etc/rc.conf` aufnehmen:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

> **Anmerkung:** Wie bereits erwähnt, benötigen Sie unter FreeBSD 7.X anstelle dieser beiden Zeilen nur eine Zeile (mit dem entsprechenden Gerätetreiber):
>
> ```
> ifconfig_ath0="DHCP"
> ```

Wollen Sie sich hingegen mit einem bestimmten Access Point verbinden, müssen Sie dessen SSID angeben:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid Ihre_SSID DHCP"
```

Gibt es in Ihrem Netzwerk mehrere Access Points mit der gleichen SSID (was der Einfachheit wegen häufig der Fall ist), können Sie sich dennoch mit einem bestimmten Access Point verbinden. Dazu müssen Sie lediglich die BSSID des Access Points angeben (die Angabe der SSID ist in diesem Fall nicht erforderlich):

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid Ihre_SSID bssid xx:xx:xx:xx:xx:xx DHCP"
```

Es gibt noch weitere Möglichkeiten, den Zugriff auf bestimmte Access Point zu beschränken, beispielsweise durch die Begrenzung der Frequenzen, auf denen eine Station nach einem Access Point sucht. Sinnvoll ist ein solches Vorgehen beispielsweise, wenn Ihr drahtloses Gerät in verschiedenen Frequenzbereichen arbeiten kann, da in diesem Fall das Prüfen aller Frequenzen sehr zeitintensiv ist. Um nur innerhalb eines bestimmten Frequenzbereichs nach einem Access Point zu suchen, verwenden Sie die Option `mode`:

```
wlans_ath0="wlan0"
ifconfig_wlan0="mode 11g ssid Ihre_SSID DHCP"
```

Dadurch sucht Ihr drahtloses Gerät nur im 2,4 GHz-Band (802.11g), aber nicht innerhalb des 5 GHz-Bandes nach einem Access Point. Mit der Option `channel` können Sie eine bestimmte Frequenz vorgeben, auf der gesucht

werden soll. Die Option `chanlist` erlaubt die Angabe mehrerer erlaubter Frequenzen. Eine umfassende Beschreibung dieser Optionen finden Sie in der Manualpage ifconfig(8).

32.3.3.1.2.2. Authentifizierung

Wenn Sie einen Access Point gefunden haben, muss sich Ihrem Station am Access Point anmelden, bevor Sie Daten übertragen kann. Dazu gibt es verschiedene Möglichkeiten. Am häufigsten wird nach wie vor die sogenannte *offene Authentifizierung* verwendet. Dabei wird es jeder Station erlaubt, sich mit einem Netzwerk zu verbinden und Daten zu übertragen. Aus Sicherheitsgründen sollte diese Methode allerdings nur zu Testzwecken bei der erstmaligen Einrichtung eines drahtlosen Netzwerks verwendet werden. Andere Authentifizierungsmechanismen erfordern den Austausch kryptografischer Informationen, bevor Sie die Übertragung von Daten erlauben. Dazu gehören der Austausch fixer (vorher vereinbarter) Schlüssel oder Kennwörter sowie der Einsatz komplexerer Verfahren mit Backend-Diensten wie RADIUS. Die meisten Netzwerke nutzen allerdings nach wie vor die offene Authentifizierung, da dies die Voreinstellung ist. Am zweithäufigsten kommt das weiter unten beschriebene WPA-PSK (das auch als *WPA Personal* bezeichnet wird) zum Einsatz.

> **Anmerkung:** Verwenden Sie eine Apple AirPort® Extreme-Basisstation als Access Point, benötigen Sie wahrscheinlich sowohl die Shared-Key-Authentifizierung als auch einen WEP-Schlüssel. Die entsprechende Konfiguration erfolgt entweder in der Datei `/etc/rc.conf` oder über das Programm wpa_supplicant(8). Verwenden Sie nur eine einzige AirPort-Basisstation, benötigen Sie einen Eintrag ähnlich dem folgenden:
>
> ```
> wlans_ath0="wlan0"
> ifconfig_wlan0="authmode shared wepmode on weptxkey 1 wepkey 01234567 DHCP"
> ```
>
> Normalerweise sollten Sie Shared-Key-Authentifizierung aber nicht verwenden, da diese die Sicherheit des WEP-Schlüssel noch weiter verringert. Müssen Sie WEP einsetzen (beispielsweise weil Sie zu veralteten Geräten kompatibel bleiben müssen), sollten Sie WEP nur zusammen mit der offenen Authentifizierung (`open` authentication) verwenden. WEP wird im Abschnitt 32.3.3.1.4 näher beschrieben.

32.3.3.1.2.3. Eine IP-Adresse über DHCP beziehen

Nachdem Sie einen Access Point gefunden und sich authentifiziert haben, benötigen Sie noch eine IP-Adresse, die Sie in der Regel über DHCP zugewiesen bekommen. Dazu müssen Sie lediglich die Option `DHCP` in Ihre in der Datei `/etc/rc.conf` vorhandene Konfiguration Ihres drahtlosen Geräts aufnehmen:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Nun können Sie Ihr drahtloses Gerät starten:

```
# /etc/rc.d/netif start
```

Nachdem Sie das Gerät aktiviert haben, können Sie mit `ifconfig` den Status des Geräts `ath0` abfragen:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
        media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
```

```
status: associated
ssid dlinkap channel 11 (2462 Mhz 11g) bssid 00:13:46:49:41:76
country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
roam:rate 5 protmode CTS wme burst
```

`status: associated` besagt, dass sich Ihr Gerät mit dem drahtlosen Netzwerk verbunden hat (konkret mit dem Netzwerk `dlinkap`). `bssid 00:13:46:49:41:76` gibt die MAC-Adresse Ihres Access Points aus und die Zeile mit `authmode OPEN` informiert Sie darüber, dass Ihre Kommunikation nicht verschlüsselt wird.

32.3.3.1.2.4. Statische IP-Adressen

Alternativ zu dynamischen IP-Adressen können Sie auch eine statische IP-Adresse verwenden. Dazu ersetzen Sie in Ihrer Konfiguration `DHCP` durch die zu verwendende IP-Adresse. Beachten Sie dabei, dass Sie die anderen Konfigurationsparameter nicht versehentlich verändern:

```
wlans_ath0="wlan0"
ifconfig_wlan0="inet 192.168.1.100 netmask 255.255.255.0 ssid your_ssid_here"
```

32.3.3.1.3. WPA

Bei WPA (Wi-Fi Protected Access) handelt es sich um ein Sicherheitsprotokoll, das in 802.11-Netzwerken verwendet wird, um die Nachteile von WEP (fehlende Authentifizierung und schwache Verschlüsselung) zu vermeiden. WPA stellt das aktuelle 802.1X-Authentifizierungsprotokoll dar und verwendet eine von mehreren Chiffren, um die Datensicherheit zu gewährleisten. Die einzige Chiffre, die von WPA verlangt wird, ist TKIP (*Temporary Key Integrity Protocol*), eine Chiffre, die die von WEP verwendete RC4-Chiffre um Funktionen zur Prüfung der Datenintegrität und zur Erkennung und Bekämpfung von Einbruchsversuchen erweitert. TKIP ist durch Softwaremodifikationen auch unter veralteter Hardware lauffähig. Im Vergleich zu WEP ist WPA zwar sehr viel sicherer, es ist aber dennoch nicht völlig immun gegen Angriffe. WPA definiert mit AES-CCMP noch eine weitere Chiffre als Alternative zu TKIP. AES-CCMP (das häufig als WPA2 oder RSN bezeichnet wird) sollte, wenn möglich, eingesetzt werden.

WPA definiert Authentifizierungs- und Verschlüsselungsprotokolle. Die Authentifizierung erfolgt in der Regel über eine der folgenden Techniken: 802.1X gemeinsam mit einem Backend-Authentifizierungsdienst wie RADIUS, oder durch einen Minimal-Handshake zwischen der Station und dem Access Point mit einem vorher vereinbarten gemeinsamen Schlüssel. Die erste Technik wird als *WPA Enterprise*, die zweite hingegen als *WPA Personal* bezeichnet. Da sich der Aufwand für das Aufsetzen eines RADIUS-Backend-Servers für die meisten drahtlosen Netzwerke nicht lohnt, wird WPA in der Regel als WPA-PSK (WPA, Pre-Shared-Key) konfiguriert.

Die Kontrolle der drahtlosen Verbindung sowie die vorangehende Authentifizierung (über Schlüssel oder durch die Kommunikation mit einem Server) erfolgt über das Programm wpa_supplicant(8), das über die Datei `/etc/wpa_supplicant.conf` eingerichtet wird. Ausführliche Informationen zur Konfiguration des Programms finden sich in der Manualpage wpa_supplicant.conf(5).

32.3.3.1.3.1. WPA-PSK

WPA-PSK oder WPA-Personal basiert auf einem gemeinsamen (vorher vereinbarten) Schlüssel (PSK), der aus einem Passwort generiert und danach als Master-Key des drahtlosen Netzwerks verwendet wird. Jeder Benutzer des

drahtlosen Netzwerks verwendet daher *den gleichen* Schlüssel. WPA-PSK sollte nur in kleinen Netzwerken eingesetzt werden, in denen die Konfiguration eines Authentifizierungsservers nicht möglich oder erwünscht ist.

> **Warnung:** Achten Sie darauf, dass Sie immer starke Passwörter verwenden, die ausreichend lang sind und, wenn möglich, auch Sonderzeichen enthalten, damit diese nicht leicht erraten und/oder umgangen werden können.

Der erste Schritt zum Einsatz von WPA-PSK ist die Konfiguration der SSID und des gemeinsamen Schlüssels Ihres Netzwerks in der Datei `/etc/wpa_supplicant.conf`:

```
network={
  ssid="freebsdap"
  psk="freebsdmall"
}
```

Danach geben Sie in `/etc/rc.conf` an, dass WPA zur Verschlüsselung eingesetzt werden soll und dass die IP-Adresse über DHCP bezogen wird:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun können Sie Ihr Netzgerät aktivieren:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Alternativ können Sie die Konfiguration von WPA-PSK auch manuell durchführen, wobei Sie wiederum die Konfigurationsdatei `/etc/wpa_supplicant.conf` verwenden:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)
Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to 00:11:95:c3:0d:ac completed (auth) [id=0 id_str=]
```

Im zweiten Schritt starten Sie nun `dhclient`, um eine IP-Adresse vom DHCP-Server zu beziehen:

```
# dhclient wlan0
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

> **Anmerkung:** Enthält Ihre `/etc/rc.conf` bereits die Zeile `ifconfig_wlan0="DHCP"`, müssen Sie `dhclient` nicht mehr manuell aufrufen, da `dhclient` in diesem Fall automatisch gestartet wird, nachdem `wpa_supplicant` die Schlüssel übergibt.

Sollte der Einsatz von DHCP nicht möglich sein, können Sie auch eine statische IP-Adresse angeben, nachdem `wpa_supplicant` Ihre Station authentifiziert hat:

```
# ifconfig wlan0 inet 192.168.0.100 netmask 255.255.255.0
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Verwenden Sie DHCP nicht, müssen Sie zusätzlich noch das Standard-Gateway sowie den/die Nameserver manuell festlegen:

```
# route add default your_default_router
# echo "nameserver your_DNS_server" >> /etc/resolv.conf
```

32.3.3.1.3.2. WPA und EAP-TLS

Die zweite Möglichkeit, WPA einzusetzen, ist die Verwendung eines 802.1X-Backend-Authentifizierungsservers. Diese Variante wird als WPA-Enterprise bezeichnet, um sie vom weniger sicheren WPA-Personal abzugrenzen, das

auf dem Austausch gemeinsamer (und vorher vereinbarter Schlüssel) basiert. Die bei WPA-Enterprise verwendete Authentifizierung basiert auf EAP (*Extensible Authentication Protocol*).

EAP selbst bietet keine Verschlüsselung, sondern operiert in einem verschlüsselten Tunnel. Es gibt verschiedene, auf EAP basierende Authentifizierungsmethoden, darunter EAP-TLS, EAP-TTLS sowie EAP-PEAP.

Bei EAP-TLS (*EAP with Transport Layers Security*) handelt es sich um sehr gut unterstütztes Authentifizierungsprotokoll, da es sich dabei um die erste EAP-Methode handelt, die von der Wi-Fi Alliance (http://www.wi-fi.org/) zertifiziert wurde. EAP-TLS erfordert drei Zertifikate: Das (auf allen Rechnern installierte) CA-Zertifikat, das Server-Zertifikat Ihres Authentifizierungsservers, sowie ein Client-Zertifikat für jeden drahtlosen Client. Sowohl der Authentifizierungsservers als auch die drahtlosen Clients authentifizieren sich gegenseitig durch ihre Zertifikate, wobei sie überprüfen, ob diese Zertifikate auch von der Zertifizierungs-Authorität (CA) des jeweiligen Unternehmens signiert wurden.

Die Konfiguration erfolgt (analog zu WPA-PSK) über die Datei `/etc/wpa_supplicant.conf`:

```
network={
  ssid="freebsdap" ❶
  proto=RSN  ❷
  key_mgmt=WPA-EAP ❸
  eap=TLS ❹
  identity="loader" ❺
  ca_cert="/etc/certs/cacert.pem" ❻
  client_cert="/etc/certs/clientcert.pem" ❼
  private_key="/etc/certs/clientkey.pem" ❽
  private_key_passwd="freebsdmallclient" ❾
}
```

❶ Der Name des Netzwerks (die SSID).

❷ Das RSN/WPA2-Protokoll (IEEE 802.11i) wird verwendet.

❸ Die `key_mgmt`-Zeile bezieht sich auf das verwendete Key-Management-Protokoll. In diesem Beispiel wird WPA gemeinsam mit der EAP-Authentifizierung verwendet (`WPA-EAP`).

❹ Die für die Verbindung verwendete EAP-Methode.

❺ Das `identity`-Feld enthält den von EAP verwendeten Identifizierungsstring.

❻ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an. Dieses Datei wird benötigt, um das Server-Zertifikat zu verifizieren.

❼ Die `client_cert`-Zeile gibt den Pfad zum Client-Zertifikat an. Jeder Client hat ein eigenes, innerhalb des Netzwerks eindeutiges, Zertifikat.

❽ Das Feld `private_key` gibt den Pfad zum privaten Schlüssel des Client-Zertifikat an.

❾ Das Feld `private_key_passwd` enthält die Passphrase für den privaten Schlüssel.

Danach fügen Sie die folgende Zeile in `/etc/rc.conf` ein:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun können Sie Ihr drahtloses Gerät über das `rc.d`-System aktivieren:

```
# /etc/rc.d/netif start
```

```
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Alternativ können Sie Ihr drahtloses Gerält wiederum manuell über `wpa_supplicant` und `ifconfig` aktivieren.

32.3.3.1.3.3. WPA und EAP-TTLS

Bei EAP-TLS müssen sowohl der Authentifizierungsserver als auch die Clients jeweils ein eigenes Zertifikat aufweisen. Setzen Sie hingegen EAP-TTLS (*EAP-Tunneled Transport Layer Security*) ein, ist das Client-Zertifikat optional. EAP-TTLS geht dabei ähnlich vor wie verschlüsselte Webseiten, bei denen der Webserver einen sicheren SSL-Tunnel erzeugen kann, ohne dass der Besucher dabei über ein client-seitiges Zertifikat verfügen muss. EAP-TTLS verwendet einen verschlüsselten TLS-Tunnel zum sicheren Transport der Authentifizierungsdaten.

Die Konfiguration von EAP-TTLS erfolgt in der Datei `/etc/wpa_supplicant.conf`:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS ❶
  identity="test" ❷
  password="test" ❸
  ca_cert="/etc/certs/cacert.pem" ❹
  phase2="auth=MD5" ❺
}
```

❶ Die für die Verbindung verwendete EAP-Methode.

❷ Das `identity`-Feld enthält den Identifizierungsstring für die EAP-Authentifizierung innerhalb des verschlüsselten TlS-Tunnels.

❸ Das `password`-Feld enthält die Passphrase für die EAP-Authentifizierung.

❹ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an, das benötigt wird, um das Server-Zertifikat zu verifizieren.

❺ Die innerhalb des verschlüsselten TLS-Tunnels verwendete Authentifizierungsmethode. In unserem Beispiel handelt es sich dabei um EAP und MD5. Diese Phase der “inneren Authentifizierung” wird oft als “phase2” bezeichnet.

Folgende Zeilen müssen zusätzlich in die Datei `/etc/rc.conf` aufgenommen werden:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun können Sie Ihr drahtloses Gerät aktivieren:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

32.3.3.1.3.4. WPA und EAP-PEAP

PEAP (*Protected EAP*) wurde als Alternative zu EAP-TTLS entwickelt. Es gibt zwei verschiedene PEAP-Methoden, wobei es sich bei PEAPv0/EAP-MSCHAPv2 um die häufiger verwendete Methode handelt. In den folgenden Ausführungen wird der Begriff PEAP synonym für diese EAP-Methode verwendet. PEAP ist nach EAP-TLS der am häufigsten verwendete und am besten unterstützte EAP-Standard.

PEAP arbeitet ähnlich wie EAP-TTLS: Es verwendet ein server-seitiges Zertifikat, um einen verschlüsselten TLS-Tunnel zu erzeugen, über den die sichere Authentifizierung zwischen den Clients und dem Authentifizierungsserver erfolgt. In Sachen Sicherheit unterscheiden sich EAP-TTLS und PEAP allerdings: PEAP überträgt den Benutzernamen im Klartext und verschlüsselt nur das Passwort, während EAP-TTLS sowohl den Benutzernamen als auch das Passwort über den TLS-Tunnel überträgt.

Um EAP-PEAP einzurichten, müssen Sie die Konfigurationsdatei `/etc/wpa_supplicant.conf` anpassen:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP ❶
  identity="test" ❷
  password="test" ❸
  ca_cert="/etc/certs/cacert.pem" ❹
  phase1="peaplabel=0" ❺
  phase2="auth=MSCHAPV2" ❻
}
```

❶ Die für die Verbindung verwendete EAP-Methode.

❷ Das `identity`-Feld enthält den Identifizierungsstring für die innerhalb des verschlüsselten TLS-Tunnels erfolgende EAP-Authentifizierung.

❸ Das Feld `password` enthält die Passphrase für die EAP-Authentifizierung.

❹ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an, das zur Verifizierung des Server-Zertifikats benötigt wird.

❺ Dieses Feld enthält die Parameter für die erste Phase der Authentifizierung (also den TLS-Tunnel). Je nach dem, welchen Authentifizierungsserver Sie verwenden, müssen Sie hier einen unterschiedlichen Wert angeben. In den meisten Fällen wird dieses Feld den Wert "client EAP encryption" aufweisen, der durch die Angabe von `peaplabel=0` gesetzt wird. Weitere Informationen zur Konfiguration von PEAP finden Sie in der Manualpage wpa_supplicant.conf(5).

❻ Das innerhalb des verschlüsselten TLS-Tunnels verwendete Authentifizierungsprotokoll. In unserem Beispiel handelt es sich dabei um `auth=MSCHAPV2`.

Danach fügen Sie die folgende Zeile in `/etc/rc.conf` ein:

```
ifconfig_ath0="WPA DHCP"
```

Zuletzt müssen Sie die Netzwerkkarte noch aktivieren:

```
# /etc/rc.d/netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

32.3.3.1.4. WEP

WEP (Wired Equivalent Privacy) ist Teil des ursprünglichen 802.11-Standards. Es enthält keinen Authentifzierungsmechanismus und verfügt lediglich über eine schwache Zugriffskontrolle, die sehr leicht umgangen werden kann.

WEP kann über `ifconfig` aktiviert werden:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 inet 192.168.1.100 netmask 255.255.255.0 \
     ssid my_net wepmode on weptxkey 3 wepkey 3:0x3456789012
```

- Mit `weptxkey` geben Sie an, welcher WEP-Schlüssel für für die Datenübertragung verwendet wird (in unserem Beispiel ist dies der dritte Schlüssel). Der gleiche Schlüssel muss auch am Access Point eingestellt sein. Kennen Sie den vom Access Point verwendeten Schlüssel nciht, sollten Sie zuerst den Wert `1` (d.h. den ersten Schlüssel) für diese Variable verwenden.
- Mit `wepkey` legen Sie den zu verwendenden WEP-Schlüssel in der Form *`Nummer:Schlüssel`* fest. Ist der Schlüssel "Nummer" nicht vorhanden, wird automatisch Schlüssel `1` verwendet. Die Angabe von "Nummer" ist zwingend nötig, wenn Sie einen anderen als den ersten Schlüssel verwenden wollen.

> **Anmerkung:** In Ihrer Konfiguration müssen Sie `0x3456789012` durch den an Ihrem Access Point konfigurierten Schlüssel ersetzen.

Weitere Informationen finden Sie in der Manualpage ifconfig(8).

Das Programm `wpa_supplicant` eignet sich ebenfalls dazu, WEP für Ihr drahtloses Gerät zu aktivieren. Obige Konfiguration lässt sich dabei durch die Aufnahme der folgenden Zeilen in die Datei `/etc/wpa_supplicant.conf` realisieren:

```
network={
  ssid="my_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Danach müssen Sie das Programm noch aufrufen:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:49:41:76 (SSID='dlinkap' freq=2437 MHz)
Associated with 00:13:46:49:41:76
```

32.3.4. Ad-hoc-Modus

Der IBSS-Modus (auch als Ad-hoc-Modus bezeichnet), ist für Punkt-zu-Punkt-Verbindungen vorgesehen. Um beispielsweise eine Ad-hoc-Verbindung zwischen den Rechnern `A` und `B` aufzubauen, benötigen Sie lediglich zwei IP-Adressen und eine SSID.

Auf dem Rechner `A` geben Sie Folgendes ein:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
       ether 00:11:95:c3:0d:ac
       inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
       media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
       status: running
```

```
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Der adhoc-Parameter gibt an, dass die Schnittstelle im IBSS-Modus läuft.

Rechner B sollte nun in der Lage sein, Rechner A zu finden:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 up scan
  SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
  freebsdap       02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

Der Wert I (Spalte CAPS) gibt an, dass sich Rechner A im Ad-hoc-Modus befindet. Nun müssen Sie nur noch Rechner B eine unterschiedliche IP-Adresse zuweisen:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Damit sind die Rechner A und B bereit und können untereinander Daten austauschen.

32.3.5. FreeBSD Host Access Points

FreeBSD kann als Access Point (AP) agieren. Dies verhindert, dass man sich einen Hardware AP kaufen oder ein ad-hoc Netzwerk laufen lassen muss. Dies kann sinnvoll sein, falls Ihre FreeBSD-Computer als Gateway zu einem anderen Netzwerk (z.B. Internet) fungiert.

32.3.5.1. Grundeinstellungen

Bevor Sie ihren FreeBSD-Computer als einen AP konfigurieren, muss der Kernel mit dem für ihre Wireless-Karte entsprechenden Treibern konfiguriert werden. Sie müssen ebenfalls die Sicherheitsprotokolle, die Sie nutzen wollen, dem Kernel hinzufügen. Für weitere Informationen siehe: Abschnitt 32.3.2.

> **Anmerkung:** Die Verwendung der NDIS und Windows Treiber erlauben zur Zeit keinen AP-Modus. Nur die nativen FreeBSD-Wireless-Treiber unterstüten den AP Modus.

Nachdem die Unterstützung für ihr drahtloses Netzwerk geladen ist, können Sie überprüfen, ob Ihr Wireless-Gerät den hostbasierenden Access-Point Modus (auch bekannt als hostap Modus) unterstützt:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 list caps
drivercaps=6f85edc1<STA,FF,TURBOP,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,MBSS,WPA1,W
```

```
cryptocaps=1f<WEP,TKIP,AES,AES_CCM,TKIPMIC>
```

Diese Ausgabe zeigt die Möglichkeiten der Karte. Das Wort `HOSTAP` bestätigt, dass diese Wireless-Karte als Access Point agieren kann. Die verschiedenen unterstützten Algorithmen (z.B. WEP, TKIP, AES usw.) werden ebenfalls angezeigt. Diese Informationen sind wichtig, wenn Sie wissen wollen, welche Sicherheitsprotokolle auf diesem Access Point verwendbar sind.

Das Wireless-Gerät kann nur während der Erzeugung des Pseudo-Geräts in den hostap-Modus gesetzt werden. Zuvor erstellte Pseudo-Geräte müssen also vorher zerstört werden:

```
# ifconfig wlan0 destroy
```

Danach muss das Gerät erneut erstellt werden, bevor die restlichen Netzwerkparameter konfiguriert werden können:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel 1
```

Benutzen Sie danach erneut den Befehl `ifconfig`, um den Status der `wlan0`-Schnittstelle abzufragen:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst dtimperiod 1 -dfs
```

Die `hostap`-Parameter geben die Schnittstelle an, die im hostbasierenden Access Point Modus läuft.

Die Konfiguration der Schnittstelle kann durch Hinzufügen der folgenden Zeilen in die Datei `/etc/rc.conf` automatisch während des Bootvorganges erfolgen:

```
wlans_ath0="wlan0"
create_args_wlan0="wlanmode hostap"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel 1"
```

32.3.5.2. Hostbasierender Access Point ohne Authentifizierung oder Verschlüsselung

Obwohl es nicht empfohlen wird, einen AP ohne jegliche Authentifizierung oder Verschlüsselung laufen zu lassen, ist es eine einfache Art zu testen, ob der AP funktioniert. Diese Konfiguration ist auch wichtig für die Fehlersuche bei Client-Problemen.

Nachdem Sie den AP, wie oben beschrieben, konfiguriert haben, ist es möglich von einem anderen drahtlosen Computer eine Suche nach dem AP zu starten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M -66:-96  100 ES   WME
```

Der Client-Rechner fand den Access Point und kann mit ihm verbunden werden:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
      scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
      roam:rate 5 protmode CTS wme burst
```

32.3.5.3. WPA-basierender Host-Access Point

Dieser Abschnitt beschäftigt sich mit dem Konfigurieren eines FreeBSD-Access-Points mit dem WPA-Sicherheitsprotokoll. Weitere Einzelheiten zu WPA und der Konfiguration von Clients mit WPA finden Sie im Abschnitt 32.3.3.1.3.

Der **hostapd**-Dienst wird genutzt, um die Client-Authentifizierung und das Schlüsselmanagement auf dem Access Point mit aktiviertem WPA zu nutzen.

In den folgenden Abschnitten werden allen Konfigurationen auf dem FreeBSD-Computer ausgeführt, der als AP agiert. Nachdem der AP korrekt arbeitet, sollte **hostapd** automatisch beim Booten durch folgende Zeile in der `/etc/rc.conf` aktiviert werden:

```
hostapd_enable="YES"
```

Bevor Sie versuchen **hostapd** zu konfigurieren, stellen Sie sicher, dass die Grundeinstellungen, wie in Abschnitt 32.3.5.1 beschrieben, ausgeführt wurden.

32.3.5.3.1. WPA-PSK

WPA-PSK ist für kleine Netzwerke gedacht, in denen die Verwendung eines Authentifizierungs-Backend-Server nicht möglich oder erwünscht ist.

Die Konfiguration wird in `/etc/hostapd.conf` durchgeführt:

```
interface=wlan0 ❶
debug=1 ❷
ctrl_interface=/var/run/hostapd ❸
ctrl_interface_group=wheel ❹
ssid=freebsdap ❺
wpa=1 ❻
wpa_passphrase=freebsdmall ❼
wpa_key_mgmt=WPA-PSK ❽
wpa_pairwise=CCMP TKIP ❾
```

❶ Dieses Feld zeigt die Wireless-Schnittstelle an, die für den Access Point verwendet wird an.

❷ Dieses Feld legt den debuglevel von **hostapd** während der Ausführung fest. Ein Wert von `1` ist der kleinste zulässige Wert.

❸ Das `ctrl_interface`-Feld gibt den Pfadnamen des Verzeichnisses an, der von **hostapd** dazu genutzt wird, um die domain socket Dateien zu speichern, die für die Kommunikation mit externen Programmen, wie z.B. hostapd_cli(8), benutzt werden. Hier wurden die Standardwerte benutzt.

❹ Die Zeile `ctrl_interface_group` legt fest, welche Gruppe (hier ist es die `wheel`-Gruppe) die Erlaubnis hat, die Schnittstellendateien zu kontrollieren.

❺ Dieses Feld setzt den Netzwerknamen.

❻ Das `wpa`-Feld aktiviert WPA und gibt an welches WPA-Authentifizierungprotokoll benötigt wird. Ein Wert von `1` konfiguriert den AP mit WPA-PSK.

❼ Das `wpa_passphrase`-Feld beinhaltet das ASCII-Passwort für die WPA-Authentifikation.

> **Warnung:** Verwenden Sie immer sichere Passwörter, die ausreichend lang sind und aus vielen unterschiedlichen Zeichen bestehen, damit sie nicht erraten werden oder umgangen werden können.

❽ Die `wpa_key_mgmt` Zeile bestimmt das Schlüsselmanagement-Protokoll, das benutzt wird. In unserem Fall ist es WPA-PSK.

❾ Das `wpa_pairwise` Feld zeigt die zulässigen Verschlüsselungs-Algorithmen des Access Points. Hier werden beide, TKIP (WPA) und CCMP (WPA2), akzeptiert. CCMP-Verschlüsselung ist eine Alternative zu TKIP und sollte wenn möglich eingesetzt werden. TKIP sollte nur da eingesetzt werden, wo kein CCMP möglich ist.

Als nächstes wird der **hostapd** gestartet:

```
# /etc/rc.d/hostapd forcestart

# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 2290
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      inet6 fe80::211:95ff:fec3:dac%ath0 prefixlen 64 scopeid 0x4
      ether 00:11:95:c3:0d:ac
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: associated
      ssid freebsdap channel 1 bssid 00:11:95:c3:0d:ac
      authmode WPA2/802.11i privacy MIXED deftxkey 2 TKIP 2:128-bit txpowmax 36 protmode CTS dtim
```

Der Access Point läft nun, die Clients können mit ihm verbunden werden. Weitere Informationen finden Sie im Abschnitt 32.3.3.1.3. Es ist möglich zu sehen, welche Stationen mit dem AP verbunden sind. Dazu geben Sie den Befehl `ifconfig wlan0 list sta` ein.

32.3.5.4. WEP hostbasierender Access Point

Es ist nicht empfehlenswert, einen Access Point mit WEP zu konfigurieren, da es keine Authentifikationsmechanismen gibt und WEP leicht zu knacken ist. Einige ältere WLAN-Karten unterstützen nur WEP als Sicherheitsprotokoll. Für solche Karten ist es notwendig den AP ohne Authentifikation, Verschlüsselung oder mit dem WEP-Protokoll zu konfigurieren.

Das Wireless-Gerät kann nun in den hostap-Modus versetzt werden und mit der korrekten SSID und IP-Adresse konfiguriert werden:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 \
      ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g
```

- Der `weptxkey` gibt an, welcher WEP-Schlüssel bei der Übertragung benutzt wird. Hier nutzen wir den 3. Schlüssel (die Nummerierung der Schlüssel beginnt bei `1`). Dieses Parameter muss angegeben sein, damit die Daten wirklich verschlüsselt werden.
- Der `wepkey` gibt den gewählten WEP-Schlüssel an. Er sollte im folgenden Format *`index:key`* vorliegen. Wenn kein Index vorhanden ist, wird der Schlüssel `1` benutzt. Das bedeutet wir brauchen einen Index, falls wir einen anderen Schlüssel als den ersten nutzen wollen.

Benutzen Sie den Befehl `ifconfig` noch einmal um den Status der `wlan0`-Schnittstelle zu sehen:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 4 (2427 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit
      txpower 21.5 scanvalid 60 protmode CTS wme burst dtimperiod 1 -dfs
```

Es ist möglich, von einem anderen drahtlosen Computer eine Suche nach dem AP zu starten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 EPS
```

Der Client-Rechner fand den Access Point und kann mit den korrekten Parametern (Schlüssel usw.) mit ihm verbunden werden. Weitere Informationen gibt es in folgendem Abschnitt 32.3.3.1.4

32.3.6. Benutzung von drahtgebundenen und drahtlosen Verbindungen

Eine Verbindung per Kabel bietet eine bessere Leistung und eine höhere Zuverlässigkeit, während die Wireless-Verbindung eine höhere Flexibilität und Mobilität bietet. Benutzer von Laptops wollen normalerweise beides nutzen und zwischen beiden hin und her schalten.

Unter FreeBSD ist es möglich zwei oder mehr Netzwerkschnittstellen in einem “failover”-Mode zu kombinieren, so dass automatisch die beste verfügbare Verbindung aus der Gruppe ausgewählt wird, sobald der Linkstatus wechselt.

Wir behandeln Link-Aggregation und Failover in dem Kapitel Abschnitt 32.6. Dort gibt es auch ein Beispiel (Beispiel 32-3) für die Verwendung von sowohl kabelgebundenen wie auch drahtlosen Verbindungen.

32.3.7. Problembehandlung

Die folgenden Auflistung zeigt, wie Sie einige häufig auftretende Probleme bei der Einrichtung Ihres drahtlosen Netzwerks beheben können.

- Wird Ihr Access Point bei der Suche nicht gefunden, sollten Sie überprüfen, ob Sie bei Konfiguration Ihres drahtlosen Geräts die Anzahl der Kanäle beschränkt haben.
- Wenn Sie sich nicht mit Ihrem Access Point verbinden können, sollten Sie überprüfen, ob die Konfiguration Ihrer Station auch der des Access Points entspricht. Achten Sie dabei speziell auf die Authentifzierungsmethode und die Sicherheitsprotokolle. Halten Sie Ihre Konfiguration so einfach wie möglich. Verwenden Sie ein Sicherheitsprotokoll wie WPA oder WEP, sollten Sie testweise Ihren Access Point auf *offene Authentifizierung* und *keine Sicherheit* einstellen. Danach versuchen Sie sich erneut mit Ihren Access Point zu verbinden.
- Nachdem Sie sich mit dem Access Point verbinden können, prüfen Sie die Sicherheitseinstellungen, beginnend mit einfachen Werkzeugen wie ping(8).

  Das Programm `wpa_supplicant` kann Ihnen bei der Fehlersuche helfen. Dazu starten Sie es manuell mit der Option `-dd` und durchsuchen anschließend die Protokollinformationen nach eventuellen Fehlermeldungen.
- Zusätzlich gibt es auch zahlreiche Low-Level-Debugging-Werkzeuge. Die Ausgabe von Debugging-Informationen des 802.11 Protocol Support Layers lassen sich mit dem Programm `wlandebug` (das sich unter `/usr/src/tools/tools/net80211` befindet) aktivieren. Um beispielsweise während der Suche nach Access Points und des Aufbaus von 802.11-Verbindungen (*Handshake*) auftretende Systemmeldungen auf die Konsole auszugeben, verwenden Sie den folgenden Befehl:

  ```
  # wlandebug -i ath0 +scan+auth+debug+assoc
    net.wlan.0.debug: 0 => 0xc80000<assoc,auth,scan>
  ```

  Der 802.11-Layer liefert umfangreiche Statistiken, die Sie mit dem Werkzeug `wlanstats` abrufen können. Diese Statistiken sollten alle Fehler identifizieren, die im 802.11-Layer auftreten. Beachten Sie aber, dass einige Fehler bereits im darunterliegenden Gerätetreiber auftreten und daher in diesen Statistiken nicht enthalten sind. Wie Sie Probleme des Gerätetreibers identifizieren, entnehmen Sie bitte der Dokumentation Ihres Gerätetreibers.

Können Sie Ihr Problem durch diese Maßnahmen nicht lösen, sollten Sie einen Problembericht (PR) erstellen und die Ausgabe der weiter oben genannten Werkzeuge in den Bericht aufnehmen.

32.4. Bluetooth

Beigetragen von Pav Lucistnik.

32.4.1. Übersicht

Bluetooth ermöglicht die Bildung von persönlichen Netzwerken über drahtlose Verbindungen bei einer maximalen Reichweite von 10 Metern und operiert im unlizensierten 2,4-GHz-Band. Solche Netzwerke werden normalerweise spontan gebildet, wenn sich mobile Geräte, wie Mobiltelefone, Handhelds oder Notebooks miteinander verbinden. Im Gegensatz zu Wireless LAN ermöglicht Bluetooth auch höherwertige Dienste, wie FTP-ähnliche Dateiserver, Filepushing, Sprachübertragung, Emulation von seriellen Verbindungen und andere mehr.

Der Bluetooth-Stack von FreeBSD verwendet das Netgraph-Framework (netgraph(4)). Viele Bluetooth-USB-Adapter werden durch den ng_ubt(4)-Treiber unterstützt. Auf dem Chip BCM2033 von Broadcom

basierende Bluetooth-Geräte werden von den Treibern ubtbcmfw(4) sowie ng_ubt(4) unterstützt. Die Bluetooth-PC-Card 3CRWB60-A von 3Com verwendet den ng_bt3c(4)-Treiber. Serielle sowie auf UART basierende Bluetooth-Geräte werden von sio(4), ng_h4(4) sowie hcseriald(8) unterstützt. Dieses Kapitel beschreibt die Verwendung von USB-Bluetooth-Adaptern.

32.4.2. Die Bluetooth-Unterstützung aktivieren

Bluetooth-Unterstützung ist in der Regel als Kernelmodul verfügbar. Damit ein Gerät funktioniert, muss der entsprechende Treiber im Kernel geladen werden:

```
# kldload ng_ubt
```

Ist das Bluetooth-Gerät beim Systemstart angeschlossen, kann das entsprechende Modul auch von `/boot/loader.conf` geladen werden:

```
ng_ubt_load="YES"
```

Schließen Sie Ihren USB-Adapter an, sollte eine Meldung ähnlich der folgenden auf der Konsole (oder in syslog) erscheinen:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Zum Starten und Beenden des Bluetooth-Stacks verwenden Sie das Skript `/etc/rc.d/bluetooth`. Es ist empfehlenswert, den Bluetooth-Stack zu beenden, bevor Sie den Adapter entfernen. Selbst wenn Sie dies nicht tun, kommt es (normalerweise) zu keinem fatalen Fehler. Wenn Sie den Bluetooth-Stack starten, erhalten Sie eine Meldung ähnlich der folgenden:

```
# /etc/rc.d/bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

32.4.3. Das Host Controller Interface (HCI)

Das *Host Controller Interface* (HCI) bietet eine Befehlsschnittstelle zum Basisbandcontroller und Linkmanager, sowie Zugriff auf den Hardwarestatus und die Kontrollregister. Dadurch wird ein einheitlicher Zugriff auf die Fähigkeiten des Bluetooth-Basisbands möglich. Die HCI-Layer des Rechners tauschen Daten und Befehle mit der HCI-Firmware der Bluetooth-Geräte aus. Über den Host Controller Transport Layer-Treiber (also den physikalischen Bus) können beide HCI-Layer miteinander kommunizieren.

Eine einzelne Netgraph-Gerätedatei vom Typ *hci* wird für ein einzelnes Bluetooth-Gerät erzeugt. Die HCI-Gerätedatei ist normalerweise mit der Bluetooth-Gerätetreiberdatei (downstream) sowie der L2CAP-Gerätedatei (upstream) verbunden. Alle HCI-Operationen müssen über die HCI-Gerätedatei und nicht über die Treiberdatei erfolgen. Der Standardname für die HCI-Gerätedatei (die in ng_hci(4) beschrieben wird) lautet “devicehci”.

Eine der wichtigsten Aufgaben ist das Auffinden von sich in Reichweite befindenden Bluetooth-Geräten. Diese Funktion wird als *inquiry* bezeichnet. Inquiry sowie andere mit HCI in Verbindung stehende Funktionen werden von hccontrol(8) zur Verfügung gestellt. Das folgende Beispiel zeigt, wie man herausfindet, welche Bluetooth-Geräte sich in Reichweite befinden. Eine solche Abfrage dauert nur wenige Sekunden. Beachten Sie, dass ein Gerät nur dann antwortet, wenn es sich im Modus *discoverable* befindet.

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR stellt, ähnlich der MAC-Adresse einer Netzwerkkarte, die eindeutige Adresse eines Bluetooth-Gerätes dar. Diese Adresse ist für die Kommunikation mit dem Gerät nötig. Es ist aber auch möglich, BD_ADDR einen Klartextnamen zuzuweisen. Die Datei `/etc/bluetooth/hosts` enthält Informationen über die bekannten Bluetooth-Rechner. Das folgende Beispiel zeigt, wie man den Klartextnamen eines entfernten Geräts in Erfahrung bringen kann:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav’s T39
```

Wenn Sie ein entferntes Bluetooth-Gerät abfragen, wird dieses Ihren Rechner unter dem Namen “your.host.name (ubt0)” finden. Dieser Name kann aber jederzeit geändert werden.

Bluetooth ermöglicht Punkt-zu-Punkt-Verbindungen (an denen nur zwei Bluetooth-Geräte beteiligt sind), aber auch Punkt-zu-Multipunkt-Verbindungen, bei denen eine Verbindung von mehreren Bluetooth-Geräten gemeinsam genutzt wird. Das folgende Beispiel zeigt, wie man die aktiven Basisbandverbindungen des lokalen Gerätes anzeigen kann:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

Ein *connection handle* ist für die Beendigung einer Basisbandverbindung nützlich. Im Normalfall werden inaktive Verbindungen aber automatisch vom Bluetooth-Stack getrennt.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Rufen Sie `hccontrol help` auf, wenn Sie eine komplette Liste aller verfügbaren HCI-Befehle benötigen. Die meisten dieser Befehle müssen nicht als `root` ausgeführt werden.

32.4.4. Das Logical Link Control and Adaptation Protocol (L2CAP)

Das *Logical Link Control and Adaptation Protocol* (L2CAP) bietet höherwertigen Protokollen verbindungsorientierte und verbindungslose Datendienste an. Dazu gehören auch Protokollmultiplexing, Segmentierung und Reassemblierung. L2CAP erlaubt höherwertigen Protokollen und Programmen den Versand und Empfang von L2CAP-Datenpaketen mit einer Länge von bis zu 64 Kilobytes.

L2CAP arbeitet *kanal*basiert. Ein Kanal ist eine logische Verbindung innerhalb einer Basisbandverbindung. Jeder Kanal ist dabei an ein einziges Protokoll gebunden. Mehrere Geräte können an das gleiche Protokoll gebunden sein, es ist aber nicht möglich, einen Kanal an mehrere Protokolle zu binden. Jedes über einen Kanal ankommende L2CAP-Paket wird an das entsprechende höherwertige Protokoll weitergeleitet. Mehrere Kanäle können sich die gleiche Basisbandverbindung teilen.

Eine einzelne Netgraph-Gerätedatei vom Typ *l2cap* wird für ein einzelnes Bluetooth-Gerät erzeugt. Die L2CAP-Gerätedatei ist normalerweise mit der Bluetooth-HCI-Gerätedatei (downstream) sowie der Bluetooth-Socket-Gerätedatei (upstream) verbunden. Der Standardname für die L2CAP-Gerätedatei, die in ng_l2cap(4) beschrieben wird, lautet “devicel2cap”.

Ein nützlicher Befehl zum Anpingen von anderen Geräten ist l2ping(8). Einige Bluetooth-Geräte senden allerdings nicht alle erhaltenen Daten zurück. Die Ausgabe `0 bytes` ist also kein Fehler:

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

Das Programm l2control(8) liefert Informationen über L2CAP-Dateien. Das folgende Beispiel zeigt, wie man die Liste der logischen Verbindungen (Kanäle) sowie die Liste der Basisbandverbindungen abfragen kann:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

btsockstat(1) ist ein weiteres Diagnoseprogramm. Es funktioniert analog zu netstat(1), arbeitet aber mit Bluetooth-Datenstrukturen. Das folgende Beispiel zeigt die gleiche Liste der logischen Verbindungen wie l2control(8) im vorherigen Beispiel.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

32.4.5. Das RFCOMM-Protokoll

Das RFCOMM-Protokoll emuliert serielle Verbindungen über das L2CAP-Protokoll. Es basiert auf dem ETSI-Standard TS 07.10. Bei RFCOMM handelt es sich um ein einfaches Transportprotokoll, das um Funktionen zur Emulation der 9poligen Schaltkreise von mit RS-232 (EIATIA-232-E) kompatiblen seriellen Ports ergänzt wurde. RFCOMM erlaubt bis zu 60 simultane Verbindungen (RFCOMM-Kanäe) zwischen zwei Bluetooth-Geräten.

Eine RFCOMM-Kommunikation besteht aus zwei Anwendungen (den Kommunikationsendpunkten), die über das Kommunikationssegment miteinander verbunden sind. RFCOMM unterstützt Anwendungen, die auf serielle Ports angewiesen sind. Das Kommunikationssegment entspricht der (direkten) Bluetooth-Verbindung zwischen den beiden Geräten.

RFCOMM kümmert sich um die direkte Verbindung von zwei Geräten, oder um die Verbindung zwischen einem Gerät und einem Modem (Netzwerkverbindung). RFCOMM unterstützt auch andere Konfigurationen. Ein Beispiel dafür sind Module, die drahtlose Bluetooth-Geräte mit einer verkabelten Schnittstelle verbinden können.

Unter FreeBSD wurde das RFCOMM-Protokoll im Bluetooth Socket-Layer implementiert.

32.4.6. Erstmaliger Verbindungsaufbau zwischen zwei Bluetooth-Geräten (*Pairing*)

In der Voreinstellung nutzt Bluetooth keine Authentifizierung, daher kann sich jedes Bluetoothgerät mit jedem anderen Gerät verbinden. Ein Bluetoothgerät (beispielsweise ein Mobiltelefon) kann jedoch für einen bestimmten Dienst (etwa eine Einwählverbindung) eine Authentifizierung anfordern. Bluetooth verwendet zu diesem Zweck *PIN-Codes*. Ein PIN-Code ist ein maximal 16 Zeichen langer ASCII-String. Damit eine Verbindung zustande kommt, muss auf beiden Geräten der gleiche PIN-Code verwendet werden. Nachdem der Code eingegeben wurde, erzeugen beide Geräte einen *link key*, der auf den Geräten gespeichert wird. Beim nächsten Verbindungsaufbau wird der zuvor erzeugte Link Key verwendet. Diesen Vorgang bezeichnet man als *Pairing*. Geht der Link Key auf einem Gerät verloren, muss das Pairing wiederholt werden.

Der hcsecd(8)-Daemon verarbeitet alle Bluetooth-Authentifzierungsanforderungen und wird über die Datei `/etc/bluetooth/hcsecd.conf` konfiguriert. Der folgende Ausschnitt dieser Datei zeigt die Konfiguration für ein Mobiltelefon, das den PIN-Code “1234” verwendet:

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

Von der Länge abgesehen, unterliegen PIN-Codes keinen Einschränkungen. Einige Geräte, beispielsweise Bluetooth-Headsets, haben einen festen PIN-Code eingebaut. Die Option `-d` sorgt dafür, dass der hcsecd(8)-Daemon im Vordergrund läuft. Dadurch kann der Ablauf einfach verfolgt werden. Stellen Sie das entfernte Gerät auf *receive pairing* und initiieren Sie die Bluetoothverbindung auf dem entfernten Gerät. Sie erhalten die Meldung, dass Pairing akzeptiert wurde und der PIN-Code benötigt wird. Geben Sie den gleichen PIN-Code ein, den Sie in `hcsecd.conf` festgelegt haben. Ihr Computer und das entfernte Gerät sind nun miteinander verbunden. Alternativ können Sie das Pairing auch auf dem entfernten Gerät initiieren.

hcsecd kann durch das Einfügen der folgenden Zeile in `/etc/rc.conf` beim Systemstart automatisch aktiviert werden:

```
hcsecd_enable="YES"
```

Es folgt nun eine beispielhafte Ausgabe des **hcsecd**-Daemons:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', link key do
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', PIN code ex
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

32.4.7. Das Service Discovery Protocol (SDP)

Das *Service Discovery Protocol* (SDP) erlaubt es Clientanwendungen, von Serveranwendungen angebotene Dienste sowie deren Eigenschaften abzufragen. Zu diesen Eigenschaften gehören die Art oder die Klasse der angebotenen Dienste sowie der Mechanismus oder das Protokoll, die zur Nutzung des Dienstes notwendig sind.

SDP ermöglicht Verbindungen zwischen einem SDP-Server und einem SDP-Client. Der Server enthält eine Liste mit den Eigenschaften der vom Server angebotenen Dienste. Jeder Eintrag beschreibt jeweils einen einzigen Serverdienst. Ein Client kann diese Informationen durch eine SDP-Anforderung vom SDP-Server beziehen. Wenn der Client oder eine Anwendung des Clients einen Dienst nutzen will, muss eine seperate Verbindung mit dem Dienstanbieter aufgebaut werden. SDP bietet einen Mechanismus zum Auffinden von Diensten und deren Eigenschaften an, es bietet aber keine Mechanismen zur Verwendung dieser Dienste.

Normalerweise sucht ein SDP-Client nur nach Diensten, die bestimmte geforderte Eigenschaften erfüllen. Es ist aber auch möglich, anhand der Dienstbeschreibungen eine allgemeine Suche nach den von einem Server angebotenen Diensten durchzuführen. Diesen Vorgang bezeichnet man als *Browsing*.

Der Bluetooth-SDP-Server sdpd(8) und der Kommandozeilenclient sdpcontrol(8) sind bereits in der Standardinstallation von FreeBSD enthalten. Das folgende Beispiel zeigt, wie eine SDP-Abfrage durchgeführt wird:

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
        LAN Access Using PPP (0x1102) ver. 1.0
```

... und so weiter. Beachten Sie, dass jeder Dienst eine Liste seiner Eigenschaften (etwa den RFCOMM-Kanal) zurückgibt. Je nach dem, welche Dienste Sie benötigen, sollten Sie sich einige dieser Eigenschaften notieren. Einige Bluetooth-Implementationen unterstützen kein *Service Browsing* und geben daher eine leere Liste zurück. Ist dies der Fall, ist es dennoch möglich, nach einem bestimmten Dienst zu suchen. Das folgende Beispiel demonstriert die Suche nach dem OBEX Object Push (OPUSH) Dienst:

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Unter FreeBSD ist es die Aufgabe des sdpd(8)-Servers, Bluetooth-Clients verschiedene Dienste anzubieten. Sie können diesen Server durch das Einfügen der folgenden Zeile in die Datei `/etc/rc.conf` aktivieren:

```
sdpd_enable="YES"
```

Nun kann der **sdpd**-Daemon durch folgene Eingabe gestartet werden:

```
# /etc/rc.d/sdpd start
```

Der lokale Server, der den entfernten Clients Bluetooth-Dienste anbieten soll, bindet diese Dienste an den lokalen SDP-Daemon. Ein Beispiel für eine solche Anwendung ist rfcomm_pppd(8). Einmal gestartet, wird der Bluetooth-LAN-Dienst an den lokalen SDP-Daemon gebunden.

Die Liste der vorhandenen Dienste, die am lokalen SDP-Server registriert sind, lässt sich durch eine SDP-Abfrage über einen lokalen Kontrollkanal abfragen:

```
# sdpcontrol -l browse
```

32.4.8. Einwahlverbindungen (Dial-Up Networking (DUN)) oder Netzwerkverbindungen mit PPP (LAN)-Profilen einrichten

Das *Dial-Up Networking (DUN)*-Profil wird vor allem für Modems und Mobiltelefone verwendet. Dieses Profil ermöglicht folgende Szenarien:

- Die Verwendung eines Mobiltelefons oder eines Modems durch einen Computer als drahtloses Modem, um sich über einen Einwahlprovider mit dem Internet zu verbinden oder andere Einwahldienste zu benutzen.
- Die Verwendung eines Mobiltelefons oder eines Modems durch einen Computers, um auf Datenabfragen zu reagieren.

Der Zugriff auf ein Netzwerk über das PPP (LAN)-Profil kann in folgenden Situationen verwendet werden:

- Den LAN-Zugriff für ein einzelnes Bluetooth-Gerät
- Den LAN-Zugriff für mehrere Bluetooth-Geräte
- Eine PC-zu-PC-Verbindung (unter Verwendung einer PPP-Verbindung über eine emulierte serielle Verbindung)

Beide Profile werden unter FreeBSD durch ppp(8) sowie rfcomm_pppd(8) implementiert - einem Wrapper, der RFCOMM Bluetooth-Verbindungen unter PPP nutzbar macht. Bevor ein Profil verwendet werden kann, muss ein neuer PPP-Abschnitt in `/etc/ppp/ppp.conf` erzeugt werden. Beispielkonfigurationen zu diesem Thema finden Sie in rfcomm_pppd(8).

Im folgenden Beispiel verwenden wir rfcomm_pppd(8), um eine RFCOMM-Verbindung zu einem entfernten Gerät mit der `BD_ADDR 00:80:37:29:19:a4` auf dem RFCOMM-Kanal `DUN` aufzubauen. Die aktuelle

RFCOMM-Kanalnummer erhalten Sie vom entfernten Gerät über SDP. Es ist auch möglich, manuell einen RFCOMM-Kanal festzulegen. In diesem Fall führt rfcomm_pppd(8) keine SDP-Abfrage durch. Verwenden Sie sdpcontrol(8), um die RFCOMM-Kanäle des entfernten Geräts herauszufinden.

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Der sdpd(8)-Server muss laufen, damit ein Netzzugriff mit dem PPP (LAN)-Profil möglich ist. Außerdem muss für den LAN-Client ein neuer Eintrag in `/etc/ppp/ppp.conf` erzeugt werden. Beispielkonfigurationen zu diesem Thema finden Sie in rfcomm_pppd(8). Danach starten Sie den RFCOMM PPP-Server über eine gültige RFCOMM-Kanalnummer. Der RFCOMM PPP-Server bindet dadurch den Bluetooth-LAN-Dienst an den lokalen SDP-Daemon. Das folgende Beispiel zeigt Ihnen, wie man den RFCOMM PPP-Server startet.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

32.4.9. Das Profil OBEX-Push (OPUSH)

OBEX ist ein häufig verwendetes Protokoll für den Dateitransfer zwischen Mobilgeräten. Sein Hauptzweck ist die Kommunikation über die Infrarotschnittstelle. Es dient daher zum Datentransfer zwischen Notebooks oder PDAs sowie zum Austausch von Visitenkarten oder Kalendereinträgen zwischen Mobiltelefonen und anderen Geräten mit PIM-Funktionen.

Server und Client von OBEX werden durch das Softwarepaket **obexapp** bereitgestellt, das als Port `comms/obexapp` verfügbar ist.

Mit dem OBEX-Client werden Objekte zum OBEX-Server geschickt oder angefordert. Ein Objekt kann etwa eine Visitenkarte oder ein Termin sein. Der OBEX-Client fordert über SDP die Nummer des RFCOMM-Kanals vom entfernten Gerät an. Dies kann auch durch die Verwendung des Servicenamens anstelle der RFCOMM-Kanalnummer erfolgen. Folgende Dienste werden unterstützt: IrMC, FTRN und OPUSH. Es ist möglich, den RFCOMM-Kanal als Nummer anzugeben. Es folgt nun ein Beispiel für eine OBEX-Sitzung, bei der ein Informationsobjekt vom Mobiltelefon angefordert und ein neues Objekt (hier eine Visitenkarte) an das Telefonbuch des Mobiltelefons geschickt wird:

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Um OBEX-Push-Dienste anbieten zu können, muss der **sdpd**-Server gestartet sein. Ein Wurzelverzeichnis, in dem alle ankommenden Objekt gespeichert werden, muss zusätzlich angelegt werden. In der Voreinstellung ist dies `/var/spool/obex`. Starten Sie den OBEX-Server mit einer gültigen Kanalnummer. Der OBEX-Server registriert nun den OBEX-Push-Dienst mit dem lokalen SDP-Daemon. Um den OBEX-Server zu starten, geben Sie Folgendes ein:

```
# obexapp -s -C 10
```

32.4.10. Das Profil Serial-Port (SPP)

Durch dieses Profil können Bluetooth-Geräte RS232- (oder damit kompatible) serielle Kabelverbindungen emulieren. Anwendungen sind dadurch in der Lage, über eine virtuelle serielle Verbindung Bluetooth als Ersatz für eine Kabelverbindung zu nutzen.

Das Profil Serial-Port wird durch rfcomm_sppd(1) verwirklicht. Pseudo-tty wird hier als virtuelle serielle Verbindung verwendet. Das folgende Beispiel zeigt, wie man sich mit einem entfernten Serial-Port-Dienst verbindet. Beachten Sie, dass Sie den RFCOMM-Kanal nicht angeben müssen, da rfcomm_sppd(1) diesen über SDP vom entfernten Gerät abfragen kann. Wenn Sie dies nicht wollen, können Sie einen RFCOMM-Kanal auch manuell festlegen.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/ttyp6
rfcomm_sppd[94692]: Starting on /dev/ttyp6...
```

Sobald die Verbindung hergestellt ist, kann pseudo-tty als serieller Port verwenden werden.

```
# cu -l ttyp6
```

32.4.11. Problembehandlung

32.4.11.1. Ein entferntes Gerät kann keine Verbindung aufbauen

Einige ältere Bluetooth-Geräte unterstützen keinen Rollentausch. Wenn FreeBSD eine neue Verbindung akzeptiert, wird versucht, die Rolle zu tauschen, um zum Master zu werden. Geräte, die dies nicht unterstützen, können keine Verbindung aufbauen. Beachten Sie, dass der Rollentausch ausgeführt wird, sobald eine neue Verbindung aufgebaut wird, daher ist es nicht möglich, das entfernte Gerät zu fragen, ob es den Rollentausch unterstützt. Dieses Verhalten von FreeBSD kann aber durch eine HCI-Option geändert werden:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

32.4.11.2. Wo finde ich genaue Informationen darüber, was schiefgelaufen ist?

Verwenden Sie **hcidump**, das Sie über den Port `comms/hcidump` installieren können. **hcidump** hat Ähnlichkeiten mit tcpdump(1). Es dient zur Anzeige der Bluetooth-Pakete in einem Terminal oder zur Speicherung der Pakete in einer Datei (Dump).

32.5. LAN-Kopplung mit einer Bridge

Geschrieben von Andrew Thompson.

32.5.1. Einführung

Manchmal ist es nützlich, ein physikalisches Netzwerk (wie ein Ethernetsegment) in zwei separate Netzwerke aufzuteilen, ohne gleich IP-Subnetze zu erzeugen, die über einen Router miteinander verbunden sind. Ein Gerät, das zwei Netze auf diese Weise verbindet, wird als *Bridge* bezeichnet. Jedes FreeBSD-System mit zwei Netzwerkkarten kann als Bridge fungieren.

Die Bridge arbeitet, indem sie die MAC Layeradressen (Ethernet Adressen) der Geräte in ihren Netzwerksegmenten lernt. Der Verkehr wird nur dann zwischen zwei Segmenten weitergeleitet, wenn sich Sender und Empfänger in verschiedenen Netzwerksegmenten befinden.

In vielerlei Hinsicht entspricht eine Bridge daher einem Ethernet-Switch mit sehr wenigen Ports.

32.5.2. Situationen, in denen *Bridging* angebracht ist

Es gibt zahlreiche Situationen, in denen der Einsatz einer Bridge sinnvoll ist:

32.5.2.1. Verbinden von Netzwerken

Die Hauptaufgabe einer Bridge ist die Verbindung von zwei oder mehreren Netzwerksegmenten zu einem gemeinsamen Netzwerk. Es ist oft sinnvoller, eine hostbasierte Bridge anstelle normaler Netzwerkkomponenten (wie Kabelverbindungen), Firewalls oder Pseudonetzwerken über die Schnittstelle einer virtuellen Maschine einzusetzen. Eine Bridge kann außerdem ein drahtloses Gerät mit einem Kabelnetzwerk verbinden. Diese Fähigkeit der Bridge wird als *HostAP-Modus* bezeichnet. Die Bridge agiert in diesem Fall als Access Point für das drahtlose Gerät.

32.5.2.2. Filtering/Traffic Shaping Firewall

Häufig kommt es vor, dass Firewallfunktionen benötigt werden, ohne dass Routing oder *Network Adress Translation* (NAT) verwendet werden soll.

Ein Beispiel dafür wäre ein kleines Unternehmen, das über DSL oder ISDN an seinen ISP angebunden ist. Es verfügt über 13 weltweit erreichbare IP-Adressen, sein Netzwerk besteht aus 10 Rechnern. In dieser Situation ist der Einsatz von Subnetzen sowie einer routerbasierten Firewall schwierig.

Eine brigdebasierte Firewall kann konfiguriert und in den ISDN/DSL-Downstreampfad ihres Routers eingebunden werden, ohne dass Sie sich um IP-Adressen kümmern müssen.

32.5.2.3. Netzwerküberwachung

Eine Bridge kann zwei Netzwerksegmente miteinander verbinden und danach alle Ethernet-Rahmen überprüfen, die zwischen den beiden Netzwerksegmenten ausgetauscht werden. Dazu verwendet man entweder bpf(4)/tcpdump(1) auf dem Netzgerät der Bridge oder schickt Kopien aller Rahmen an ein zusätzliches Netzgerät (den sogenannten *Span Port*).

32.5.2.4. Layer 2-VPN

Zwei Ethernetnetzwerke können über einen IP-Link miteinander verbunden werden, indem Sie die beiden Netzwerke über einen EtherIP-Tunnel koppeln oder eine tap(4)-basierte Lösung wie OpenVPN einsetzen.

32.5.2.5. Layer 2-Redundanz

Die Systeme eines Netzwerks können redundant miteinander verbunden sein. In diesem Fall verwenden Sie das *Spanning Tree Protocol*, um redundante Pfade zu blockieren. Damit ein Ethernetnetzwerk korrekt arbeitet, darf immer nur ein aktiver Pfad zwischen zwei Geräten des Netzwerks existieren. Aufgabe des Spanning Tree Protocols ist es daher, Schleifen zu entdecken und redundante Links in den Status *blockiert* zu versetzen. Fällt ein aktiver Link

aus, so berechnet das Protokoll einen neuen Pfad. Dazu wird ein blockierter Pfad in den Status *aktiv* versetzt, damit alle Systeme des Netzwerks wieder miteinander kommunizieren können.

32.5.3. Kernelkonfiguration

Dieser Abschnitt beschreibt nur die if_bridge(4)-Bridge-Implementierung. Ein Netgraph-Bridge-Treiber ist ebenfalls verfügbar, wird hier aber nicht behandelt. Lesen Sie die Manualpage ng_bridge(4), wenn Sie diesen Treiber einsetzen wollen.

Bei diesem Treiber handelt es sich um ein Kernelmodul, das von ifconfig(8) automatisch geladen wird, wenn ein Bridge-Interface erzeugt wird. Alternativ ist es aber auch möglich, die Unterstützung für den Treiber in Ihren Kernel zu kompilieren. Dazu fügen Sie die Zeile `device if_bridge` in Ihre Kernelkonfigurationsdatei ein und bauen danach den Kernel neu.

Paketfilter können mit allen Firewallpaketen verwendet werden, die das pfil(9)-Framework benutzen. Die Firewall kann dabei entweder als Kernelmodul geladen oder in den Kernel kompiliert werden.

Eine Bridge kann auch als *Traffic Shaper* verwendet werden, wenn Sie altq(4) oder dummynet(4) einsetzen.

32.5.4. Die LAN-Kopplung aktivieren

Eine Bridge wird durch das Klonen von Schnittstellen erzeugt. Um eine Bridge zu erzeugen, verwenden Sie den Befehl ifconfig(8). Ist der Bridge-Treiber nicht in Ihren Kernel kompiliert, wird er automatisch geladen.

```
# ifconfig bridge create
bridge0
# ifconfig bridge0
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

Im obigen Beispiel wird die Bridge erzeugt und erhält automatisch eine zufällig generierte Ethernet-Adresse zugewiesen. Die Parameter `maxaddr` sowie `timeout` legen fest, wie viele MAC-Adressen die Bridge in ihrer Forward-Tabelle halten kann beziehungsweise wie viele Sekunden jeder Eintrag erhalten bleiben soll, nachdem er zuletzt verwendet wurde. Die restlichen Parameter sind für die Konfiguration von Spanning Tree notwendig.

Im nächsten Schritt werden die Schnittstellen, die die Bridge verbinden soll, zugewiesen. Damit die Bridge Datenpakete weiterleiten kann, müssen sowohl die Bridge als auch die Schnittstellen (der zu verbindenden Netzwerksegmente) aktiviert sein:

```
# ifconfig bridge0 addm fxp0 addm fxp1 up
# ifconfig fxp0 up
# ifconfig fxp1 up
```

Danach ist die Bridge in der Lage, Ethernet-Rahmen zwischen den Schnittstellen `fxp0` und `fxp1` weiterzuleiten. Um diese Konfiguration beim Systemstart automatisch zu aktivieren, müssen Sie folgende Einträge in die Datei `/etc/rc.conf` aufnehmen:

```
cloned_interfaces="bridge0"
```

```
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

Benötigen Sie für die Bridge eine IP-Adresse, müssen Sie diese der Schnittstelle der Bridge zuweisen (und nicht einer der Schnittstellen der gekoppelten Netzwerksegmente). Dabei können Sie die IP-Adresse sowohl statisch als auch dynamisch über DHCP zuweisen:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

Sie können der Bridge-Schnittstelle auch eine IPv6-Adresse zuweisen.

32.5.5. Firewalls

Nachdem ein Paketfilter aktiviert wurde, können Datenpakete, die von den Schnittstellen der gekoppelten Netzwerksegmente gesendet und empfangen werden, über die Bridge weitergeleitet oder nach bestimmten Regeln gefiltert oder auch komplett geblockt werden. Ist die Richtung des Paketflusses wichtig, ist es am besten, eine Firewall auf den Schnittstellen der einzelnen Netzwerksegmente einzurichten und nicht auf der Bridge selbst.

Eine Bridge verfügt über verschiedene Optionen, über die Sie die Weiterleitung von Nicht-IP- und ARP-Paketen sowie den Einsatz von Layer 2-Firewalls (mit IPFW) steuern können. Lesen Sie die Manualpage if_bridge(4), wenn Sie diese Funktionen benötigen.

32.5.6. Spanning Tree

Der Bridge-Treiber implementiert das *Rapid Spanning Tree Protocol* (RSTP oder 802.1w), das abwärtskompatibel zum veralteten *Spanning Tree Protocol* (STP) ist. Spanning Tree dient dazu, Schleifen in einer Netzwerktopologie zu entdecken und zu entfernen. RSTP arbeitet dabei schneller als das veraltete STP. RSTP tauscht Informationen mit benachbarten Switchen aus, um Pakete korrekt weiterzuleiten und eine Schleifenbildung zu verhindern.

FreeBSD unterstützt die Betriebsmode RSTP sowie STP, von denen RSTP als Standardmodus voreingestellt ist.

Spanning Tree kann auf den Schnittstellen der durch die Bridge verbundenen Netzwerksegmente über die Option `stp` aktiviert werden. Für eine Bridge, die die Schnittstellen `fxp0` und `fxp1` verbindet, aktivieren Sie STP wie folgt:

```
# ifconfig bridge0 stp fxp0 stp fxp1
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether d6:cf:d5:a0:94:6d
        id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 3 priority 128 path cost 200000 proto rstp
                role designated state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

Diese Bridge hat die Spanning-Tree-ID `00:01:02:4b:d4:50` und die Priorität `32768`. Da diese ID mit der `Root-ID` identisch ist, handelt es sich um die Root-Bridge dieses Netzwerks.

Auf einer anderen Bridge des Netzwerks ist Spanning Tree ebenfalls aktiviert:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role root state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 5 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

Die Zeile `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` zeigt an, dass die Root-Bridge wie im obigen Beispiel die ID `00:01:02:4b:d4:50` hat. Die Pfadkosten hin zur Root-Bridge betragen `400000`, wobei der Pfad zur Root-Bridge über `Port 4` geht (der wiederum der Schnittstelle `fxp0` entspricht).

32.5.7. Fortgeschrittene Funktionen

32.5.7.1. Den Datenfluss rekonstruieren

Die Bridge unterstützt den Monitormodus. Dabei werden alle Pakete verworfen, nachdem sie von bpf(4) verarbeitet wurden. In diesem Modus erfolgt keine weitere Bearbeitung und auch keine Weiterleitung von Datenpaketen. Es ist daher möglich, die Eingabe von zwei oder mehr Netzwerkschnittstellen in einen einzigen gemeinsamen bpf(4)-Stream zu vereinen. Ein solcher Datenstrom ist beispielsweise nützlich, um den Datenverkehr für ""network taps"" zu rekonstruieren, die ihre RX/TX-Signale über verschiedene Schnittstellen senden.

Um die Eingabe von vier Netzwerkschnittstellen in einzigen gemeinsamen Datenstrom zu vereinen, geben Sie Folgendes ein:

```
# ifconfig bridge0 addm fxp0 addm fxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

32.5.7.2. Span Ports

Eine Kopie jedes Ethernet-Rahmens, der an der Bridge ankommt, wird über einen festgelegten *Span Port* verschickt. Auf einer Bridge können beliebig viele Span Ports festgelegt werden. Wird eine Schnittstelle als Span Port konfiguriert, kann sie nicht mehr als normaler Bridge-Port verwendet werden. Eine derartige Konfiguration ist beispielsweise sinnvoll, um den Datenverkehr, der in einem Netzwerk über die Bridge läuft, auf einen Rechner zu übertragen, der mit einem Span Port der Bridge verbunden ist.

Um eine Kopie aller Ethernet-Rahmen über die Schnittstelle `fxp4` zu verschicken, geben Sie Folgendes ein:

```
# ifconfig bridge0 span fxp4
```

32.5.7.3. Private Schnittstellen

Eine private Schnittstelle leitet keine Daten an einen Port weiter, bei dem es sich ebenfalls um eine private Schnittstelle handelt. Der Datenverkehr wird dabei komplett blockiert, auch Ethernet-Rahmen und ARP-Pakete

werden nicht weitergeleitet. Wollen Sie hingegen nur spezifische Datenpakete blockieren, sollten Sie eine Firewall einsetzen.

32.5.7.4. Schnittstellen als *sticky* kennzeichnen

Wenn die Schnittstelle eines über eine Bridge verbundenen Netzwerksegments als *sticky* gekennzeichnet wird, werden alle dynamisch gelernten Adressen als statische Adressen behandelt, sobald sie in den Forward-Cache der Bridge aufgenommen wurden. Sticky-Einträge werden niemals aus dem Cache entfernt oder ersetzt. Selbst dann nicht, wenn die Adresse von einer anderen Schnittstelle verwendet wird. Sie können dadurch die Vorteile statischer Adresseinträge nutzen, ohne die Forward-Tabelle vor dem Einsatz der Bridge mit statischen Einträgen füllen zu müssen. Clients, die sich in einem bestimmten von der Bridge verwalteten Segmente befinden, können dabei nicht in ein anderes Segment wechseln.

Ein weiteres Beispiel für den Einsatz von Sticky-Adressen wäre die Kombination einer Bridge mit mehreren VLANs, um einen Router zu konfigurieren, der in in der Lage ist, einzelne Kundennetzwerke voneinander zu trennen, ohne IP-Adressbereiche zu verschwenden. Für das folgende Beispiel nehmen wir an, dass sich der Client `CustomerA` im VLAN `vlan100` und der Client `CustomerB` im VLAN `vlan101` befinden. Die Bridge hat die IP-Adresse `192.168.0.1` und ist als Internet-Router konfiguriert.

```
# ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101 sticky vlan101
# ifconfig bridge0 inet 192.168.0.1/24
```

Beide Clients sehen `192.168.0.1` als Ihr Default-Gateway. Da der Brücken-Cache *sticky* ist, sind Sie nicht dazu in der Lage, die MAC-Adresse des anderen Kunden zu spoofen und dessen Datenverkehr abzufangen.

Sie können die Kommunikation zwischen den VLANs vollständig unterbinden, wenn Sie private Schnittstellen (oder eine Firewall) einsetzen:

```
# ifconfig bridge0 private vlan100 private vlan101
```

Die Kunden sind nun komplett voneinander isoliert und der komplette `/24`-Adressbereich kann zugewiesen werden, ohne dass Sie Subnetze einsetzen müssen.

32.5.7.5. Adressen-Limitierung

Die maximale mögliche Anzahl an eindeutigen MAC-Adressen hinter einer Schnittstelle kann festgelegt werden. Sobald das Limit erreicht ist, werden Pakete mit einer unbekannten Quell-Adresse solange verworfen, bis ein exisitierender Eintrag gelöscht wird oder abläuft.

Das folgende Beispiel setzt die maximale Anzahl von Netzgeräten für `CustomerA` für das VLAN `vlan100` auf 10.

```
# ifconfig bridge0 ifmaxaddr vlan100 10
```

32.5.7.6. SNMP-Monitoring

Die Schnittstelle der Bridge sowie die STP-Parameter können durch den bereits im Basissystem enthaltenen SNMP-Daemon überwacht werden. Die exportierten Bridge-MIBs entsprechen den IETF-Standards, daher können Sie einen beliebigen SNMP-Client oder ein beliebiges Monitoring-Werkzeug einsetzen, um die benötigten Daten zu erhalten.

Auf dem Rechner, auf dem die Bridge konfiguriert ist, aktivieren Sie die Zeile `begemotSnmpdModulePath."bridge" = "/usr/lib/snmp_bridge.so"` in der Datei `/etc/snmp.config` und starten danach den **bsnmpd**-Daemon. Eventuell benötigen Sie noch weitere Konfigurationsparameter wie Community-Namen und Zugriffslisten. Die Konfiguration dieser Parameter wird in den Manualpages bsnmpd(1) sowie snmp_bridge(3) beschrieben.

Die folgenden Beispiele verwenden das Softwarepaket **Net-SNMP** (`net-mgmt/net-snmp`), um die Bridge abzufragen. Alternativ können Sie dafür auch den Port `net-mgmt/bsnmptools` einsetzen. Auf dem SNMP-Client fügen Sie danach die folgenden Zeilen in die Datei `$HOME/.snmp/snmp.conf` ein, um die MIB-Definitionen der Bridge in **Net-SNMP** zu importieren:

```
mibdirs +/usr/share/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

Um eine einzelne Bridge über den IETF BRIDGE-MIB (RFC4188) zu überwachen, geben Sie Folgendes ein:

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centi-seconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
...
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding(5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled(1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp(2)
```

Der Wert der Variable `dot1dStpTopChanges.0` ist hier 2, die STP-Topologie der Bridge wurde also bereits zweimal geändert. Unter einer Änderung versteht man dabei die Anpassung eines oder mehrerer Links und die Kalkulation eines neuen Baums. Der Wert der Variable `dot1dStpTimeSinceTopologyChange.0` gibt an, wann dies zuletzt geschah.

Um mehrere Bridge-Schnittstellen zu überwachen, können Sie den privaten BEGEMOT-BRIDGE-MIB einsetzen:

```
% snmpwalk -v 2c -c public bridge1.example.com
enterprises.fokus.begemot.begemotBridge
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge0" = STRING: bridge0
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge2" = STRING: bridge2
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge0" = STRING: e:ce:3b:5a:9e:13
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge2" = STRING: 12:5e:4d:74:d:fc
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge0" = INTEGER: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge2" = INTEGER: 1
...
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge0" = Timeticks: (116927) 0:19:
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge2" = Timeticks: (82773) 0:13:4
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge0" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge2" = Counter32: 1
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge0" = Hex-STRING: 80 00 00 40 95 30 5E 3
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge2" = Hex-STRING: 80 00 00 50 8B B8 C6 A
```

Um die über den `mib-2.dot1dBridge`-Subtree überwachte Bridge-Schnittstelle zu ändern, geben Sie Folgendes ein:

```
% snmpset -v 2c -c private bridge1.example.com
BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 s bridge2
```

32.6. Link-Aggregation und Failover

Geschrieben von Andrew Thompson. Übersetzt von Benedict Reuschling und Sharon Bahagi.

32.6.1. Einleitung

Die lagg(4)-Schnittstelle erlaubt die Aggregation von mehreren Netzwerkadaptern als eine virtuelle Schnittstelle mit dem Ziel, Ausfallsicherheit (Failover) und Hochgeschwindigkeitsverbindungen bereitzustellen.

32.6.2. Anwendungsoptionen

Ausfallsicherheit (Failover)

Sendet und empfängt Netzwerkverkehr nur auf dem Masterport. Sollte der Masterport nicht zur Verfügung stehen, wird der nächste aktive Port verwendet. Der zuerst hinzugefügte Adapter wird zum Masterport, jeder weitere Adapter dient als Gerät zur Ausfallsicherheit.

Cisco® Fast EtherChannel®

Cisco Fast EtherChannel (FEC), ist eine statische Konfiguration und handelt weder Aggregation mit der Gegenstelle aus, noch werden Frames zur Überwachung der Verbindung ausgetauscht. Wenn der Switch LACP unterstützt, sollte diese Option auch verwendet werden.

FEC balanciert den ausgehenden Verkehr über die aktiven Ports, basierend auf gehashten Protokollheaderinformationen und akzeptiert eingehenden Verkehr auf jedem aktiven Port. Der Hash enthält die Ethernet-Quell- und Zieladresse, und, falls verfügbar, den VLAN-Tag, sowie die IPv4/IPv6 Quell- und Zieladresse.

LACP

Das IEEE 802.3ad Link-Aggregation Control Protokoll (LACP) und das Marker Protocol. LACP wird eine Menge von aggregierbaren Verbindungen mit der Gegenstelle in einer oder mehreren Link Aggregated Groups (LAG) aushandeln. Jede LAG besteht aus Ports der gleichen Geschwindigkeit, eingestellt auf Voll-Duplex-Betrieb. Der Verkehr wird über die Ports in der LAG mit der größten Gesamtgeschwindigkeit balanciert, in den meisten Fällen wird es nur eine LAG geben, die alle Ports enthält. Im Falle von Änderungen in der physischen Anbindung wird die Link-Aggregation schnell zu einer neuen Konfiguration konvergieren.

LACP balanciert ausgehenden Verkehr über die aktiven Ports basierend auf der gehashten Protokollheaderinformation und akzeptiert eingehenden Verkehr auf jedem aktiven Port. Der Hash beinhaltet

die Ethernet-Quell- und Zieladresse, und, soweit verfügbar, den VLAN-Tag, sowie die IPv4/IPv6 Quell- und Zieladresse.

Lastverteilung (Loadbalance)

Dabei handelt es sich um einen Alias des *FEC*-Modus.

Round-Robin

Verteilt ausgehenden Verkehr mittels einer Round-Robin-Zuteilung über alle aktiven Ports und akzeptiert eingehenden Verkehr auf jedem aktiven Port. Dieser Modus verletzt die Reihenfolge von Ethernet-Frames und sollte mit Vorsicht eingesetzt werden.

32.6.3. Beispiele

Beispiel 32-1. LACP Aggregation mit einem Switch von Cisco®

Dieses Beispiel verbindet zwei Adapter auf einer FreeBSD-Maschine mit dem Switch als eine einzelne, lastverteilte und ausfallsichere Verbindung. Weitere Adapter können hinzugefügt werden, um den Durchsatz zu erhöhen und die Ausfallsicherheit zu steigern. Da die Reihenfolge der Frames bei Ethernet zwingend eingehalten werden muss, fließt auch jeglicher Verkehr zwischen zwei Stationen über den gleichen physischen Kanal, was die maximale Geschwindigkeit der Verbindung auf die eines einzelnen Adapters beschränkt. Der Übertragungsalgorithmus versucht, so viele Informationen wie möglich zu verwenden, um die verschiedenen Verkehrsflüsse zu unterscheiden und balanciert diese über die verfügbaren Adapter.

Fügen Sie auf dem Cisco-Switch die Adapter *`FastEthernet0/1`* und *`FastEthernet0/2`* zu der channel-group *`1`* hinzu:

```
interface FastEthernet0/1
 channel-group 1 mode active
 channel-protocol lacp
!
interface FastEthernet0/2
 channel-group 1 mode active
 channel-protocol lacp
```

Auf der Maschine mit FreeBSD erstellen Sie die lagg(4)-Schnittstelle unter Verwendung von *`fxp0`* und *`fxp1`*:

```
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1
```

Überprüfen Sie den Status der Schnittstelle, indem Sie folgendes eingeben:

```
# ifconfig lagg0
```

Ports, die als *ACTIVE* markiert sind, sind Teil der aktiven Aggregations-Gruppe, die mit dem Switch ausgehandelt wurde und der Verkehr wird über diese übertragen und empfangen. Benutzen Sie die ausführliche Ausgabe von ifconfig(8), um sich die LAG-Identifikatoren anzeigen zu lassen.

```
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        media: Ethernet autoselect
        status: active
        laggproto lacp
```

```
        laggport: fxp1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: fxp0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

Um den Status der Ports auf dem Switch anzuzeigen, geben Sie **`show lacp neighbor`** ein:

```
switch# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

                  LACP port                        Oper    Port     Port
Port      Flags   Priority  Dev ID         Age     Key     Number   State
Fa0/1     SA      32768     0005.5d71.8db8  29s    0x146   0x3      0x3D
Fa0/2     SA      32768     0005.5d71.8db8  29s    0x146   0x4      0x3D
```

Benutzen Sie das Kommando **`show lacp neighbor detail`**, um weitere Informationen zu erhalten.

Beispiel 32-2. Ausfallsicherer Modus

Der ausfallsichere Modus kann verwendet werden, um zu einer zweiten Schnittstelle zu wechseln, sollte die Verbindung mit der Master-Schnittstelle ausfallen. Erstellen und konfigurieren Sie die *`lagg0`*-Schnittstelle mit *`fxp0`* als Master und *`fxp1`* als die sekundäre Schnittstelle:

```
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1
```

Die Schnittstelle wird so ähnlich wie im folgenden aussehen, mit dem großen Unterschied, dass die MAC-Adresse und die Gerätenamen unterschiedlich sein werden:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        media: Ethernet autoselect
        status: active
        laggproto failover
        laggport: fxp1 flags=0<>
        laggport: fxp0 flags=5<MASTER,ACTIVE>
```

Der Verkehr wird auf *`fxp0`* übertragen und empfangen. Wenn die Verbindung auf *`fxp0`* abbricht, so wird *`fxp1`* die Verbindung übernehmen. Sobald die Verbindung auf der Master-Schnittstelle wiederhergestellt ist, wird diese auch wieder als aktive Schnittstelle genutzt.

Beispiel 32-3. Failover Modus zwischen drahtgebundenen und drahtlosen Schnittstellen

Für Laptop-Benutzer ist es normalerweise wünschenswert, wireless als sekundäre Schnittstelle einzurichten, die verwendet wird, wenn die Verbindung via Kabel nicht verfügbar ist. Mit lagg(4) ist es möglich, eine IP-Adresse für die Kabelverbindung zu verwenden. Sie ist leistungsfähig und sicher. Gleichzeitig haben Sie die Möglichkeit Daten über die drahtlose Verbindung zu übertragen.

In dieser Konfiguration, müssen wir die zugrunde liegenden MAC-Adresse der WLAN-Schnittstelle überschreiben, damit sie zur Adresse von lagg(4) passt, welche von der drahtgebundenen Masterschnittstelle vererbt wurde.

In dieser Konfiguration behandeln wir die drahtgebundene Schnittstelle *bge0* als die Master und die drahtlose Schnittstelle *wlan0* als die Failover-Schnittstelle. Die *wlan0* wurde von der *iwn0* mit der MAC-Adresse der kabelgebundenen eingerichtet. Im ersten Schritt erhalten wir die MAC-Adresse der kabelgebundenen Schnittstelle:

```
# ifconfig bge0
bge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
 options=19b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING,VLAN_HWCSUM,TSO4>
 ether 00:21:70:da:ae:37
 inet6 fe80::221:70ff:feda:ae37%bge0 prefixlen 64 scopeid 0x2
 nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
 media: Ethernet autoselect (1000baseT <full-duplex>)
 status: active
```

Sie können *bge0* in ihre tatsächliche ändern und werden eine andere `ether`-Zeile mit der MAC-Adresse ihrer kabelgebundenen Schnittstelle erhalten. Nun ändern wir die zugrunde liegende drahtlose Schnittstelle *iwn0*:

```
# ifconfig iwn0 ether 00:21:70:da:ae:37
```

Starten Sie den Wireless-Schnittstelle, aber ohne IP-Adresse:

```
# ifconfig wlan0 create wlandev iwn0 ssid my_router up
```

Erstellen Sie die lagg(4) Schnittstelle mit *bge0* als Master und *wlan0* als Failover falls notwendig:

```
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport bge0 laggport wlan0
```

Die Schnittstelle sieht ähnlich aus, die Hauptunterschiede werden die MAC-Adresse und die Gerätenamen sein:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:21:70:da:ae:37
        media: Ethernet autoselect
        status: active
        laggproto failover
        laggport: wlan0 flags=0<>
        laggport: bge0 flags=5<MASTER,ACTIVE>
```

Um zu vermeiden, dass Sie dies nach jedem Neustart machen müssen, können Sie etwas in der Art in ihre `/etc/rc.conf` Datei schreiben:

```
ifconfig_bge0="up"
ifconfig_iwn0="ether 00:21:70:da:ae:37"
wlans_iwn0="wlan0"
ifconfig_wlan0="WPA"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport bge0 laggport wlan0 DHCP"
```

32.7. Start und Betrieb von FreeBSD über ein Netzwerk

Aktualisiert von Jean-François Dockès. Reorganisiert und erweitert von Alex Dupre.

FreeBSD kann über ein Netzwerk starten und arbeiten, ohne eine lokale Festplatte zu verwenden, indem es Dateisysteme eines NFS-Servers in den eigenen Verzeichnisbaum einhängt. Dazu sind, von den Standardkonfigurationsdateien abgesehen, keine Systemänderungen nötig. Ein solches System kann leicht installiert werden, da alle notwendigen Elemente bereits vorhanden sind:

- Es gibt mindestens zwei Möglichkeiten, den Kernel über das Netzwerk zu laden:
  - PXE: Das “Preboot eXecution Environment System” von Intel ist eine Art intelligentes Boot-ROM, das in einigen Netzwerkkarten oder Hauptplatinen verwendet wird. Weitere Informationen finden Sie in pxeboot(8).
  - Der Port **Etherboot** (`net/etherboot`) erzeugt ROM-fähigen Code, um einen Kernel über das Netzwerk zu laden. Dieser Code kann entweder auf ein Boot-PROM einer Netzwerkkarte gebrannt werden, was von vielen Netzwerkkarten unterstützt wird. Oder er kann von einer lokalen Diskette, Festplatte oder von einem laufenden MS-DOS-System geladen werden.

- Das Beispielskript `/usr/share/examples/diskless/clone_root` erleichtert die Erzeugung und die Wartung des root-Dateisystems auf dem Server. Das Skript muss wahrscheinlich angepasst werden, dennoch werden Sie schnell zu einem Ergebnis kommen.
- Die Startdateien, die einen plattenlosen Systemstart erkennen und unterstützen, sind nach der Installation in `/etc` vorhanden.
- Dateiauslagerungen können sowohl via NFS als auch auf die lokale Platte erfolgen.

Es gibt verschiedene Wege, einen plattenlosen Rechner einzurichten. Viele Elemente sind daran beteiligt, die fast immer an den persönlichen Geschmack angepasst werden können. Im folgenden Abschnitt wird die Installation eines kompletten Systems beschrieben, wobei der Schwerpunkt auf Einfachheit und Kompatibilität zu den Standardstartskripten von FreeBSD liegt. Das beschriebene System hat folgende Eigenschaften:

- Die plattenlosen Rechner haben ein gemeinsames `/`- sowie ein gemeinsames `/usr`-Dateisystem, die jeweils schreibgeschützt sind.

  Das root-Dateisystem ist eine Kopie eines Standardwurzelverzeichnisses von FreeBSD (üblicherweise das des Servers), bei dem einige Konfigurationsdateien durch für den plattenlosen Betrieb geeignete Versionen ersetzt wurden.

  Für die Bereiche des root-Dateisystems, die beschreibbar sein müssen, werden mit md(4) virtuelle Dateisysteme erzeugt. Dies bedeutet aber auch, dass alle Veränderungen verloren gehen, wenn das System neu gestartet wird.
- Der Kernel wird, in Abhängigkeit von der jeweiligen Situation, entweder von **Etherboot** oder von PXE transferiert und geladen.

> **Achtung:** Das hier beschriebene System ist nicht sicher. Es sollte nur in einem gesicherten Bereich eines Netzwerks verwendet werden und für andere Rechner nicht erreichbar sein.

Alle Informationen in diesem Abschnitt wurden unter FreeBSD 5.2.1-RELEASE getestet.

32.7.1. Hintergrundinformationen

Die Einrichtung von plattenlosen Rechnern ist einfach, aber auch fehleranfällig. Der Grund dafür sind auftretende Fehler, die sich oft nur schwer zuordnen lassen. Unter anderem sind dafür folgende Umstände verantwortlich:

- Kompilierte Optionen haben zur Laufzeit unterschiedliche Auswirkungen.
- Fehlermeldungen sind oft kryptisch oder fehlen vollständig.

Daher ist es nützlich, über die im Hintergrund ablaufenden Mechanismen Bescheid zu wissen. Dadurch wird es einfacher, eventuell auftretende Fehler zu beheben.

Verschiedene Operationen müssen ausgeführt werden, um ein System erfolgreich zu starten:

- Der Rechner benötigt einige Startparameter, wie seine IP-Adresse, die Namen ausführbarer Dateien, den Servernamen sowie den root-Pfad. Für die Übermittlung dieser Informationen wird entweder das DHCP- oder das BOOTP-Protokoll verwendet. Bei DHCP handelt es sich um eine abwärtskompatible Erweiterung von BOOTP, die die gleichen Portnummern und das gleiche Paketformat verwendet.

  Es ist möglich, das System so zu konfigurieren, dass es nur BOOTP verwendet. Das Serverprogramm bootpd(8) ist bereits im FreeBSD-Basissystem enthalten.

  DHCP hat im Vergleich zu BOOTP allerdings mehrere Vorteile (bessere Konfigurationsdateien, die Möglichkeit zur Verwendung von PXE, sowie viele andere, die nicht in direktem Zusammenhang mit dem plattenlosen Betrieb stehen). Dieser Abschnitt beschreibt die Konfiguration mittels DHCP. Wenn möglich, werden aber entsprechende Beispiele für bootpd(8) angeführt. Die Beispielkonfiguration nutzt das Softwarepaket **ISC DHCP**.
- Der Rechner muss ein oder mehrere Programme in den lokalen Speicher laden. Dazu wird entweder TFTP oder NFS verwendet. Die Auswahl zwischen TFTP und NFS erfolgt über das Setzen von verschiedenen Kompilieroptionen. Ein häufig gemachter Fehler ist es, Dateinamen für das falsche Protokoll anzugeben: TFTP transferiert normalerweise alle Dateien aus einem einzigen Verzeichnis des Servers, und erwartet einen Pfad relativ zu diesem Verzeichnis. NFS verlangt hingegen absolute Dateipfade.
- Die möglichen Bootstrap-Programme und der Kernel müssen initialisiert und ausgeführt werden. Dabei gibt es zwei Möglichkeiten:
  - PXE lädt pxeboot(8). Dabei handelt es sich um eine modifizierte Version des FreeBSD-Laders der Boot-Phase drei. Der loader(8) beschafft alle für den Systemstart notwendigen Parameter, und hinterlegt diese in der Kernelumgebung, bevor er die Kontrolle übergibt. Es ist hier möglich, den `GENERIC`-Kernel zu verwenden.
  - **Etherboot** lädt den Kernel hingegen direkt. Dafür müssen Sie allerdings einen Kernel mit spezifischen Optionen erzeugen.

  PXE und **Etherboot** sind zwar im Großen und Ganzen gleichwertig, da der Kernel aber viele Aufgaben an loader(8) übergibt, sollte bevorzugt PXE eingesetzt werden.

  Wenn Ihr BIOS und Ihre Netzwerkkarten PXE unterstützen, sollten Sie es auch verwenden.
- Zuletzt muss der Rechner auf seine Dateisysteme zugreifen können. Dafür wird stets NFS verwendet.

Weitere Informationen finden Sie in diskless(8).

32.7.2. Installationsanweisungen

32.7.2.1. Konfiguration unter Verwendung von ISC DHCP

Der **ISC DHCP**-Server kann Anfragen sowohl von BOOTP als auch von DHCP beantworten.

isc-dhcp 3.1 ist nicht Teil des Basissystems. Sie müssen es daher zuerst installieren. Verwenden Sie dazu den Port `net/isc-dhcp31-server` oder das entsprechende Paket.

Nachdem **ISC DHCP** installiert ist, muss das Programm konfiguriert werden (normalerweise in `/usr/local/etc/dhcpd.conf`). Im folgenden Beispiel verwendet Rechner `margaux` **Etherboot**, während Rechner `corbieres` PXE verwendet:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
  use-host-decl-names on; ❶
  option subnet-mask 255.255.255.0;
  option broadcast-address 192.168.4.255;

  host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
    next-server 192.168.4.4;❷
    filename "/tftpboot/kernel.diskless";❸
    option root-path "192.168.4.4:/data/misc/diskless";❹
  }
 host corbieres {
   hardware ethernet 00:02:b3:27:62:df;
    fixed-address corbieres.example.com;
    next-server 192.168.4.4;
    filename "pxeboot";
    option root-path "192.168.4.4:/data/misc/diskless";
  }
}
```

❶ Diese Option weist **dhcpd** an, den Wert der `host`-Deklaration als Rechnernamen des plattenlosen Rechners zu senden. Alternativ kann man der `host`-Deklaration Folgendes hinzufügen: `option host-name` *`margaux`*

❷ Die Anweisung `next-server` bestimmt den TFTP- oder NFS-Server, von dem der Loader oder der Kernel geladen werden (in der Voreinstellung ist das der DHCP-Server selbst).

❸ Die Anweisung `filename` bestimmt die Datei, die **Etherboot** als nächstes lädt. Das genaue Format hängt von der gewählten Transfermethode ab. **Etherboot** kann sowohl mit NFS als auch mit TFTP kompiliert werden. In der Voreinstellung wird der FreeBSD-Port mit NFS-Unterstützung kompiliert. PXE verwendet TFTP, daher wird im Beispiel ein relativer Dateipfad verwendet. Dies kann aber, je nach Konfiguration des TFTP-Servers,

auch anders sein. Beachten Sie, dass PXE `pxeboot` lädt, und nicht den Kernel. Es ist auch möglich, das Verzeichnis `/boot` einer FreeBSD-CD-ROM von `pxeboot` laden zu lassen. pxeboot(8) kann einen `GENERIC`-Kernel laden, dadurch ist es möglich, PXE von einer entfernten CD-ROM zu starten.

❹ Die Option `root-path` bestimmt den Pfad des root-Dateisystems in normaler NFS-Schreibweise. Wird PXE verwendet, ist es möglich, die IP-Adresse des Rechners wegzulassen, solange nicht die Kerneloption BOOTP aktiviert wird. Der NFS-Server entspricht in diesem Fall dem TFTP-Server.

32.7.2.2. Konfiguration bei Verwendung von BOOTP

Es folgt nun eine der Konfiguration von DHCP entsprechende Konfiguration (für einen Client) für **bootpd**. Zu finden ist die Konfigurationsdatei unter `/etc/bootptab`.

Beachten Sie bitte, dass **Etherboot** mit der Option `NO_DHCP_SUPPORT` kompiliert werden muss, damit BOOTP verwendet werden kann. PXE hingegen *benötigt* DHCP. Der einzige offensichtliche Vorteil von **bootpd** ist, dass es bereits im Basissystem vorhanden ist.

```
.def100:\
  :hn:ht=1:sa=192.168.4.4:vm=rfc1048:\
  :sm=255.255.255.0:\
  :ds=192.168.4.1:\
  :gw=192.168.4.1:\
  :hd="/tftpboot":\
  :bf="/kernel.diskless":\
  :rp="192.168.4.4:/data/misc/diskless":

margaux:ha=0123456789ab:tc=.def100
```

32.7.2.3. Ein Startprogramm unter Verwendung von Etherboot erstellen

Die Internetseite von Etherboot (http://etherboot.sourceforge.net) enthält ausführliche Informationen (http://etherboot.sourceforge.net/doc/html/userman/t1.html), die zwar vor allem für Linux gedacht sind, aber dennoch nützliche Informationen enthalten. Im Folgenden wird daher nur grob beschrieben, wie Sie **Etherboot** auf einem FreeBSD-System einsetzen können.

Als Erstes müssen Sie `net/etherboot` als Paket oder als Port installieren.

Sie können **Etherboot** so konfigurieren, dass TFTP anstelle von NFS verwendet wird, indem Sie die Datei `Config` im Quellverzeichnis von **Etherboot** bearbeiten.

Für unsere Installation verwenden wir eine Startdiskette. Für Informationen zu anderen Methoden (PROM oder MS-DOS-Programme) lesen Sie bitte die Dokumentation zu **Etherboot**.

Um eine Startdiskette zu erzeugen, legen Sie eine Diskette in das Laufwerk des Rechners ein, auf dem Sie **Etherboot** installiert haben. Danach wechseln Sie in das Verzeichnis `src` des **Etherboot**-Verzeichnisbaums und geben Folgendes ein:

```
# gmake bin32/devicetype.fd0
```

`devicetype` hängt vom Typ der Ethernetkarte ab, über die der plattenlose Rechner verfügt. Lesen Sie dazu `NIC` im gleichen Verzeichnis, um den richtigen Wert für *`devicetype`* zu bestimmen.

32.7.2.4. Das System mit PXE starten

In der Voreinstellung lädt der pxeboot(8)-Loader den Kernel über NFS. Soll stattdessen TFTP verwendet werden, muss beim Kompilieren die Option `LOADER_TFTP_SUPPORT` in der Datei `/etc/make.conf` eingetragen sein. Sehen Sie sich die Datei `/usr/share/examples/etc/make.conf` für weitere Anweisungen an.

Es gibt zwei Optionen für `make.conf`, die nützlich sein können, wenn Sie eine plattenlose serielle Konsole einrichten wollen: `BOOT_PXELDR_PROBE_KEYBOARD`, und `BOOT_PXELDR_ALWAYS_SERIAL`.

Um PXE beim Systemstart zu verwenden, müssen Sie im BIOS des Rechner die Option `Über das Netzwerk starten` aktivieren. Alternativ können Sie während der PC-Initialisierung auch eine Funktionstaste drücken.

32.7.2.5. Serverkonfiguration - TFTP und NFS

Wenn Sie PXE oder **Etherboot** so konfiguriert haben, dass diese TFTP verwenden, müssen Sie auf dem Dateiserver **tftpd** aktivieren:

1. Erzeugen Sie ein Verzeichnis, in dem **tftpd** seine Dateien ablegt, beispielsweise `/tftpboot`.
2. Fügen Sie folgende Zeile in `/etc/inetd.conf` ein:

   ```
   tftp    dgram   udp     wait    root  /usr/libexec/tftpd    tftpd -s /tftpboot
   ```

   > **Anmerkung:** Anscheinend benötigen zumindest einige PXE-Versionen die TCP-Version von TFTP. Sollte dies bei Ihnen der Fall sein, fügen Sie eine zweite Zeile ein, in der Sie `dgram udp` durch `stream tcp` ersetzen.

3. Weisen Sie **inetd** an, seine Konfiguration erneut einzulesen (Damit der folgende Befehl funktioniert, muss die Option `inetd_enable="YES"` in der Datei `/etc/rc.conf` vorhanden sein.):

   ```
   # /etc/rc.d/inetd restart
   ```

Sie können das Verzeichnis `/tftpboot` an einem beliebigen Ort auf dem Server ablegen. Stellen Sie aber sicher, dass Sie diesen Ort sowohl in `inetd.conf` als auch in `dhcpd.conf` eingetragen haben.

Außerdem müssen Sie NFS aktivieren und die entsprechenden Verzeichnisse exportieren.

1. Fügen Sie folgende Zeile in `/etc/rc.conf` ein:

   ```
   nfs_server_enable="YES"
   ```

2. Exportieren Sie das Verzeichnis, in dem sich das Wurzelverzeichnis für den plattenlosen Betrieb befindet, indem Sie folgende Zeile in `/etc/exports` einfügen (passen Sie dabei den *mountpoint* an und ersetzen Sie *`margaux corbieres`* durch den Namen Ihres plattenlosen Rechners):

   ```
   /data/misc -alldirs -ro margaux
   ```

3. Weisen sie nun **mountd** an, seine Konfigurationsdatei erneut einzulesen. Wenn Sie NFS erst in der Datei `/etc/rc.conf` aktivieren mussten, sollten Sie stattdessen den Rechner neu starten. Dadurch wird die Konfigurationsdatei ebenfalls neu eingelesen.

   ```
   # /etc/rc.d/mountd restart
   ```

32.7.2.6. Einen plattenlosen Kernel erzeugen

Wenn Sie **Etherboot** verwenden, müssen Sie in die Kernelkonfigurationsdatei Ihres plattenlosen Clients zusätzlich folgende Optionen einfügen:

```
options     BOOTP          # Use BOOTP to obtain IP address/hostname
options     BOOTP_NFSROOT  # NFS mount root file system using BOOTP info
```

Außerdem können Sie die Optionen `BOOTP_NFSV3`, `BOOT_COMPAT` sowie `BOOTP_WIRED_TO` verwenden (sehen Sie sich dazu auch die Datei `NOTES` an).

Die Namen dieser Optionen sind historisch bedingt. Sie ermöglichen eine unterschiedliche Verwendung von DHCP und BOOTP innerhalb des Kernels. Es ist auch möglich, eine strikte Verwendung von BOOTP oder DHCP zu erzwingen.

Erzeugen Sie den neuen Kernel (lesen Sie dazu auch Kapitel 9) und kopieren Sie ihn an den in `dhcpd.conf` festgelegten Ort.

> **Anmerkung:** Wenn Sie PXE verwenden, ist die Erzeugung eines Kernels zwar nicht unbedingt nötig, sie wird allerdings dennoch empfohlen. Die Aktivierung dieser Optionen bewirkt, dass die Anzahl der möglichen DHCP-Anforderungen während des Kernelstarts erhöht wird. Ein kleiner Nachteil sind eventuell auftretende Inkonsistenzen zwischen den neuen Werten und den von pxeboot(8) erhaltenen Werten. Der große Vorteil dieser Variante ist es, dass dabei der Rechnername gesetzt wird, den Sie ansonsten durch eine andere Methode, beispielsweise in einer clientspezifischen `rc.conf`-Datei festlegen müssten.

> **Anmerkung:** Damit der Kernel von **Etherboot** geladen werden kann, müssen *device hints* im Kernel einkompiliert sein. Dazu setzen Sie normalerweise folgende Option in die Kernelkonfigurationsdatei (sehen Sie sich dazu auch die kommentierte Datei `NOTES` an):
>
> ```
> hints         "GENERIC.hints"
> ```

32.7.2.7. Das root-Dateisystem erzeugen

Sie müssen für den plattenlosen Rechner ein root-Dateisystem erzeugen, und zwar an dem in `dhcpd.conf` als `root-path` festgelegten Ort.

32.7.2.7.1. `make world` zum Füllen des Dateisystems einsetzen

Diese schnelle Methode installiert ein komplettes “jungfräuliches” System (und nicht nur ein root-Dateisystem) nach `DESTDIR`. Dazu müssen Sie lediglich das folgende Skript ausführen:

```
#!/bin/sh
export DESTDIR=/data/misc/diskless
mkdir -p ${DESTDIR}
cd /usr/src; make buildworld && make buildkernel
make installworld && make installkernel
cd /usr/src/etc; make distribution
```

Danach müssen Sie noch die dadurch in `DESTDIR` erzeugten Dateien `/etc/rc.conf` sowie `/etc/fstab` Ihren Wünschen anpassen.

32.7.2.8. Den Auslagerungsbereich konfigurieren

Falls nötig, kann eine auf dem NFS-Server liegende Datei als Auslagerungsdatei eingerichtet werden.

32.7.2.8.1. Eine NFS-Auslagerungsdatei einrichten

Der Kernel unterstützt beim Systemstart keine NFS-Auslagerungsdatei. Diese muss daher in den Startskripten aktiviert werden, indem ein beschreibbares Dateisystem eingehängt wird, um dort die Auslagerungsdatei zu erzeugen und zu aktivieren. Um eine Auslagerungsdatei zu erzeugen, gehen Sie wie folgt vor:

```
# dd if=/dev/zero of=/path/to/swapfile bs=1k count=1 oseek=100000
```

Um die Auslagerungsdatei zu aktivieren, fügen Sie folgende Zeile in `rc.conf` ein:

```
swapfile=/path/to/swapfile
```

32.7.2.9. Verschiedenes

32.7.2.9.1. Schreibgeschütztes Dateisystem `/usr`

Wenn am plattenlosen Rechner X läuft, müssen Sie die Konfigurationsdatei von **XDM** anpassen, da Fehlermeldungen in der Voreinstellung auf `/usr` geschrieben werden.

32.7.2.9.2. Der Server läuft nicht unter FreeBSD

Wenn das root-Dateisystem nicht auf einem FreeBSD-Rechner liegt, muss das Dateisystem zuerst unter FreeBSD erzeugt werden. Anschließend wird es beispielsweise mit `tar` oder `cpio` an den gewünschten Ort kopiert.

Dabei kann es Probleme mit den Gerätedateien in `/dev` geben, die durch eine unterschiedliche Darstellung der Major- und Minor-Number von Geräten auf beiden Systemen hervorgerufen werden. Eine Problemlösung besteht darin, das root-Verzeichnis auf einem FreeBSD-Rechner einzuhängen und die Gerätedateien dort mit devfs(5) zu erzeugen.

32.8. ISDN – diensteintegrierendes digitales Netzwerk

Eine gute Quelle für Informationen zu ISDN ist die ISDN-Seite (http://www.alumni.caltech.edu/~dank/isdn/) von Dan Kegel.

Welche Informationen finden Sie in diesem Abschnitt?

- Wenn Sie in Europa leben, könnte der Abschnitt über ISDN-Karten für Sie interessant sein.
- Wenn Sie ISDN hauptsächlich dazu verwenden wollen, um sich über einen Anbieter ins Internet einzuwählen, sollten Sie den Abschnitt über Terminaladapter lesen. Dies ist die flexibelste Methode, die auch die wenigsten Probleme verursacht.
- Wenn Sie zwei Netzwerke miteinander verbinden, oder sich über eine ISDN-Standleitung mit dem Internet verbinden wollen, finden Sie entsprechende Informationen im Abschnitt über Router und Bridges.

Bei der Wahl der gewünschten Lösung sind die entstehenden Kosten ein entscheidender Faktor. Die folgenden Beschreibungen reichen von der billigsten bis zur teuersten Variante.

32.8.1. ISDN-Karten

Beigetragen von Hellmuth Michaelis.

Das ISDN-Subsystem von FreeBSD unterstützt den DSS1/Q.931- (oder Euro-ISDN)-Standard nur für passive Karten. Zusätzlich werden aber auch einige aktive Karten unterstützt, bei denen die Firmware auch andere Signalprotokolle unterstützt; dies schließt auch die erste ISDN-Karte mit Primärmultiplex-Unterstützung mit ein.

isdn4bsd ermöglicht es Ihnen, sich unter Nutzung von *IP over raw HDLC* oder *synchronem PPP* mit anderen ISDN-Routern zu verbinden. Dazu verwenden Sie entweder Kernel-ppp(8) (via `isppp`, einem modifizierten sppp-Treiber), oder Sie benutzen User-ppp(8). Wenn Sie User-ppp(8) verwenden, können Sie zwei oder mehrere ISDN-B-Kanäle bündeln. Im Paket enthalten ist auch ein Programm mit Anrufbeantworterfunktion sowie verschiedene Werkzeuge, wie ein Softwaremodem, das 300 Baud unterstützt.

FreeBSD unterstützt eine ständig wachsende Anzahl von PC-ISDN-Karten, die weltweit erfolgreich eingesetzt werden.

Von FreeBSD unterstützte passive ISDN-Karten enthalten fast immer den ISAC/HSCX/IPAC ISDN-Chipsatz von Infineon (ehemals Siemens). Unterstützt werden aber auch Karten mit Cologne Chip (diese allerdings nur für den ISA-Bus), PCI-Karten mit Winbond W6692 Chipsatz, einige Karten mit dem Tiger 300/320/ISAC Chipsatz sowie einige Karten mit einem herstellerspezifischen Chipsatz, wie beispielsweise die Fritz!Card PCI V.1.0 und die Fritz!Card PnP von AVM.

An aktiven ISDN-Karten werden derzeit die AVM B1 BRI-Karten (ISA und PCI-Version) sowie die AVM T1 PRI-Karten (PCI-Version) unterstützt.

Informationen zu **isdn4bsd** finden Sie auf der Internetseite (http://www.freebsd-support.de/i4b/) von **isdn4bsd**. Dort finden Sie auch Verweise zu Tipps, Korrekturen, sowie weiteren Informationen, wie dem isdn4bsd-Handbuch (http://people.FreeBSD.org/~hm/).

Falls Sie an der Unterstützung eines zusätzlichen ISDN-Protokolls, einer weiteren ISDN-Karte oder an einer anderen Erweiterung von **isdn4bsd** interessiert sind, wenden Sie sich bitte an Hellmuth Michaelis.

Für Fragen zur Installation, Konfiguration und zu sonstigen Problemen von **isdn4bsd** gibt es die Mailingliste freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn).

32.8.2. ISDN-Terminaladapter

Terminaladapter (TA) sind für ISDN, was Modems für analoge Telefonleitungen sind.

Die meisten Terminaladapter verwenden den Standardbefehlssatz für Modems von Hayes (AT-Kommandos) und können daher als Modemersatz verwendet werden.

Ein Terminaladapter funktioniert prinzipiell wie ein Modem, allerdings erfolgt der Verbindungsaufbau um einiges schneller. Die Konfiguration von PPP entspricht dabei exakt der eines Modems. Stellen Sie dabei allerdings die serielle Geschwindigkeit so hoch wie möglich ein.

Der Hauptvorteil bei der Verwendung eines Terminaladapters zur Verbindung mit einem Internetanbieter ist die Möglichkeit zur Nutzung von dynamischem PPP. Da IP-Adressen immer knapper werden, vergeben die meisten Provider keine statischen IP-Adressen mehr. Die meisten Router unterstützen allerdings keine dynamische Zuweisung von IP-Adressen.

Der PPP-Daemon bestimmt die Stabilität und Eigenschaften der Verbindung, wenn Sie einen Terminaladapter verwenden. Daher können Sie unter FreeBSD einfach von einer Modemverbindung auf eine ISDN-Verbindung wechseln, wenn Sie PPP bereits konfiguriert haben. Allerdings bedeutet dies auch, das bereits bestehende Probleme mit PPP auch unter ISDN auftreten werden.

Wenn Sie an maximaler Stabilität interessiert sind, verwenden Sie Kernel-PPP, und nicht das User-PPP.

Folgende Terminaladapter werden von FreeBSD unterstützt:

- Motorola BitSurfer und Bitsurfer Pro
- Adtran

Die meisten anderen Terminaladapter werden wahrscheinlich ebenfalls funktionieren, da die Hersteller von Terminaladaptern darauf achten, dass ihre Produkte den Standardbefehlssatz möglichst gut unterstützen.

Das wirkliche Problem mit einem externen Terminaladapter ist, dass, ähnlich wie bei Modems, eine gute serielle Karte eine Grundvoraussetzung ist.

Sie sollten sich die Anleitung für die Nutzung serieller Geräte unter FreeBSD (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/serial-uart/index.html) ansehen, wenn Sie detaillierte Informationen über serielle Geräte und die Unterschiede zwischen asynchronen und synchronen seriellen Ports benötigen.

Ein Terminaladapter, der an einem (asynchronen) seriellen Standardport angeschlossen ist, beschränkt Sie auf 115,2 Kbs. Dies selbst dann, wenn Sie eine Verbindung mit 128 Kbs haben. Um die volle Leistungsfähigkeit von ISDN (128 Kbs) nutzen zu können, müssen Sie den Terminaladapter daher an eine synchrone serielle Karte anschließen.

Kaufen Sie keinen internen Terminaladapter in der Hoffnung, damit das synchron/asynchron-Problem vermeiden zu können. Interne Terminaladapter haben einen (asynchronen) seriellen Standardportchip eingebaut. Der einzige Vorteil interner Terminaladapter ist es, dass Sie ein serielles sowie ein Stromkabel weniger benötigen.

Eine synchrone Karte mit einem Terminaladapter ist mindestens so schnell wie ein autonomer ISDN-Router, und, in Kombination mit einem einfachen 386-FreeBSD-System, wahrscheinlich flexibler.

Die Entscheidung zwischen synchroner Karte/Terminaladapter und einem autonomen ISDN-Router ist beinahe eine religiöse Angelegenheit. Zu diesem Thema gibt es viele Diskussionen in den Mailinglisten. Suchen Sie in den Archiven (http://www.FreeBSD.org/search/index.html) danach, wenn Sie an der kompletten Diskussion interessiert sind.

32.8.3. ISDN-Bridges und Router

ISDN-Bridges und Router sind keine Eigenheit von FreeBSD oder eines anderen Betriebssystems. Für eine vollständigere Beschreibung von Routing und Netzwerkkopplungen mit einer Bridge informieren Sie sich bitte durch

weiterführende Literatur.

In diesem Abschnitt werden die Begriffe Router und Bridge synonym verwendet.

ISDN-Router und Bridges werden immer günstiger und damit auch immer beliebter. Ein ISDN-Router ist eine kleine Box, die direkt an Ihr lokales Ethernet-Netzwerk angeschlossen wird und sich mit einem Router oder einer Bridge verbindet. Die eingebaute Software ermöglicht die Kommunikation über PPP oder andere beliebte Protokolle.

Ein Router ermöglicht einen deutlich höheren Datendurchsatz als ein herkömmlicher Terminaladapter, da er eine vollsynchrone ISDN-Verbindung nutzt.

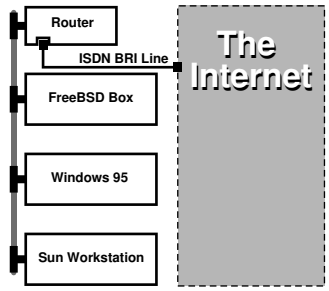
Das Hauptproblem mit ISDN-Routern und Bridges ist, dass die Zusammenarbeit zwischen Geräten verschiedener Hersteller nach wie vor ein Problem ist. Wenn Sie sich auf diese Weise mit einem Internetanbieter verbinden wollen, klären Sie daher vorher ab, welche Anforderungen Ihre Geräte erfüllen müssen.

Eine ISDN-Bridge ist eine einfache und wartungsarme Lösung, zwei Netze, beispielsweise Ihr privates Netz und Ihr Firmennetz, miteinander zu verbinden. Da Sie die technische Ausstattung für beide Seiten kaufen müssen, ist sichergestellt, dass die Verbindung funktionieren wird.

Um beispielsweise einen privaten Computer oder eine Zweigstelle mit dem Hauptnetzwerk zu verbinden, könnte folgende Konfiguration verwendet werden:

Beispiel 32-4. Kleines Netzwerk (Privatnetz)

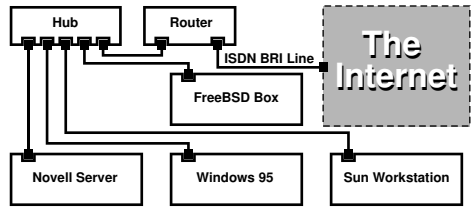
Das Netzwerk basiert auf der Bustopologie mit 10base2 Ethernet (“Thinnet”). Falls nötig, stellen Sie die Verbindung zwischen Router und Netzwerkkabel mit einem AUI/10BT-Transceiver her.



Wenn Sie nur einen einzelnen Rechner verbinden wollen, können Sie auch ein Twisted-Pair-Kabel (Cross-Over) verwenden, das direkt an den Router angeschlossen wird.

Beispiel 32-5. Großes Netzwerk (Firmennetz)

Dieses Netzwerk basiert auf der Sterntopologie und 10baseT Ethernet (“Twisted Pair”).



Ein großer Vorteil der meisten Router und Bridges ist es, dass man *gleichzeitig* zwei *unabhängige* PPP-Verbindungen zu zwei verschiedenen Zielen aufbauen kann. Diese Funktion bieten die meisten Terminaladapter nicht. Die Ausnahme sind spezielle (meist teure) Modelle, die über zwei getrennte serielle Ports verfügen. Verwechseln Sie dies aber nicht mit Kanalbündelung oder MPP.

Dies kann sehr nützlich sein, wenn Sie eine ISDN-Standleitung in Ihrem Büro haben, die sie aufteilen wollen, ohne eine zusätzliche ISDN-Leitung zu installieren. Ein ISDN-Router kann über einen B-Kanal (64 Kbps) eine dedizierte Verbindung ins Internet aufbauen, und gleichzeitig den anderen B-Kanal für eine separate Datenverbindung nutzen. Der zweite B-Kanal kann beispielsweise für ein- oder ausgehende Verbindungen verwendet werden. Sie können ihn aber auch dynamisch mit dem ersten B-Kanal bündeln, um Ihre Bandbreite zu erhöhen.

Eine Ethernet-Bridge kann Daten nicht nur im IP-Protokoll, sondern auch in beliebigen anderen Protokollen versenden.

32.9. NAT - Network Address Translation

Beigetragen von Chern Lee.

32.9.1. Überblick

natd(8), der Network-Address-Translation-Daemon von FreeBSD, akzeptiert ankommende Raw-IP-Pakete, ändert den Sender der Daten in den eigenen Rechner und leitet diese Pakete in den ausgehenden IP-Paketstrom um, indem IP-Adresse und Port des Senders so geändert werden, dass bei einer Antwort der ursprüngliche Sender wieder bestimmt und die Daten an ihn weitergeleitet werden können.

Der häufigste Grund für die Verwendung von NAT ist die gemeinsame Nutzung einer Internetverbindung.

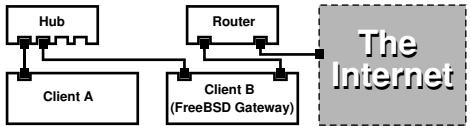
32.9.2. Einrichtung

Wegen der begrenzten Verfügbarkeit von IPv4-Adressen und der gestiegenen Anzahl von Breitbandverbindungen über Kabelmodem oder DSL, wird die gemeinsame Nutzung von Internetverbindungen immer wichtiger. Der natd(8)-Daemon ermöglicht die Anbindung von mehreren Rechnern an das Internet unter Nutzung einer gemeinsamen Verbindung und einer IP-Adresse.

Häufig soll ein über Kabelmodem oder DSL und eine IP-Adresse an das Internet angebundener Rechner mehreren Rechnern eines lokalen Netzwerks Internetdienste anbieten.

Um dies zu ermöglichen, muss der FreeBSD-Rechner als Gateway fungieren. Dazu sind zwei Netzwerkkarten notwendig. Eine für die Verbindung zum Internet, die zweite für die Verbindung mit dem lokalen Netzwerk. Sämtliche Rechner des lokalen Netzwerks sind über einen Hub oder einen Switch miteinander verbunden.

> **Anmerkung:** Es gibt verschiedene Möglichkeiten, ein LAN über ein FreeBSD-Gateway an das Internet anzubinden. Das folgende Beispiel beschreibt ein Gateway, das zumindest zwei Netzwerkkarten enthält.



Eine derartige Netzwerkkonfiguration wird vor allem zur gemeinsamen Nutzung einer Internetverbindung verwendet. Ein Rechner des lokalen Netzwerks (LAN) ist mit dem Internet verbunden. Alle anderen Rechner des lokalen Netzwerks haben nur über diesen “Gateway”-Rechner Zugriff auf das Internet.

32.9.3. Boot Loader Konfiguration

Die Kerneleigenschaften für Network Address Translation mit natd(8) sind im `GENERIC`-Kernel nicht aktiviert, können aber bereits zur Bootzeit geladen werden, indem ein paar Zeilen in die Datei `/boot/loader.conf` hinzugefügt werden:

```
ipfw_load="YES"
ipdivert_load="YES"
```

Zusätzlich kann die Option `net.inet.ip.fw.default_to_accept` auf `1` gesetzt werden:

```
net.inet.ip.fw.default_to_accept="1"
```

> **Anmerkung:** Es ist eine gute Idee, diese Option während den ersten Versuchen, eine Firewall und ein NAT-Gateway aufzusetzen, zu aktivieren. Damit ist die Standardvorgehensweise von ipfw(8) diejenige, `allow ip from any to any`, anstatt der weniger freizügigen `deny ip from any to any`. Es wird dadurch etwas schwieriger, sich aus Versehen nach einem Neustart aus dem System auszusperren.

32.9.4. Kernelkonfiguration

Wenn Module nicht in Frage kommen oder Sie bevorzugen, alle notwendigen Eigenschaften in den laufenden Kernel einzubauen, müssen die folgenden Optionen in die Kernelkonfigurationsdatei eingetragen werden:

```
options IPFIREWALL
options IPDIVERT
```

Die folgende Optionen können ebenfalls eingetragen werden:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

32.9.5. System Bootkonfiguration

Um Firewall- und NAT-Unterstützung zur Bootzeit zu aktivieren, tragen Sie Folgendes in `/etc/rc.conf` ein:

```
gateway_enable="YES" ❶
firewall_enable="YES" ❷
```

```
firewall_type="OPEN" ❸
natd_enable="YES"
natd_interface="fxp0" ❹
natd_flags="" ❺
```

❶ Richtet den Rechner als Gateway ein. Die Ausführung von `sysctl net.inet.ip.forwarding=1` hätte den gleichen Effekt.

❷ Aktiviert die Firewallregeln in `/etc/rc.firewall` beim Systemstart.

❸ Ein vordefinierter Satz von Firewallregeln, der alle Pakete durchlässt. Sehen Sie sich `/etc/rc.firewall` an, wenn Sie diese Option verwenden wollen.

❹ Die Netzwerkkarte, die Pakete weiterleitet (und mit dem Internet verbunden ist).

❺ Zusätzliche Konfigurationsoptionen, die beim Systemstart an natd(8) übergeben werden.

Durch die Definition dieser Optionen in `/etc/rc.conf` wird die Anweisung `natd -interface fxp0` beim Systemstart ausgeführt. Dies kann aber auch manuell erfolgen.

> **Anmerkung:** Falls Sie viele Optionen an natd(8) übergeben müssen, können Sie auch eine Konfigurationsdatei verwenden. Dazu fügen Sie folgende Zeile in `/etc/rc.conf` ein:
>
> ```
> natd_flags="-f /etc/natd.conf"
> ```
>
> Die Datei `/etc/natd.conf` enthält verschiedene Konfigurationsoptionen, wobei jede Option in einer Zeile steht. Das Beispiel im nächsten Abschnitt würde folgende Konfigurationsdatei verwenden:
>
> ```
> redirect_port tcp 192.168.0.2:6667 6667
> redirect_port tcp 192.168.0.3:80 80
> ```
>
> Wenn Sie eine Konfigurationsdatei verwenden wollen, sollten Sie sich die Handbuchseite zu natd(8) durchlesen, insbesondere den Abschnitt über die Nutzung der Option `-f`.

Jedem Rechner und jeder Schnittstelle des lokalen Netzwerks sollte eine IP-Adresse des im RFC 1918 (ftp://ftp.isi.edu/in-notes/rfc1918.txt) definierten privaten Adressraums zugewiesen werden. Der Standardgateway entspricht der internen IP-Adresse des **natd**-Rechners.

Im Beispiel werden den LAN-Clients `A` und `B` die IP-Adressen `192.168.0.2` und `192.168.0.3` zugewiesen, während die LAN-Netzwerkkarte des **natd**-Rechners die IP-Adresse `192.168.0.1` erhält. Der **natd**-Rechner mit der IP-Adresse `192.168.0.1` wird als Standardgateway für die Clients `A` und `B` gesetzt. Die externe Netzwerkkarte des **natd**-Rechners muss für die korrekte Funktion von natd(8) nicht konfiguriert werden.

32.9.6. Ports umleiten

Wenn Sie natd(8) verwenden, sind Ihre LAN-Clients von aussen nicht erreichbar. LAN-Clients können zwar Verbindungen nach aussen aufbauen, sind aber für ankommende Verbindungen nicht erreichbar. Wenn Sie Internetdienste auf einem LAN-Client anbieten wollen, haben Sie daher ein Problem. Eine einfache Lösung ist die Umleitung von bestimmten Internetports des **natd**-Rechners auf einen LAN-Client.

Beispielsweise könnte ein IRC-Server auf Client `A` und ein Webserver auf Client `B` laufen. Damit diese Konfiguration funktioniert, müssen Verbindungen, die auf den Ports 6667 (IRC) und 80 (Web) ankommen, auf die entsprechenden Clients umgeleitet werden.

Dazu wird die Option `-redirect_port` unter Nutzung folgender Syntax an natd(8) übergeben:

```
 -redirect_port proto targetIP:targetPORT[-targetPORT]
             [aliasIP:]aliasPORT[-aliasPORT]
             [remoteIP[:remotePORT[-remotePORT]]]
```

Für unser Beispiel heißt das:

```
-redirect_port tcp 192.168.0.2:6667 6667
-redirect_port tcp 192.168.0.3:80 80
```

Dadurch werden die entsprechenden *tcp*-Ports auf die jeweiligen LAN-Clients umgeleitet.

Mit `-redirect_port` können auch ganze Portbereiche statt einzelner Ports umgeleitet werden. So werden mit *`tcp 192.168.0.2:2000-3000 2000-3000`* alle Verbindungen, die auf den Ports 2000 bis 3000 ankommen, auf die entsprechenden Ports des Clients `A` umgeleitet.

Diese Optionen können während des Betriebs von natd(8) oder über die Option `natd_flags=""` in `/etc/rc.conf` gesetzt werden.

Eine ausführliche Konfigurationsanleitung finden Sie in natd(8).

32.9.7. Adressen umleiten

Die Umleitung von Adressen ist nützlich, wenn mehrere IP-Adressen verfügbar sind, die aber alle auf einem Rechner verbleiben sollen. In diesem Fall kann natd(8) jedem LAN-Client eine eigene externe IP-Adresse zuweisen. Ausgehende Pakete eines LAN-Clients werden so der entsprechenden externen IP-Adresse des Clients zugeordnet. Ankommender Verkehr für diese IP-Adresse wird automatisch an den entsprechenden LAN-Client weitergeleitet. Diesen Vorgang bezeichnet man auch als statisches NAT. Dem **natd**-Gatewayrechner könnten beispielsweise die IP-Adressen `128.1.1.1`, `128.1.1.2` sowie `128.1.1.3` zugewiesen werden. `128.1.1.1` wird als die externe IP-Adresse des **natd**-Gatewayrechners verwendet, während `128.1.1.2` und `128.1.1.3` an die LAN-Clients `A` und `B` weitergegeben werden.

`-redirect_address` benutzt folgende Syntax:

```
-redirect_address localIP publicIP
```

localIP	Die interne IP-Adresse des LAN-Clients
publicIP	Die externe IP-Adresse des LAN-Clients

Für unser Beispiel hieße dies:

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

Analog zur Option `-redirect_port` können Sie diese Argumente auch in der Option `natd_flags=""` in `/etc/rc.conf` angeben. Bei der Nutzung der Adressumleitung ist die Portumleitung überflüssig, weil alle für eine bestimmte IP-Adresse ankommenden Daten umgeleitet werden.

Die externe IP-Adresse des **natd**-Rechners muss aktiv sein und der externen Netzwerkkarte zugewiesen sein. Weitere Informationen zu diesem Thema finden Sie in rc.conf(5).

32.10. PLIP – Parallel Line IP

PLIP ermöglicht TCP/IP-Verbindungen zwischen zwei Rechnern, die über ihre parallelen Schnittstellen verbunden sind. Eine solche Verbindung ist nützlich, wenn zwei Rechner nicht mit Netzwerkkarten ausgestattet sind, oder wenn eine Installation auf einem Laptop erfolgen soll. Dieser Abschnitt behandelt folgende Themen:

- Die Herstellung eines parallelen (Laplink-) Kabels
- Die Verbindung von zwei Computern über PLIP

32.10.1. Ein paralleles Kabel herstellen

Ein paralleles (Laplink-)Kabel können Sie in fast jedem Computergeschäft kaufen. Falls dies nicht möglich sein sollte, oder Sie einfach wissen wollen, wie ein solches Kabel aufgebaut ist, sollten Sie sich die folgende Tabelle ansehen. Sie beschreibt die Herstellung eines parallelen Netzwerkkabels aus einem gewöhnlichen parallelen Druckerkabel.

Tabelle 32-1. Die Netzwerk-Verdrahtung eines parallelen Kabels

A-Name	A-Ende	B-Ende	Beschreibung	Post/Bit
DATA0 -ERROR	2 15	15 2	Data	0/0x01 1/0x08
DATA1 +SLCT	3 13	13 3	Data	0/0x02 1/0x10
DATA2 +PE	4 12	12 4	Data	0/0x04 1/0x20
DATA3 -ACK	5 10	10 5	Strobe	0/0x08 1/0x40
DATA4 BUSY	6 11	11 6	Data	0/0x10 1/0x80
GND	18-25	18-25	GND	-

32.10.2. PLIP einrichten

Als Erstes benötigen Sie ein Laplink-Kabel. Danach müssen Sie sicherstellen, dass beide Computerkernel den lpt(4)-Treiber unterstützen:

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

Der Parallelport muss Interrupt-gesteuert sein, daher sollte die Datei `/boot/device.hints` zwei Zeilen ähnlich den folgenden enthalten:

```
hint.ppc.0.at="isa"
hint.ppc.0.irq="7"
```

Danach überprüfen Sie, ob die Kernelkonfigurationsdatei die Zeile `device plip` enthält, oder ob das Kernelmodul `plip.ko` geladen wurde. In beiden Fällen sollte die parallele Schnittstelle von ifconfig(8) angezeigt werden:

```
# ifconfig plip0
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

Verbinden Sie die parallelen Schnittstellen der beiden Computer über das (Laplink-)Kabel.

Konfigurieren Sie die Netzwerkparameter auf beiden Rechnern als `root`. Wenn Sie beispielsweise den Rechner `host1` mit dem Rechner `host2` verbinden wollen, gehen Sie folgendermaßen vor:

```
                host1 <-----> host2
IP Address    10.0.0.1      10.0.0.2
```

Richten Sie die parallele Schnittstelle von `host1` ein, indem Sie Folgendes eingeben:

```
# ifconfig plip0 10.0.0.1 10.0.0.2
```

Danach richten Sie die parallele Schnittstelle von `host2` ein:

```
# ifconfig plip0 10.0.0.2 10.0.0.1
```

Sie sollten nun über eine funktionierende Verbindung verfügen. Bei Problemen lesen Sie bitte die Hilfeseiten lp(4) sowie lpt(4).

Zusätzlich sollten beide Rechner in `/etc/hosts` eingetragen werden:

```
127.0.0.1               localhost.my.domain localhost
10.0.0.1                host1.my.domain host1
10.0.0.2                host2.my.domain host2
```

Um die Verbindung zu überprüfen, pingen Sie jeden Rechner vom anderen Rechner aus an. Auf `host1` gehen Sie dazu folgendermaßen vor:

```
# ifconfig plip0
plip0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables

Internet:
Destination        Gateway          Flags     Refs     Use      Netif Expire
host2              host1              UH          0       0       plip0
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms
```

32.11. IPv6 – Internet Protocol Version 6

Beigetragen von Aaron Kaplan. Überarbeitet und erweitert von Tom Rhodes. Erweitert von Brad Davis.

Bei IPv6 (auch als IPng oder *IP next generation* bekannt) handelt es sich um die neueste Version des bekannten IP-Protokolls (das auch als IPv4 bezeichnet wird). FreeBSD enthält, genauso wie die anderen frei erhältlichen BSD-Systeme, die IPv6-Referenzimplementation von KAME. FreeBSD erfüllt damit bereits alle für die Nutzung von IPv6 nötigen Voraussetzungen. Dieser Abschnitt konzentriert sich daher auf die Konfiguration und den Betrieb von IPv6.

Anfang der 90er Jahre wurde man auf den stark steigenden Verbrauch von IPv4-Adressen aufmerksam. Im Hinblick auf das Wachstums des Internets gab es zwei Hauptsorgen:

- Die drohende Knappheit von IPv4-Adressen. Dieses Problem konnte durch die Einführung von privaten Adressräumen gemäß RFC1918 (mit Adressen wie `10.0.0.0/8`, `172.16.0.0/12`, oder `192.168.0.0/16`) sowie der Entwicklung von *Network Address Translation* (NAT) weitestgehend entschärft werden.
- Die immer größer werdenden Einträge in Router-Tabellen. Dieses Problem ist auch heute noch aktuell.

IPv6 ist in der Lage, diese, aber auch viele andere Probleme zu lösen:

- IPv6 hat einen 128 Bit großen Adressraum. Es sind also theoretisch 340.282.366.920.938.463.463.374.607.431.768.211.456 Adressen verfügbar. In anderen Worten: Für jeden Quadratmeter der Erdoberfläche sind etwa 6,67 * 10^27 IPv6-Adressen verfügbar.
- Router speichern nur noch Netzwerk-Aggregationsadressen in Ihren Routingtabellen. Dadurch reduziert sich die durchschnittliche Größe einer Routingtabelle auf 8192 Einträge.

Weitere nützliche Eigenschaften von IPv6 sind:

- Die automatische Konfiguration von Adressen, die im RFC2462 (http://www.ietf.org/rfc/rfc2462.txt) beschrieben wird.
- Anycast-Adressen (“eine-von-vielen”)
- Verpflichtende Multicast-Adressen
- Die Unterstützung von IPsec (IP-Security)
- Eine vereinfachte Headerstruktur
- Mobile IP-Adressen
- Die Umwandlung von IPv4- in IPv6-Adressen

Weitere Informationsquellen:

- Beschreibung von IPv6 auf playground.sun.com (http://playground.sun.com/pub/ipng/html/ipng-main.html)
- KAME.net (http://www.kame.net)

32.11.1. Hintergrundinformationen zu IPv6-Adressen

Es gibt verschiedene Arten von IPv6-Adressen: Unicast-, Anycast- und Multicast-Adressen.

Unicast-Adressen sind die herkömlichen Adressen. Ein Paket, das an eine Unicast-Adresse gesendet wird, kommt nur an der Schnittstelle an, die dieser Adresse zugeordnet ist.

Anycast-Adressen unterscheiden sich in ihrer Syntax nicht von Unicast-Adressen, sie wählen allerdings aus mehreren Schnittstellen eine Schnittstelle aus. Ein für eine Anycast-Adresse bestimmtes Paket kommt an der nächstgelegenen (entsprechend der Router-Metrik) Schnittstelle an. Anycast-Adressen werden nur von Routern verwendet.

Multicast-Adressen bestimmen Gruppen, denen mehrere Schnittstellen angehören. Ein Paket, das an eine Multicast-Adresse geschickt wird, kommt an allen Schnittstellen an, die zur Multicast-Gruppe gehören.

> **Anmerkung:** Die von IPv4 bekannte Broadcast-Adresse (normalerweise `xxx.xxx.xxx.255`) wird bei IPv6 durch Multicast-Adressen verwirklicht.

Tabelle 32-2. Reservierte IPv6-Adressen

IPv6-Adresse	Präfixlänge	Beschreibung	Anmerkungen
`::`	128 Bit	nicht festgelegt	entspricht `0.0.0.0` bei IPv4
`::1`	128 Bit	Loopback-Adresse	entspricht `127.0.0.1` bei IPv4
`::00:xx:xx:xx:xx`	96 Bit	Eingebettete IPv4-Adresse	Die niedrigen 32 Bit entsprechen der IPv4-Adresse. Wird auch als “IPv4-kompatible IPv6-Adresse bezeichnet”.
`::ff:xx:xx:xx:xx`	96 Bit	Eine auf IPv6 abgebildete IPv4-Adresse	Die niedrigen 32 Bit entsprechen der IPv4-Adresse. Notwendig für Rechner, die IPv6 nicht unterstützen.
`fe80::` - `feb::`	10 Bit	*link-local*	Entspricht der Loopback-Adresse bei IPv4
`fec0::` - `fef::`	10 Bit	*site-local*	
`ff::`	8 Bit	Multicast	
`001` (im Dualsystem)	3 Bit	Globaler Unicast	Alle globalen Unicastadressen stammen aus diesem Pool. Die ersten 3 Bit lauten “001”.

32.11.2. IPv6-Adressen verstehen

Die kanonische Form von IPv6-Adressen lautet `x:x:x:x:x:x:x:x`, jedes “x” steht dabei für einen 16-Bit-Hexadezimalwert. Ein Beispiel für eine IPv6-Adresse wäre etwa `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`.

Eine IPv6-Adresse enthält oft Teilzeichenfolgen aus lauter Nullen. Eine solche Zeichenfolge kann zu "::" verkürzt werden. Bis zu drei führende Nullen eines Hexquads können ebenfalls weggelassen werden. `fe80::1` entspricht also der Adresse `fe80:0000:0000:0000:0000:0000:0000:0001`.

Eine weitere Möglichkeit ist die Darstellung der letzten 32 Bit in der bekannten (dezimalen) IPv4-Darstellung, bei der Punkte (".") zur Trennung verwendet werden. `2002::10.0.0.1` ist also nur eine andere Schreibweise für die (hexadezimale) kanonische Form `2002:0000:0000:0000:0000:0000:0a00:0001`, die wiederum der Adresse `2002::a00:1` entspricht.

Sie sollten nun in der Lage sein, die folgende Ausgabe zu verstehen:

```
# ifconfig
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
         inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
         inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
         ether 00:00:21:03:08:e1
         media: Ethernet autoselect (100baseTX )
         status: active
```

Bei `fe80::200:21ff:fe03:8e1%rl0` handelt es sich um eine automatisch konfigurierte *link-local*-Adresse. Sie wird im Rahmen der automatischen Konfiguration aus der MAC-Adresse erzeugt.

Weitere Informationen zum Aufbau von IPv6-Adressen finden Sie im RFC3513 (http://www.ietf.org/rfc/rfc3513.txt).

32.11.3. Eine IPv6-Verbindung herstellen

Es gibt derzeit vier Möglichkeiten, sich mit anderen IPv6-Rechnern oder Netzwerken zu verbinden:

- Fragen Sie Ihren Internetprovider, ob er IPv6 bereits unterstützt.
- SixXS (http://www.sixxs.net) bietet weltweit IPv6-Tunnelverbindungen an.
- Die Verwendung eines 6-nach-4-Tunnels (RFC3068 (http://www.ietf.org/rfc/rfc3068.txt)).
- Die Verwendung des Ports `/usr/ports/net/freenet6` bei der Einwahl ins Internet.

32.11.4. DNS in der IPv6-Welt

Ursprünglich gab es zwei verschiedene DNS-Einträge für IPv6. Da A6-Einträge von der IETF für obsolet erklärt wurden, sind AAAA-Einträge nun Standard.

Weisen Sie die erhaltene IPv6-Adresse Ihrem Rechnernamen zu, indem Sie den Eintrag

```
MYHOSTNAME            AAAA     MYIPv6ADDR
```

in Ihre primäre DNS-Zonendatei einfügen. Falls Sie nicht für Ihre DNS-Zone verantwortlich sind, bitten Sie den dafür Zuständigen, diese Änderung durchzuführen. Die aktuellen Versionen von **bind** (Version 8.3 oder 9) sowie `dns/djbdns` (bei Verwendung des IPv6-Patches) unterstützen AAAA-Einträge.

32.11.5. `/etc/rc.conf` für die Nutzung von IPv6 anpassen

32.11.5.1. Einen Client unter IPv6 einrichten

Dieser Abschnitt beschreibt die Konfiguration eines Rechners, der in Ihrem LAN als Client, aber nicht als Router verwendet wird. Um die Schnittstelle während des Systemstarts mit rtsol(8) automatisch einzurichten, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
ipv6_enable="YES"
```

Durch die folgende Zeile weisen Sie Ihrer Schnittstelle `fxp0` die statische IP-Adresse `2001:471:1f11:251:290:27ff:fee0:2093` zu:

```
ipv6_ifconfig_fxp0="2001:471:1f11:251:290:27ff:fee0:2093"
```

Um `2001:471:1f11:251::1` als Standardrouter festzulegen, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
ipv6_defaultrouter="2001:471:1f11:251::1"
```

32.11.5.2. Gateways und Router unter IPv6 einrichten

Dieser Abschnitt beschreibt, wie Sie Ihren Rechner mit Hilfe der von Ihrem Tunnel-Anbieter erhaltenen Anweisungen dauerhaft für die Nutzung von IPv6 einrichten. Um den Tunnel beim Systemstart wiederherzustellen, passen Sie `/etc/rc.conf` wie folgt an:

Listen Sie die einzurichtenden Tunnelschnittstellen (hier `gif0`) auf:

```
gif_interfaces="gif0"
```

Um den lokalen Endpunkt *`MY_IPv4_ADDR`* über diese Schnittstelle mit dem entfernten Endpunkt *`REMOTE_IPv4_ADDR`* zu verbinden, verwenden Sie folgende Zeile:

```
gifconfig_gif0="MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

Um die Ihnen zugewiesene IPv6-Adresse als Endpunkt Ihres IPv6-Tunnels zu verwenden, fügen Sie folgende Zeile ein:

```
ipv6_ifconfig_gif0="MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Nun müssen Sie nur noch die IPv6-Standardroute angeben. Diese legt das andere Ende des IPv6-Tunnels fest.

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

32.11.5.3. Einen IPv6-Tunnel einrichten

Wenn Ihr Server IPv6-Verkehr zwischen Ihrem Netzwerk und der Außenwelt routen muss, benötigen Sie zusätzlich die folgenden Zeilen in Ihrer `/etc/rc.conf`:

```
ipv6_gateway_enable="YES"
```

32.11.6. Bekanntmachung von Routen und automatische Rechnerkonfiguration

Dieser Abschnitt beschreibt die Einrichtung von rtadvd(8), das Sie bei der Bekanntmachung der IPv6-Standardroute unterstützt.

Um rtadvd(8) zu aktivieren, fügen Sie folgende Zeile in `/etc/rc.conf` ein:

```
rtadvd_enable="YES"
```

Es ist wichtig, die Schnittstelle anzugeben, über die IPv6-Routen bekanntgemacht werden sollen. Soll rtadvd(8) `fxp0` verwenden, ist folgender Eintrag nötig:

```
rtadvd_interfaces="fxp0"
```

Danach erzeugen Sie die Konfigurationsdatei `/etc/rtadvd.conf`. Dazu ein Beispiel:

```
fxp0:\
    :addrs#1:addr="2001:471:1f11:246::":prefixlen#64:tc=ether:
```

Ersetzen Sie dabei `fxp0` durch die zu verwendende Schnittstelle.

Anschließend ersetzen Sie `2001:471:1f11:246::` durch das Präfix der Ihnen zugewiesenen Verbindung.

Wenn Sie eine `/64`-Netzmaske verwenden, müssen Sie keine weiteren Anpassungen vornehmen. Anderenfalls müssen Sie `prefixlen#` auf den korrekten Wert setzen.

32.12. ATM - Asynchronous Transfer Mode

Beigetragen von Harti Brandt.

32.12.1. *Classical IP over ATM* als PVC-Verbindung einrichten

Classical IP over ATM (CLIP) ist die einfachste Möglichkeit, um IP-Verkehr über ATM (*Asynchronous Transfer Mode*-Verbindungen zu übertragen. CLIP kann sowohl mit geschalteten Verbindungen (SVCs) als auch mit permanenten Verbindungen (PVCs) verwendet werden. Dieser Abschnitt beschreibt die Einrichtung eines PVC-basierten Netzwerks.

32.12.1.1. Ein vollständig vermaschtes Netzwerk aufbauen

Bei einem vollständig vermaschten (*fully meshed*) Netzwerk ist jeder Rechner über eine dezidierte Verbindung mit jedem anderen Rechner des Netzwerks verbunden. Die Konfiguration ist - vor allem für kleinere Netzwerke - relativ einfach. Unser Beispielnetzwerk besteht aus vier Rechnern, die jeweils über eine ATM-Adapterkarte mit dem ATM-Netzwerk verbunden sind. Als ersten Konfigurationsschritt planen wir die Vergabe von IP-Adressen sowie die anzulegenden ATM-Verbindungen:

Rechner	IP-Adresse
`hostA`	`192.168.173.1`
`hostB`	`192.168.173.2`
`hostC`	`192.168.173.3`

Rechner	IP-Adresse
`hostD`	`192.168.173.4`

Um ein vollständiges Netz aufzubauen, benötigen wir für jedes Rechnerpaar eine eigene ATM-Verbindung:

Rechnerpaar	VPI.VCI-Paar
`hostA` - `hostB`	0.100
`hostA` - `hostC`	0.101
`hostA` - `hostD`	0.102
`hostB` - `hostC`	0.103
`hostB` - `hostD`	0.104
`hostC` - `hostD`	0.105

Die Werte VPI und VCI an den Verbindungsenden können natürlich unterschiedlich sein. Wir nehmen hier aber an, dass sie gleich sind. Nun müssen wir die ATM-Schnittstellen auf jedem Rechner einrichten:

```
hostA# ifconfig hatm0 192.168.173.1 up
hostB# ifconfig hatm0 192.168.173.2 up
hostC# ifconfig hatm0 192.168.173.3 up
hostD# ifconfig hatm0 192.168.173.4 up
```

Dabei setzen wir voraus, dass `hatm0` auf allen Rechnern die ATM-Schnittstelle darstellt. Danach werden, beginnend mit `hostA`, die PVCs auf den einzelnen Rechnern eingerichtet (Wir nehmen an, dass die PVCs auf den ATM-Switches bereits eingerichet sind. Lesen Sie die entsprechenden Handbücher, wenn Sie einen Switch einrichten müssen.):

```
hostA# atmconfig natm add 192.168.173.2 hatm0 0 100 llc/snap ubr
hostA# atmconfig natm add 192.168.173.3 hatm0 0 101 llc/snap ubr
hostA# atmconfig natm add 192.168.173.4 hatm0 0 102 llc/snap ubr

hostB# atmconfig natm add 192.168.173.1 hatm0 0 100 llc/snap ubr
hostB# atmconfig natm add 192.168.173.3 hatm0 0 103 llc/snap ubr
hostB# atmconfig natm add 192.168.173.4 hatm0 0 104 llc/snap ubr

hostC# atmconfig natm add 192.168.173.1 hatm0 0 101 llc/snap ubr
hostC# atmconfig natm add 192.168.173.2 hatm0 0 103 llc/snap ubr
hostC# atmconfig natm add 192.168.173.4 hatm0 0 105 llc/snap ubr

hostD# atmconfig natm add 192.168.173.1 hatm0 0 102 llc/snap ubr
hostD# atmconfig natm add 192.168.173.2 hatm0 0 104 llc/snap ubr
hostD# atmconfig natm add 192.168.173.3 hatm0 0 105 llc/snap ubr
```

Statt UBR können auch andere *traffic contracts* verwendet werden. Voraussetzung ist allerdings, dass diese von Ihrem ATM-Adapter unterstützt werden. Ist dies der Fall, folgen auf den Namen des *traffic contracts* die entsprechenden Konfigurationsparameter. Weitere Informationen zur Konfiguration von ATM-Adapterkarten erhalten Sie über den Befehl

```
# atmconfig help natm add
```

oder durch das Lesen von atmconfig(8).

Die Konfiguration von ATM-Adaptern kann auch über die Datei `/etc/rc.conf` erfolgen. Für `hostA` sähe die Konfiguration so aus:

```
network_interfaces="lo0 hatm0"
ifconfig_hatm0="inet 192.168.173.1 up"
natm_static_routes="hostB hostC hostD"
route_hostB="192.168.173.2 hatm0 0 100 llc/snap ubr"
route_hostC="192.168.173.3 hatm0 0 101 llc/snap ubr"
route_hostD="192.168.173.4 hatm0 0 102 llc/snap ubr"
```

Mit dem folgenden Befehl lässt sich der derzeitige Status aller CLIP-Routen anzeigen:

```
hostA# atmconfig natm show
```

32.13. CARP - Common Address Redundancy Protocol

Beigetragen von Tom Rhodes.

Das *Common Address Redundancy Protocol* (CARP) erlaubt es, mehreren Rechnern die gleiche IP-Adresse zuzuweisen. Durch ein solches Vorgehen läßt sich beispielsweise die Verfügbarkeit bestimmter Dienste verbessern oder die Last zwischen einzelnen Systemen besser verteilen. Den auf diese Art und Weise konfigurierten Systemen kann zusätzlich eine eigene (im Netzwerk eindeutige) IP-Adresse zugewiesen werden (wie dies auch im folgenden Beispiel erfolgt).

Um CARP zu aktivieren, müssen Sie die FreeBSD-Kernelkonfigurationsdatei um die folgende Option erweitern und danach den FreeBSD-Kernel (wie in Kapitel 9 beschrieben) neu bauen:

```
device  carp
```

Alternativ können Sie aber auch das Kernelmodul `if_carp.ko` beim Systemstart automatisch laden. Dazu nehmen Sie die folgende Zeile in die Datei `/boot/loader.conf` auf:

```
if_carp_load="YES"
```

Danach ist CARP auf Ihrem System verfügbar und kann über verschiedene `sysctl`-Optionen (OIDs) gesteuert werden.

OID	Beschreibung
`net.inet.carp.allow`	Akzeptiert ankommende CARP-Pakete. In der Voreinstellung aktiviert.
`net.inet.carp.preempt`	Diese Option deaktiviert alle CARP-Geräte, sobald eines von ihnen ausfällt. In der Voreinstellung deaktiviert.
`net.inet.carp.log`	Hat diese Variable den Wert `0`, wird kein Protokoll generiert, während mit dem Wert `1` nur inkorrekte CARP-Pakete protokolliert werden. Hat die Variable einen Wert größer `1`, werden nur die Statuswechsel von CARP-Geräten protokolliert. In der Voreinstellung hat diese Variable den Wert `1`.

OID	Beschreibung
`net.inet.carp.arpbalance`	Gleicht die Netzwerklast im lokalen Netzwerk durch den Einsatz von ARP aus. In der Voreinstellung deaktiviert.
`net.inet.carp.suppress_preempt`	Eine nur lesbare OID, die den *Preemption Suppression*-Status anzeigt. Preemption kann verhindert werden. Dies auch dann, wenn ein Gerät ausfällt. Hat die Variable den Wert `0`, bedeutet dies, dass Preemption nicht verhindert wird. Tritt ein Problem auf, wird der Wert dieser OID um `1` erhöht.

Das CARP-Gerät selbst erzeugen Sie mit dem `ifconfig`-Befehl:

```
# ifconfig carp0 create
```

Damit Sie dieses Protokoll in Ihrem Netzwerk einsetzen können, muss jede Netzwerkkarte eine eindeutige Identifikationsnummer, die sogenannte VHID (*Virtual Host Identification*), besitzen, da sich ansonsten die Rechner Ihres Netzwerks nicht voneinander unterscheiden lassen.

32.13.1. Die Serververfügbarkeit mit CARP verbessern

Wie bereits weiter oben erwähnt wurde, können Sie CARP dazu verwenden, die Verfübarkeit Ihrer Server zu verbessern. Im folgenden Bespiel werden insgesamt drei Server (mit jeweils eigener, eindeutiger IP-Adresse), die alle den gleichen Inhalt anbieten, in einer *Round Robin* DNS-Konfiguration eingerichtet. Der Backup-Server verfügt über zwei CARP-Schnittstellen (für die beiden IP-Adressen der Content-Server). Tritt bei einem Content-Server ein Problem auf, übernimmt der Backup-Server die IP-Adresse des ausgefallenen Servers. Dadurch sollte die Auswahl eines Servers vom Anwender nicht bemerkt werden. Der Backup-Server muss identisch konfiguriert sein und die gleichen Daten und Dienste anbieten wie das System, das er ersetzen soll.

Die beiden Content-Server werden (abgesehen von ihren jeweiligen Hostnamen und VHIDs) identisch konfiguriert und heißen in unserem Beispiel `hosta.example.org` beziehungsweise `hostb.example.org`. Damit Sie CARP einsetzen können, müssen Sie als Erstes die Datei `rc.conf` auf beiden Systemen anpassen. Für das System `hosta.example.org` nehmen Sie dazu folgende Zeilen in `rc.conf` auf:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpass 192.168.1.50/24"
```

Für das System `hostb.example.org` benötigen Sie zusätzlich folgende Zeilen in `rc.conf`:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```

> **Anmerkung:** Achten Sie unbedingt darauf, dass die durch die Option `pass` an `ifconfig` übergebenen Passwörter auf beiden Systemen identisch sind, da `carp`-Geräte nur mit Systemen kommunizieren können, die über ein korrektes Passwort verfügen. Beachten Sie weiters, dass sich die VHIDs der beiden Systeme unterscheiden müssen.

Nun richten Sie noch das dritte System, `provider.example.org`, ein, das aktiviert wird, wenn eines der beiden zuvor konfigurierten Systeme ausfällt. Dieses dritte System benötigt zwei `carp`-Geräte, um bei Bedarf eines der beiden anderen Systeme ersetzen zu können. Dazu konfigurieren Sie `rc.conf` analog zur folgenden Beispielkonfiguration:

```
hostname="provider.example.org"
ifconfig_fxp0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass testpass 192.168.1.51/24"
```

Durch die beiden `carp`-Geräte ist es `provider.example.org` möglich, festzustellen, ob eines der beiden anderen Systeme nicht mehr reagiert. In diesem Fall übernimmt `provider.example.org` die IP-Adresse des betroffenen Systems.

> **Anmerkung:** Ist im installierten FreeBSD-Kernel die Option "preemption" aktiviert, kann es sein, dass `provider.example.org` die übernommene IP-Adresse nicht mehr an den Content-Server zurückgibt (wenn dieser wieder funktioniert). In diesem Fall muss ein Administrator die entsprechende Schnittstelle dazu zwingen, dies zu tun. Dazu gibt er auf dem Rechner `provider.example.org` den folgenden Befehl ein:
>
> ```
> # ifconfig carp0 down && ifconfig carp0 up
> ```
>
> Dieser Befehl muss auf das `carp`-Gerät ausgeführt werden, das dem betroffenen System zugeordnet ist.

Damit ist CARP vollständig konfiguriert und der Testbetrieb kann beginnen. Zuvor müssen Sie allerdings noch alle Systeme neu starten (beziehungsweise die Netzwerkkonfiguration auf allen Systemen neu einlesen), um die Einstelllungen zu übernehmen.

Für weitere Informtionen lesen Sie bitte die Manualpage carp(4).

V. Anhang

Anhang A. Bezugsquellen für FreeBSD

A.1. CD-ROM und DVD Verleger

A.1.1. FreeBSD-Pakete

FreeBSD-Pakete (FreeBSD-CDs, zusätzliche Software und gedruckte Dokumentation) erhalten Sie von mehreren Händlern:

- CompUSA
  WWW: http://www.compusa.com/
- Frys Electronics
  WWW: http://www.frys.com/

A.1.2. FreeBSD-CDs und -DVDs

Die FreeBSD-CDs und -DVDs werden von vielen Online-Händlern angeboten:

- FreeBSD Mall, Inc.
  700 Harvest Park Ste F
  Brentwood, CA 94513
  USA
  Telefon: +1 925 240-6652
  Fax: +1 925 674-0821
  E-Mail: <info@freebsdmall.com>
  WWW: http://www.freebsdmall.com/
- Dr. Hinner EDV
  St. Augustinus-Str. 10
  D-81825 München
  Germany
  Telefon: (089) 428 419
  WWW: http://www.hinner.de/linux/freebsd.html
- Ikarios
  22-24 rue Voltaire
  92000 Nanterre
  France
  WWW: http://ikarios.com/form/#freebsd
- JMC Software
  Ireland
  Telefon: 353 1 6291282
  WWW: http://www.thelinuxmall.com
- The Linux Emporium
  Hilliard House, Lester Way

Wallingford
OX10 9TA
United Kingdom
Telefon: +44 1491 837010
Fax: +44 1491 837016
WWW: http://www.linuxemporium.co.uk/products/bsd/

- Linux+ DVD Magazine
  Lewartowskiego 6
  Warsaw
  00-190
  Poland
  Telefon: +48 22 860 18 18
  E-Mail: <editors@lpmagazine.org>
  WWW: http://www.lpmagazine.org/
- Linux System Labs Australia
  21 Ray Drive
  Balwyn North
  VIC - 3104
  Australia
  Telefon: +61 3 9857 5918
  Fax: +61 3 9857 8974
  WWW: http://www.lsl.com.au/
- LinuxCenter.Ru
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Russia
  Telefon: +7-812-3125208
  E-Mail: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/shop/freebsd

A.1.3. Lieferanten

Wenn Sie FreeBSD-CD-ROM-Produkte weiterverkaufen möchten, kontaktieren Sie einen der folgenden Lieferanten:

- Cylogistics
  809B Cuesta Dr., #2149
  Mountain View, CA 94040
  USA
  Telefon: +1 650 694-4949
  Fax: +1 650 694-4953
  E-Mail: <sales@cylogistics.com>
  WWW: http://www.cylogistics.com/
- Ingram Micro
  1600 E. St. Andrew Place

Santa Ana, CA 92705-4926
USA
Telefon: 1 (800) 456-8000
WWW: http://www.ingrammicro.com/

- Kudzu, LLC
  7375 Washington Ave. S.
  Edina, MN 55439
  USA
  Telefon: +1 952 947-0822
  Fax: +1 952 947-0876
  E-Mail: <sales@kudzuenterprises.com>

- LinuxCenter.Kz
  Ust-Kamenogorsk
  Kazakhstan
  Telefon: +7-705-501-6001
  E-Mail: <info@linuxcenter.kz>
  WWW:
  http://linuxcenter.kz/page.php?page=fr

- LinuxCenter.Ru
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Russia
  Telefon: +7-812-3125208
  E-Mail: <info@linuxcenter.ru>
  WWW:
  http://linuxcenter.ru/freebsd

- Navarre Corp
  7400 49th Ave South
  New Hope, MN 55428
  USA
  Telefon: +1 763 535-8333
  Fax: +1 763 535-0341
  WWW: http://www.navarre.com/

A.2. FTP-Server

Die offiziellen Quellen von FreeBSD sind mit anonymous FTP über ein weltweites Netz von FTP-Spiegeln erhältlich. Obwohl ftp://ftp.FreeBSD.org/pub/FreeBSD/ über eine gute Anbindung verfügt, sollten Sie einen Spiegel in Ihrer Nähe verwenden (insbesondere, wenn Sie selber einen Spiegel einrichten wollen).

Sie können FreeBSD auch über anonymous FTP von den folgenden Spiegeln beziehen. Wenn Sie FreeBSD über anonymous FTP beziehen wollen, wählen Sie bitte einen Spiegel in Ihrer Nähe. Die unter “Haupt-Spiegel” aufgeführten Spiegel stellen normalerweise das komplette FreeBSD-Archiv (alle momentan erhältlichen Versionen für jede unterstützte Architektur) zur Verfügung. Wahrscheinlich geht es aber schneller, wenn Sie einen Spiegel in

Ihrer Nähe benutzen. Die Länder-Spiegel stellen die neusten Versionen für die beliebtesten Architekturen bereit, sie stellen aber unter Umständen nicht das komplette FreeBSD-Archiv bereit. Auf alle Server kann mit anonymous FTP zugegriffen werden, einige Server bieten auch andere Zugriffsmethoden an. Die zur Verfügung stehenden Zugriffsmethoden sind bei jedem Server in Klammern angegeben.

Hauptserver, Hauptspiegel, Armenien, Australien, Brasilien, China, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Großbritannien, Hong Kong, Irland, Island, Japan, Kanada, Korea, Lettland, Litauen, Neuseeland, Niederlande, Norwegen, Österreich, Polen, Russland, Saudi Arabien, Schweden, Schweiz, Slowakische Republik, Slowenien, Spanien, Südafrika, Taiwan, Tschechische Republik, Türkei, Ukraine, USA.

(aktualisiert am: UTC)

Hauptserver

- ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.FreeBSD.org/pub/FreeBSD/))

Hauptspiegel

Bei Problemen wenden Sie sich bitte an den Betreuer <mirror-admin@FreeBSD.org> dieser Domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp4.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp4.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp10.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp10.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp14.FreeBSD.org/pub/FreeBSD/))

Armenien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@am.FreeBSD.org> dieser Domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp1.am.FreeBSD.org/pub/FreeBSD/) / rsync)

Australien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@au.FreeBSD.org> dieser Domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

Brasilien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@br.FreeBSD.org> dieser Domain.

- ftp://ftp.br.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.br.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http (http://ftp2.br.FreeBSD.org/))
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp5.br.FreeBSD.org

China

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@cn.FreeBSD.org> dieser Domain.

- ftp://ftp.cn.FreeBSD.org/pub/FreeBSD/ (ftp)

Dänemark

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@dk.FreeBSD.org> dieser Domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp.dk.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.dk.FreeBSD.org/pub/FreeBSD/))

Deutschland

Bei Problemen wenden Sie sich bitte an den Betreuer <de-bsd-hubs@de.FreeBSD.org> dieser Domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http (http://www1.de.FreeBSD.org/freebsd/) / rsync (rsync://rsync3.de.FreeBSD.org/freebsd/))
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.de.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http (http://ftp4.de.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp7.de.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp8.de.FreeBSD.org/pub/FreeBSD/ (ftp)

Estland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ee.FreeBSD.org> dieser Domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

Finnland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@fi.FreeBSD.org> dieser Domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

Frankreich

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@fr.FreeBSD.org> dieser Domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp1.fr.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.fr.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp4.fr.FreeBSD.org/pub/FreeBSD/) / http (http://ftp4.fr.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp4.fr.FreeBSD.org/pub/FreeBSD/) / rsync (rsync://ftp4.fr.FreeBSD.org/FreeBSD/) / rsyncv6 (rsync://ftp4.fr.FreeBSD.org/FreeBSD/))
- ftp://ftp5.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

Griechenland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@gr.FreeBSD.org> dieser Domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

Großbritannien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@uk.FreeBSD.org> dieser Domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.uk.FreeBSD.org/) / rsync)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

Hong Kong

- ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

Irland

Bei Problemen wenden Sie sich bitte an den Betreuer <`hostmaster@ie.FreeBSD.org`> dieser Domain.

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp3.ie.FreeBSD.org/pub/FreeBSD/) / rsync)

Island

Bei Problemen wenden Sie sich bitte an den Betreuer <`hostmaster@is.FreeBSD.org`> dieser Domain.

- ftp://ftp.is.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Japan

Bei Problemen wenden Sie sich bitte an den Betreuer <`hostmaster@jp.FreeBSD.org`> dieser Domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

Kanada

Bei Problemen wenden Sie sich bitte an den Betreuer <`hostmaster@ca.FreeBSD.org`> dieser Domain.

- ftp://ftp.ca.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.ca.FreeBSD.org/pub/FreeBSD/ (ftp)

Korea

Bei Problemen wenden Sie sich bitte an den Betreuer <`hostmaster@kr.FreeBSD.org`> dieser Domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.kr.FreeBSD.org/pub/FreeBSD/))

Lettland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@lv.FreeBSD.org> dieser Domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.lv.FreeBSD.org/pub/FreeBSD/))

Litauen

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@lt.FreeBSD.org> dieser Domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.lt.FreeBSD.org/pub/FreeBSD/))

Neuseeland

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.nz.FreeBSD.org/pub/FreeBSD/))

Niederlande

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@nl.FreeBSD.org> dieser Domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.nl.FreeBSD.org/os/FreeBSD/) / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

Norwegen

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@no.FreeBSD.org> dieser Domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Österreich

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@at.FreeBSD.org> dieser Domain.

- ftp://ftp.at.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp.at.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.at.FreeBSD.org/pub/FreeBSD/))

Polen

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@pl.FreeBSD.org> dieser Domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.pl.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp2.pl.FreeBSD.org/pub/FreeBSD/) / http (http://ftp2.pl.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp2.pl.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)

Russland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ru.FreeBSD.org> dieser Domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.ru.FreeBSD.org/FreeBSD/) / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.ru.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp4.ru.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp5.ru.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

Saudi Arabien

Bei Problemen wenden Sie sich bitte an den Betreuer <ftpadmin@isu.net.sa> dieser Domain.

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org/ (ftp)

Schweden

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@se.FreeBSD.org> dieser Domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync (rsync://ftp2.se.FreeBSD.org/))
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/) / http (http://ftp4.se.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp4.se.FreeBSD.org/pub/FreeBSD/) / rsync (rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/) / rsyncv6 (rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp5.se.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp5.se.FreeBSD.org/) / rsync)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD (ftp / http (http://ftp6.se.FreeBSD.org/pub/FreeBSD))

Schweiz

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ch.FreeBSD.org> dieser Domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.ch.FreeBSD.org/pub/FreeBSD/))

Slowakische Republik

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@sk.FreeBSD.org> dieser Domain.

- ftp://ftp.sk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp.sk.FreeBSD.org/pub/FreeBSD/) / http (http://ftp.sk.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.sk.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp2.sk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp2.sk.FreeBSD.org/pub/FreeBSD/) / http (http://ftp2.sk.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp2.sk.FreeBSD.org/pub/FreeBSD/))

Slowenien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@si.FreeBSD.org> dieser Domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

Spanien

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@es.FreeBSD.org> dieser Domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.es.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

Südafrika

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@za.FreeBSD.org> dieser Domain.

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

Taiwan

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@tw.FreeBSD.org> dieser Domain.

- ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / http (http://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp2.tw.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp3.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp6.tw.FreeBSD.org/) / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp11.tw.FreeBSD.org/FreeBSD/))
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

Tschechische Republik

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@cz.FreeBSD.org> dieser Domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 (ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/) / http (http://ftp.cz.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp.cz.FreeBSD.org/pub/FreeBSD/) / rsync / rsyncv6)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp2.cz.FreeBSD.org/pub/FreeBSD/))

Türkei

- ftp://ftp.tr.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.tr.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp2.tr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

Ukraine

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp.ua.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

USA

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@us.FreeBSD.org> dieser Domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http (http://ftp4.us.FreeBSD.org/pub/FreeBSD/) / httpv6 (http://ftp4.us.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp13.us.FreeBSD.org/pub/FreeBSD/) / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http (http://ftp14.us.FreeBSD.org/pub/FreeBSD/))
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

A.3. BitTorrent

Die ISO-Images für die Release-CDs sind via BitTorrent abrufbar. Eine Sammlung von Torrent-Dateien zum Herunterladen der Images ist unter http://torrents.freebsd.org:8080 (http://torrents.freebsd.org:8080/) verfügbar.

Die BitTorrent Client-Software ist als Port `net-p2p/py-bittorrent` oder als vorkompiliertes Paket erhältlich.

Nach dem Herunterladen der ISO-Images mit BitTorrent können Sie diese auf CD oder DVD brennen, so wie im burncd-Abschnitt 19.6.3 beschrieben.

A.4. Anonymous CVS

A.4.1. Einführung

Anonymous CVS (oder *anoncvs*) dient zum Synchronisieren mit entfernten Repositories und steht mit den **CVS** Werkzeugen, die im FreeBSD Basissystem enthalten sind, zur Verfügung. Benutzer von FreeBSD können damit unter anderem lesende Operationen auf den **Anoncvs** Servern des FreeBSD Projects durchführen, ohne über besondere Berechtigungen zu verfügen. Um es zu benutzen, setzen Sie einfach die `CVSROOT` Umgebungsvariable auf einen **Anoncvs** Server und geben beim Login mit `cvs login` das Passwort `anoncvs` an. Danach können Sie mit cvs(1) wie auf jedes lokale Repository (allerdings nur lesend) zugreifen.

> **Anmerkung:** `cvs login` speichert Passwörter zur Authentifizierung an einem CVS Server in der Datei `.cvspass` in Ihrem `HOME`-Verzeichnis. Wenn diese Datei beim ersten Benutzen von `cvs login` nicht existiert, erhalten Sie vielleicht eine Fehlermeldung. In diesem Fall legen Sie einfach eine leere `.cvspass` Datei an und melden sich erneut an.

CVSup und **Anoncvs** bieten dieselbe Funktionalität, die folgenden Kriterien helfen Ihnen zu entscheiden, welche Methode Sie benutzen sollen. **CVSup** geht wesentlich effizienter mit Netzwerk-Ressourcen um und ist auch technisch ausgereifter. Allerdings müssen Sie zuerst einen speziellen Client installieren und konfigurieren, bevor Sie **CVSup** benutzen können. Weiterhin können Sie mit **CVSup** nur relativ große Teile der Quellen, die *Sammlungen* genannt werden, synchronisieren.

Im Gegensatz dazu können Sie mit **Anoncvs** jede beliebige Datei oder indem Sie einfach den **CVS** Namen des Moduls angeben, ein beliebiges Programm, wie `ls` oder `grep`, bearbeiten. Natürlich können Sie mit **Anoncvs** nur lesend auf ein **CVS** Repository zugreifen. Wenn Sie lokal mit dem FreeBSD-Repository entwickeln wollen, dann ist **CVSup** die einzige Wahl.

A.4.2. Benutzen von Anonymous CVS

Setzen Sie einfach die `CVSROOT` Umgebungsvariable, um cvs(1) das **CVS** Repository eines FreeBSD **Anoncvs**-Servers bekannt zu geben. Zurzeit stehen folgende Server zur Verfügung:

- *Frankreich*: :pserver:anoncvs@anoncvs.fr.FreeBSD.org:/home/ncvs (Das Passwort für pserver ist `anoncvs`, SSH-Zugriffe verwenden kein Passwort.)
- *Taiwan*: :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs (pserver: Benutzen Sie `cvs login` und ein beliebiges Passwort. SSH-Zugriffe erfordern kein Passwort.)

```
SSH2 HostKey: 1024 02:ed:1b:17:d6:97:2b:58:5e:5c:e2:da:3b:89:88:26 /etc/ssh/ssh_host_rsa_key.pub
SSH2 HostKey: 1024 e8:3b:29:7b:ca:9f:ac:e9:45:cb:c8:17:ae:9b:eb:55 /etc/ssh/ssh_host_dsa_key.pub
```

- *USA*: anoncvs@anoncvs1.FreeBSD.org:/home/ncvs (nur SSH2 ohne Passwort).

  ```
  SSH2 HostKey: 2048 53:1f:15:a3:72:5c:43:f6:44:0e:6a:e9:bb:f8:01:62 /etc/ssh/ssh_host_dsa_key.pub
  ```

Mit **CVS** können Sie praktisch jede Version von FreeBSD, die schon einmal existiert hat (oder in manchen Fällen existieren wird) auschecken. Sie sollten daher damit vertraut sein, wie Sie mit Tags unter cvs(1) arbeiten (die `-r` Option). Zudem müssen Sie die Namen der Tags im FreeBSD-Repository kennen.

Es gibt zwei verschiedene Tags[1]: Tags, die Revisionen bezeichnen und Tags, die Zweige bezeichnen. Die Ersten sind statisch und fest an eine Revision gebunden. Ein Tag, das einen Zweig bezeichnet, bezieht sich dagegen zu einem gegebenen Zeitpunkt immer auf die aktuellste Revision. Da ein Tag eines Zweiges nicht an eine bestimmte Revision gebunden ist, kann sich dessen Bedeutung von heute auf morgen ändern.

In Abschnitt A.7 finden Sie eine Liste der gültigen Tags. Beachten Sie bitte, dass keines der Tags auf die Ports-Sammlung anwendbar ist, da diese nicht über Zweige verfügt.

Wenn Sie ein Tag eines Zweiges verwenden, erhalten Sie die aktuellsten Dateien dieses Entwicklungszweiges. Wenn Sie eine frühere Revision erhalten möchten, können Sie zum Beispiel einen Zeitpunkt mit der `-D` Option angeben. Weitere Informationen dazu entnehmen Sie bitte cvs(1).

A.4.3. Beispiele

Im Folgenden finden Sie einige Beispiele für den Umgang mit **Anonymous CVS**. Sie sollten sich aber die Manualpage von cvs(1) sorgfältig durchlesen, bevor Sie anfangen.

Beispiel A-1. ls(1) von -CURRENT auschecken

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Wenn Sie dazu aufgefordert werden, benutzen Sie ein beliebiges “Passwort”.
% cvs co ls
```

Beispiel A-2. Den `src/`-Baum über SSH auschecken

```
% cvs -d anoncvs@anoncvs1.FreeBSD.org:/home/ncvs co src
The authenticity of host ’anoncvs1.freebsd.org (216.87.78.137)’ can’t be established.
DSA key fingerprint is 53:1f:15:a3:72:5c:43:f6:44:0e:6a:e9:bb:f8:01:62.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added ’anoncvs1.freebsd.org’ (DSA) to the list of known hosts.
```

Beispiel A-3. ls(1) aus dem 8-STABLE-Zweig auschecken

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Wenn Sie dazu aufgefordert werden, benutzen Sie ein beliebiges “Passwort”.
% cvs co -rRELENG_8 ls
```

Beispiel A-4. Änderungen in ls(1) zwischen 5.3 RELEASE und 5.4 RELEASE (als unified diff)

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Wenn Sie dazu aufgefordert werden, benutzen Sie ein beliebiges "Passwort".
% cvs rdiff -u -rRELENG_8_0_0_RELEASE -rRELENG_8_1_0_RELEASE ls
```

Beispiel A-5. Gültige Modulnamen herausfinden

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.tw.FreeBSD.org:/home/ncvs
% cvs login
Wenn Sie dazu aufgefordert werden, benutzen Sie ein beliebiges "Passwort".
% cvs co modules
% more modules/modules
```

A.4.4. Weitere Ressourcen

Die folgenden Ressourcen sind nützlich, um den Umgang mit **CVS** zu lernen:

- CVS Tutorial (http://users.csc.calpoly.edu/~gfisher/classes/205/handouts/cvs-basics.html) von der California Polytechnic State University.
- CVS Home (http://ximbiot.com/cvs/wiki/), die Homepage des **CVS**-Projekts.
- CVSweb (http://www.FreeBSD.org/cgi/cvsweb.cgi) das Web Interface zu **CVS** des FreeBSD Projekts.

A.5. CTM

Mit **CTM**[2] können Sie einen entfernten Verzeichnisbaum mit einem zentralen Baum synchronisieren. Es wurde extra zum Synchronisieren der FreeBSD Quellen entwickelt, obwohl es mit der Zeit vielleicht auch andere Anwendungen geben wird. Zurzeit existiert leider so gut wie keine Dokumentation zum Erstellen der Deltas. Wenn Sie Hilfe benötigen oder **CTM** für andere Zwecke einsetzen wollen, wenden Sie sich bitte an die Mailingliste ctm-users (http://lists.FreeBSD.org/mailman/listinfo/ctm-users).

A.5.1. Warum soll ich CTM benutzen?

Mit **CTM** erhalten Sie eine lokale Kopie des FreeBSD-Quellbaums, den es in mehreren "Varianten" gibt. Sie können das ganze Repository oder nur einen Zweig spiegeln. Wenn Sie ein aktiver FreeBSD-Entwickler mit einer schlechten oder gar keiner TCP/IP Verbindung sind, oder die Änderungen einfach automatisch zugesandt bekommen wollen, dann ist **CTM** das Richtige für Sie. Für die Zweige mit der meisten Aktivität müssen Sie sich täglich bis zu drei Deltas beschaffen, Sie sollten allerdings erwägen, die Deltas automatisch über E-Mail zu beziehen. Die Größe der Updates wird so klein wie möglich gehalten. Normalerweise sind sie kleiner als 5 kB, manchmal sind sie 10-50 kB groß (etwa jedes 10. Update) und ab und an werden Sie auch einmal ein Update mit 100 kB oder mehr erhalten.

Sie sollten sich über die Vorbehalte gegen die Verwendung der Quellen anstelle eines offiziellen Releases bewusst sein. Das trifft besonders auf FreeBSD-CURRENT zu, lesen Sie dazu bitte den Abschnitt FreeBSD-CURRENT.

A.5.2. Was brauche ich, um CTM zu benutzen?

Zwei Sachen: Das **CTM** Programm und die initialen Deltas, von denen aus Sie auf die “aktuellen” Stände kommen.

CTM ist schon seit der Version 2.0 Teil des FreeBSD-Basissystems. Sie finden es in `/usr/src/usr.sbin/ctm`, wenn Sie eine Kopie der Quellen besitzen.

Die Deltas, die **CTM** verarbeitet, können Sie über FTP oder E-Mail beziehen. Wenn Sie über einen FTP Zugang zum Internet verfügen, erhalten Sie die Deltas unter der folgenden URL:

ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/

Die Deltas werden auch von CTM Spiegeln bereitgehalten.

Wechseln Sie in das passende Verzeichnisse zum Beispiel `src-cur` für FreeBSD-CURRENT und laden Sie sich von dort die Deltas herunter.

Sie können die Deltas auch über E-Mail beziehen.

Abonnieren Sie dazu eine der **CTM**-Verteilerlisten. Über ctm-cvs-cur (http://lists.FreeBSD.org/mailman/listinfo/ctm-cvs-cur) erhalten Sie den kompletten **CVS**-Baum, über ctm-src-cur (http://lists.FreeBSD.org/mailman/listinfo/ctm-src-cur) erhalten Sie FreeBSD-CURRENT und über ctm-src-7 (http://lists.FreeBSD.org/mailman/listinfo/ctm-src-7) erhalten Sie den FreeBSD 7.X-Zweig. Wenn Sie nicht wissen, wie Sie eine der Mailinglisten abonnieren, folgen Sie einem der Verweise von oben oder besuchen Sie die Seite http://lists.FreeBSD.org/mailman/listinfo. Weitere Informationen erhalten Sie, wenn Sie dort auf die gewünschte Liste klicken.

Benutzen Sie `ctm_rmail`, um die **CTM** Updates, die Sie per E-Mail empfangen, auszupacken und anzuwenden. Wenn Sie diesen Prozess automatisiert ablaufen lassen möchten, können Sie dazu einen Eintrag in `/etc/aliases` verwenden. Genauere Informationen finden Sie in der Manualpage von `ctm_rmail`.

> **Anmerkung:** Sie sollten die Mailingliste ctm-announce (http://lists.FreeBSD.org/mailman/listinfo/ctm-announce) abonnieren, egal wie Sie die **CTM**-Deltas erhalten. Ankündigungen, die den Betrieb des **CTM**-Systems betreffen, werden nur auf dieser Liste bekannt gegeben. Klicken Sie auf den Namen der Liste oder besuchen Sie die Seite http://lists.FreeBSD.org/mailman/listinfo, um diese Liste zu abonnieren.

A.5.3. Initialisieren von CTM

Bevor Sie die **CTM** Deltas benutzen können, brauchen Sie einen Startpunkt, auf den die nachfolgenden Deltas angewendet werden.

Sie können natürlich mit einem leeren Verzeichnis beginnen. In diesem Fall benötigen Sie ein `XEmpty`-Delta, mit dem Sie den **CTM**-Verzeichnisbaum initialisieren. Wenn Sie Glück haben, finden Sie ein `XEmpty`-Delta, mit dem sie beginnen können, auf einer der CDs Ihrer Distribution.

Da die Verzeichnisbäume mehrere Megabyte groß sind, sollten Sie nach Möglichkeit etwas schon vorhandenes benutzen. Wenn Sie eine -RELEASE CD besitzen, können Sie die Quellen von dieser CD benutzen. Sie ersparen sich damit das Übertragen großer Datenmengen.

Die Deltas, mit denen Sie beginnen können, enthalten ein `X` in ihrem Namen, wie in `src-cur.3210XEmpty.gz`. Hinter dem `X` wird der Startpunkt der Deltas angegeben, in diesem Fall steht `Empty` für ein leeres Verzeichnis. Nach etwa 100 Deltas wird ein neues `XEmpty`-Delta erstellt. Mit ungefähr 75 Megabyte komprimierter Daten sind diese `XEmpty`-Deltas übrigens sehr groß.

Nachdem Sie Ihren Startpunkt festgelegt haben, benötigen Sie alle Deltas mit einer höheren Nummer.

A.5.4. Benutzen von CTM

Um ein Delta einzuspielen, benutzen Sie das folgende Kommando:

```
# cd /Pfad/zu/den/Quellen
# ctm -v -v /Pfad/zu/den/Deltas/src-xxx.*
```

CTM kann mit Deltas arbeiten, die mit `gzip` komprimiert wurden. Sie brauchen die Deltas vorher nicht mit `gunzip` zu dekomprimieren und sparen damit Plattenplatz.

Ihr Quellbaum wird erst dann verändert, wenn **CTM** die Deltas sauber verarbeiten kann. Die Integrität der Deltas und ihre Anwendbarkeit auf den Quellbaum lassen sich durch die Angabe des Schalters `-c` überprüfen, **CTM** ändert in diesem Fall Ihren Quellbaum nicht.

CTM verfügt über weitere Kommandozeilenoptionen, Informationen dazu finden Sie in der Manualpage oder dem Quellcode.

Das war schon alles. Um Ihre Quellen aktuell zu halten, verwenden Sie **CTM** jedes Mal, wenn Sie neue Deltas bekommen.

Löschen Sie die Deltas nicht, wenn Sie diese nur schwer wieder beschaffen können. Behalten Sie sie für den Fall, das etwas passiert. Auch wenn Sie nur Disketten besitzen, sollten Sie erwägen, die Deltas mit `fdwrite` zu sichern.

A.5.5. Umgang mit lokalen Änderungen

Entwickler wollen mit den Dateien im Quellbaum experimentieren und diese verändern. In beschränkter Weise werden lokale Änderungen von **CTM** unterstützt. Wenn **CTM** die Datei `foo` bearbeiten will, überprüft es zuerst ob die Datei `foo.ctm` existiert. Wenn diese Datei existiert, werden Änderungen in ihr anstatt in `foo` vorgenommen.

Mit diesem Verfahren ist eine leichte Handhabung lokaler Änderungen möglich. Kopieren Sie die Dateien, die Sie ändern möchten, in Dateien, die das Suffix `.ctm` tragen. Sie können dann ungestört mit dem Quellcode arbeiten, während `CTM` die `.ctm` Dateien aktualisiert.

A.5.6. Weitere CTM-Optionen

A.5.6.1. Was wird aktualisiert?

Eine Liste der Änderungen, die **CTM** an Ihrem Quellbaum vornehmen wird, erhalten Sie, wenn Sie die Option `-l` angeben.

Das ist nützlich, wenn Sie Logs über die Änderungen führen wollen, geänderte Dateien vor- oder nachbearbeiten wollen, oder einfach ein bisschen paranoid sind.

A.5.6.2. Sicherungen vor einer Aktualisierung erstellen

Sie wollen vielleicht die Dateien, die durch eine **CTM** Aktualisierung verändert werden, sichern.

Mit `-B` *`backup-file`* weisen Sie **CTM** an, alle Dateien, die durch ein **CTM** Delta verändert würden, nach `backup-file` zu sichern.

A.5.6.3. Dateien ausschließen

Manchmal wollen Sie nur bestimmte Teile aktualisieren oder nur bestimmte Dateien aus einer Folge von Deltas extrahieren.

Sie können die Liste der Dateien, mit denen **CTM** arbeitet, einschränken, indem Sie reguläre Ausdrücke mit den Optionen `-e` und `-x` angeben.

Wenn Sie eine aktuelle Kopie von `lib/libc/Makefile` aus den gesicherten **CTM** Deltas erhalten wollen, setzen Sie das folgende Kommando ab:

```
# cd /wo/Sie/es/auspacken/wollen/
# ctm -e '^lib/libc/Makefile' ~ctm/src-xxx.*
```

Die Optionen `-e` und `-x` werden in der Reihenfolge angewandt, in der sie auf der Kommandozeile angegeben wurden. Eine Datei wird nur dann von **CTM** verarbeitet, wenn dies nach der Anwendung der Optionen `-e` und `-x` noch erlaubt ist.

A.5.7. Pläne für CTM

Mehrere:

- Hinzufügen eines Authentifizierungsmechanismus, damit gefälschte **CTM**-Deltas erkannt werden können.
- Aufräumen der **CTM**-Optionen, die mit der Zeit unübersichtlich und irreführend wurden.

A.5.8. Verschiedenes

Es gibt Deltas für die Ports-Sammlung, die aber nicht intensiv genutzt werden.

A.5.9. CTM-Spiegel

Die CTM-Deltas können Sie mit anonymous FTP von den folgenden Spiegeln beziehen. Versuchen Sie bitte einen Spiegel in Ihrer Nähe zu benutzen.

Bei Problemen wenden Sie sich bitte an die Mailingliste ctm-users (http://lists.FreeBSD.org/mailman/listinfo/ctm-users).

Kalifornien, Bay Area, Offizieller Server

- ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/

Südafrika, Backup-Server für alte Deltas

- ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/

Taiwan/R.O.C.

- ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/
- ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/

Wenn die Liste keinen Spiegel in Ihrer Nähe enthält oder Sie Probleme mit dem ausgewählten Spiegel haben, versuchen Sie einen Spiegel mit einer Suchmaschine, wie alltheweb (http://www.alltheweb.com/), zu finden.

A.6. Benutzen von CVSup

A.6.1. Einführung

CVSup ist eine Anwendung, die Verzeichnisbäume von einem entfernten **CVS**-Server bereitstellt und aktualisiert. Die Quellen von FreeBSD werden in einem **CVS**-Repository auf einer Entwicklungsmaschine in Kalifornien gepflegt. Mit **CVSup** können sich FreeBSD-Benutzer den eigenen Quellbaum auf aktuellem Stand halten.

Zum Aktualisieren benutzt **CVSup** die Pull-Methode, bei der die Aktualisierungen vom Client angefragt werden. Der Server wartet dabei passiv auf Anfragen von Clients, das heißt er verschickt nicht unaufgefordert Aktualisierungen. Somit gehen alle Anfragen vom Client aus und die Benutzer müssen **CVSup** entweder manuell starten oder einen `cron` Job einrichten, um regelmäßig Aktualisierungen zu erhalten.

CVSup in genau dieser Schreibweise bezeichnet die Anwendung, die aus dem Client `cvsup` und dem Server `cvsupd` besteht. `cvsup` läuft auf den Maschinen der Benutzer, `cvsupd` läuft auf jedem der FreeBSD-Spiegel.

Wenn Sie die FreeBSD-Dokumentation und die Mailinglisten lesen, werden Sie oft auf **Sup**, dem Vorgänger von **CVSup** stoßen. **CVSup** wird in gleicher Weise wie **Sup** benutzt und verfügt sogar über Konfigurationsdateien, die kompatibel zu denen von **Sup** sind. Da **CVSup** schneller und flexibler als **Sup** ist, wird **Sup** vom FreeBSD Project nicht mehr benutzt.

> **Anmerkung:** Mit **csup** gibt es in inzwischen auch eine in C geschriebene Neuimplementierung von **CVSup**. Der größte Vorteil dieser neuen Version ist neben einer höheren Geschwindigkeit der, dass dieses Programm nicht von der Sprache Modula-3 abhängig ist und Sie daher dieses Paket nicht mitinstallieren müssen. **csup** ist bereits im Basissystem enthalten und kann sofort verwendet werden. Wollen Sie künftig **csup** einsetzen, überspringen Sie in den folgenden Ausführungen einfach den Abschnitt zur Installation von **CVSup** und ersetzen alle Vorkommen von **CVSup** durch **csup**.

A.6.2. Installation von CVSup

CVSup können Sie leicht installieren, wenn Sie das vorkompilierte Paket `net/cvsup` aus der Ports-Sammlung benutzen. Alternativ können Sie `net/cvsup` auch ausgehend von den Quellen bauen, doch seien Sie gewarnt: `net/cvsup` hängt vom **Modula-3** System ab, das viel Zeit und Platz zum Herunterladen und Bauen braucht.

> **Anmerkung:** Wenn Sie **CVSup** auf einer Maschine ohne **Xorg** (also beispielsweise auf einem Server), benutzen, stellen Sie bitte sicher, dass Sie den Port ohne das **CVSup**-GUI, (`net/cvsup-without-gui`) verwenden.

A.6.3. Konfiguration von CVSup

Das Verhalten von **CVSup** wird mit einer Konfigurationsdatei gesteuert, die `supfile` genannt wird. Beispiele für Konfigurationsdateien finden Sie in dem Verzeichnis /usr/share/examples/cvsup/.

Ein `supfile` enthält die folgenden Informationen:

- Welche Dateien Sie erhalten wollen.
- Welche Versionen der Dateien Sie benötigen.
- Woher Sie die Dateien beziehen wollen.
- Wo Sie die erhaltenen Dateien speichern.
- Wo Sie die Status-Dateien aufbewahren wollen.

In den folgenden Abschnitten erstellen wir ein typisches `supfile` indem wir nach und nach diese Punkte klären. Zuerst beschreiben wir aber den Aufbau dieser Konfigurationsdatei.

Ein `supfile` ist eine Textdatei. Kommentare beginnen mit einem # und gelten bis zum Zeilenende. Leerzeilen und Zeilen, die nur Kommentare enthalten, werden ignoriert.

Die anderen Zeilen legen die Dateien fest, die ein Benutzer erhalten will. Der Server organisiert verschiedene Dateien in einer “Sammlung”, deren Name auf einer Zeile angegeben wird. Nach dem Namen der Sammlung können mehrere durch Leerzeichen getrennte Felder folgen, die die oben angesprochenen Informationen festlegen. Es gibt zwei Arten von Feldern: Felder, die Optionen festlegen und Felder mit Parametern. Optionen bestehen aus einem Schlüsselwort, wie `delete` oder `compress` und stehen alleine. Ein Parameterfeld beginnt mit einem Schlüsselwort, dem = und ein Parameter, wie in `release=cvs`, folgt. Dieses Feld darf keine Leerzeichen enthalten.

In einem `supfile` werden normalerweise mehrere Sammlungen angefordert. Die erforderlichen Felder können explizit für jede Sammlung angegeben werden, dann werden jedoch die Zeilen ziemlich lang. Außerdem ist dieses Vorgehen sehr unhandlich, da die meisten Felder für alle Sammlungen gleich sind. **CVSup** bietet die Möglichkeit, Vorgaben für die Felder der Sammlungen festzulegen. Zeilen, die mit der Pseudo-Sammlung `*default` beginnen, legen Optionen und Parameter für nachfolgende Sammlungen im `supfile` fest. Der Vorgabewert kann in der Zeile einer bestimmten Sammlung überschrieben werden. Durch Hinzufügen weiterer `*default` Zeilen können die Vorgaben auch mitten im `supfile` überschrieben oder erweitert werden.

Mit diesem Wissen können wir nun ein `supfile` erstellen, das den Quellbaum von FreeBSD-CURRENT anfordert und aktualisiert.

- Welche Dateien wollen Sie empfangen?

Dateien werden von **CVSup** in “Sammlungen” organisiert. Die erhältlichen Sammlungen werden später beschrieben. Wir wollen den Quellbaum von FreeBSD empfangen, der in der Sammlung `src-all` enthalten ist. Das `supfile` enthält pro Zeile eine Sammlung, in diesem Fall also nur eine einzige Zeile:

```
src-all
```

- Welche Versionen der Dateien werden benötigt?

  Mit **CVSup** können Sie jede Version der Quellen bekommen, da der **cvsupd**-Server seine Daten direkt aus dem **CVS**-Repository bezieht. Sie können die benötigten Versionen in den Parameterfeldern `tag=` und `date=` angeben.

  > **Warnung:** Achten Sie darauf, dass Sie das richtige `tag=`-Feld angeben. Einige Tags sind nur für spezielle Sammlungen gültig. Wenn Sie ein falsches Tag angeben oder sich verschreiben, wird **CVSup** Dateien löschen, die Sie wahrscheinlich gar nicht löschen wollten. Achten Sie insbesondere bei den `ports-*`-Sammlungen darauf, *ausschließlich* `tag=.` zu verwenden.

  Mit `tag=` wird ein symbolischer Name aus dem Repository angegeben. Es gibt zwei verschiedene Tags: Tags, die Revisionen bezeichnen und Tags, die Zweige bezeichnen. Die ersteren sind statisch und fest an eine Revision gebunden. Ein Tag, das einen Zweig bezeichnet, bezieht sich dagegen zu einem gegebenen Zeitpunkt immer auf die aktuellste Revision. Da ein Tag eines Zweiges nicht an eine bestimmte Revision gebunden ist, kann sich dessen Bedeutung von heute auf morgen ändern.

  Abschnitt A.7 zählt für Benutzer relevante Tags auf. Wenn Sie in der Konfigurationsdatei ein Tag, wie `RELENG_8`, angeben, müssen Sie diesem `tag=` vorstellen: `tag=RELENG_8`. Denken Sie daran, dass es für die Ports-Sammlung nur `tag=.` gibt.

  > **Warnung:** Achten Sie darauf, dass Sie den Namen eines Tags richtig angeben. **CVSup** kann nicht zwischen richtigen und falschen Tags unterscheiden. Wenn Sie sich bei der Angabe eines Tags vertippen, nimmt **CVSup** an, Sie hätten ein gültiges Tag angegeben, dem nur keine Dateien zugeordnet sind. Die Folge davon ist, dass Ihre vorhandenen Quellen gelöscht werden.

  Wenn Sie ein Tag angeben, das sich auf einen Zweig bezieht, erhalten Sie die aktuellsten Revisionen der Dateien auf diesem Zweig. Wenn Sie eine frühere Revision erhalten möchten, können Sie diese im `date=` Feld angeben. Einzelheiten dazu finden Sie in der Manualpage von `cvsup`.

  Wir möchten gerne FreeBSD-CURRENT beziehen und fügen die folgende Zeile *am Anfang* der Konfigurationsdatei ein:

  ```
  *default tag=.
  ```

  Eine wichtige Ausnahme ist wenn Sie weder ein `tag=`-Feld noch ein `date=`-Feld angeben. In diesem Fall erhalten Sie anstelle einer speziellen Revision die wirklichen RCS-Dateien aus dem CVS-Repository des Servers. Diese Vorgehensweise wird von Entwicklern bevorzugt, da sie mit einem eigenen Repository leicht die Entwicklungsgeschichte und Veränderungen von Dateien verfolgen können. Dieser Vorteil muss allerdings mit sehr viel Plattenplatz bezahlt werden.

- Woher sollen die Dateien bezogen werden?

  Im `host=`-Feld wird angegeben, woher `cvsup` die Dateien holen soll. Sie können hier jeden der CVSup-Spiegel angeben, doch sollten Sie einen Server in Ihrer Nähe auswählen. Für dieses Beispiel wollen wir den erfundenen Server `cvsup99.FreeBSD.org` verwenden:

```
*default host=cvsup99.FreeBSD.org
```

Bevor Sie **CVSup** laufen lassen, sollten Sie hier einen existierenden Server einsetzen. Den zu verwendenden Server können Sie auf der Kommandozeile mit `-h` *`hostname`* überschreiben.

- Wo sollen die Dateien gespeichert werden?

  Im `prefix=`-Feld teilen Sie `cvsup` mit, wo die Dateien gespeichert werden sollen. In diesem Beispiel werden wir die Quelldateien direkt im Verzeichnisbaum für Quellen `/usr/src` ablegen. Das Verzeichnis `src` ist schon in der Sammlung, die wir beziehen enthalten, so dass wir die folgende Zeile angeben:

  ```
  *default prefix=/usr
  ```

- Wo sollen die Statusinformationen von `cvsup` gespeichert werden?

  `cvsup` legt in einem Verzeichnis Statusinformationen ab, die festhalten, welche Versionen schon empfangen wurden. Wir verwenden das Verzeichnis `/var/db`:

  ```
  *default base=/var/db
  ```

  Wenn das Verzeichnis für die Statusinformationen nicht existiert, sollten Sie es jetzt anlegen, da `cvsup` ohne dieses Verzeichnis nicht startet.

- Verschiedene Einstellungen:

  Eine weitere Zeile sollte normalerweise in jedem `supfile` sein:

  ```
  *default release=cvs delete use-rel-suffix compress
  ```

  Mit `release=cvs` wird angegeben, dass der Server das FreeBSD-Haupt-Repository abfragen soll, was praktisch immer der Fall ist (die Ausnahmen werden in diesem Text nicht diskutiert).

  `delete` erlaubt es **CVSup**, Dateien zu löschen. Diese Option sollten Sie immer angeben, damit **CVSup** Ihren Quellbaum auch wirklich aktuell halten kann. **CVSup** löscht nur Dateien für die es auch verantwortlich ist. Andere Dateien, die sich in einem Baum unter Kontrolle von **CVSup** befinden, werden nicht verändert.

  Wenn Sie wirklich etwas über das obskure `use-rel-suffix` erfahren wollen, lesen Sie bitte in der Manualpage nach, ansonsten geben Sie es einfach an und vergessen es.

  Wenn Sie `compress` angeben, werden Daten auf dem Kommunikationskanal komprimiert. Wenn Sie über eine T1-Leitung oder eine schnellere Netzanbindung verfügen, brauchen Sie diese Option vielleicht nicht. In allen anderen Fällen beschleunigt sie aber den Ablauf.

- Zusammenfassung:

  Das vollständige `supfile` unseres Beispiels sieht nun so aus:

  ```
  *default tag=.
  *default host=cvsup99.FreeBSD.org
  *default prefix=/usr
  *default base=/var/db
  *default release=cvs delete use-rel-suffix compress

  src-all
  ```

A.6.3.1. Die `refuse` Datei

CVSup benutzt die Pull-Methode, das heißt wenn sich ein Client mit einem Server verbindet, erhält er eine Liste der verfügbaren Sammlungen und wählt aus diesen die herunterzuladenden Dateien aus. In der Voreinstellung wählt der Client alle Dateien aus, die zu einer gegebenen Sammlung und zu einem gegebenen Tag passen. Dieses Verhalten ist

aber nicht immer erwünscht, besonders wenn Sie die `doc`, `ports` oder `www` Verzeichnisbäume synchronisieren. Die wenigsten Leute beherrschen vier oder fünf Sprachen und benötigen Dateien mit speziellen Anpassungen für eine Sprache. Wenn Sie die Ports-Sammlung synchronisieren, können Sie anstelle von `ports-all` einzelne Ports, wie `ports-astrology` oder `ports-biology` angeben. Die `doc` und `www` Verzeichnisbäume verfügen aber nicht über Sammlungen für spezielle Sprachen. In diesem Fall müssen Sie eines der vielen eleganten Merkmale von **CVSup** benutzen: Die `refuse` Datei.

Mit einer `refuse` Datei können Sie bestimmte Dateien einer Sammlung von der Übertragung ausschließen. Der Ort der `refuse` ist *`base`*`/sup/refuse`, wobei *`base`* in Ihrem `supfile` festgelegt wurde. Wir verwenden das Verzeichnis `/var/db`, der Ort der `refuse` Datei ist daher `/var/db/sup/refuse`.

Das Format der `refuse` Datei ist einfach: Sie enthält eine Liste der Dateien und Verzeichnisse, die Sie nicht herunterladen wollen. Wenn Sie zum Beispiel die Dokumentation nicht in anderen Sprachen als Englisch lesen wollen, könnte Ihre `refuse`-Datei wie folgt aussehen:

```
doc/bn_*
doc/da_*
doc/de_*
doc/el_*
doc/es_*
doc/fr_*
doc/hu_*
doc/it_*
doc/ja_*
doc_mn_*
doc/nl_*
doc/no_*
doc/pl_*
doc/pt_*
doc/ru_*
doc/sr_*
doc/tr_*
doc/zh_*
```

Die Aufzählung setzt sich für andere Sprachen fort. Eine vollständige Liste finden Sie im FreeBSD CVS Repository (http://www.FreeBSD.org/cgi/cvsweb.cgi/).

Die `refuse` Datei spart Anwendern von **CVSup**, die über eine langsame Internetanbindung verfügen oder deren Internetverbindung zeitlich abgerechnet wird, wertvolle Zeit, da sie Dateien, die sie nicht benötigen, nicht mehr herunterladen müssen. Weitere Informationen zu `refuse` Dateien und anderen Eigenschaften von **CVSup** entnehmen Sie bitte der Manualpage.

A.6.4. Ausführen von CVSup

Wir können nun eine Aktualisierung mit der folgenden Kommandozeile starten:

```
# cvsup supfile
```

`supfile` gibt dabei das eben erstelle `supfile` an. Wenn Sie **X11** benutzen, wird `cvsup` ein GUI starten. Drücken Sie `go` und schauen Sie zu.

Das Beispiel aktualisiert die Dateien im Verzeichnisbaum `/usr/src`. Sie müssen `cvsup` als `root` starten, damit Sie die nötigen Rechte haben, die Dateien zu aktualisieren. Sie sind vielleicht ein bisschen nervös weil Sie das Programm zum ersten Mal anwenden und möchten zuerst einmal einen Testlauf durchführen. Legen Sie dazu ein temporäres Verzeichnis an und übergeben es auf der Kommandozeile von `cvsup`:

```
# mkdir /var/tmp/dest
# cvsup supfile /var/tmp/dest
```

Aktualisierungen werden dann nur in dem angegebenen Verzeichnis vorgenommen. **CVSup** untersucht die Dateien in `/usr/src`, wird aber keine dieser Dateien verändern. Die veränderten Dateien finden Sie stattdessen in `/var/tmp/dest/usr/src`. Die Statusdateien von **CVSup** werden ebenfalls nicht geändert, sondern in dem angegebenen Verzeichnis abgelegt. Wenn Sie Leseberechtigung in `/usr/src` haben, brauchen Sie das Programm noch nicht einmal unter `root` laufen zu lassen.

Wenn Sie **X11** nicht benutzen wollen oder keine GUIs mögen, sollten Sie `cvsup` wie folgt aufrufen:

```
# cvsup -g -L 2 supfile
```

`-g` verhindert den Start des GUIs. Wenn Sie kein **X11** laufen haben, passiert das automatisch, ansonsten müssen Sie diesen Schalter angeben.

Mit `-L 2` gibt **CVSup** Einzelheiten zu jeder Aktualisierung aus. Die Wortfülle der Meldungen können Sie von `-L 0` bis `-L 2` einstellen. In der Voreinstellung `-L 0` werden nur Fehlermeldungen ausgegeben.

Eine Zusammenfassung der Optionen von **CVSup** erhalten Sie mit `cvsup -H`. Genauere Informationen finden Sie in der Manualpage von **CVSup**.

Wenn Sie mit dem Ablauf der Aktualisierung zufrieden sind, können Sie **CVSup** regelmäßig aus cron(8) ausführen. In diesem Fall sollten Sie natürlich nicht das GUI benutzen.

A.6.5. CVSup Sammlungen

Die **CVSup** Sammlungen sind hierarchisch organisiert. Es gibt wenige große Sammlungen, die in kleinere Teilsammlungen unterteilt sind. Wenn Sie eine große Sammlung beziehen, entspricht das dem Beziehen aller Teilsammlungen. Der Hierarchie der Sammlung wird in der folgenden Aufzählung durch Einrückungen dargestellt.

Die am häufigsten benutzen Sammlungen sind `src-all` und `ports-all`. Die anderen Sammlungen werden von wenigen Leuten zu speziellen Zwecken benutzt und es kann sein, dass diese nicht auf allen Spiegeln zur Verfügung stehen.

`cvs-all release=cvs`

Das FreeBSD-Haupt-Repository einschließlich der Kryptographie-Module.

`distrib release=cvs`

Dateien, die zum Verteilen und Spiegeln von FreeBSD benötigt werden.

`doc-all release=cvs`

Quellen des FreeBSD-Handbuchs und weiterer Dokumentation. Diese Sammlung enthält nicht die FreeBSD-Webseite.

`ports-all release=cvs`

Die FreeBSD-Ports-Sammlung.

> **Wichtig:** Wenn Sie nicht die gesamte Ports-Sammlung (`ports-all`) aktualisieren wollen, sondern nur eine der nachstehend aufgeführten Teilsammlungen, aktualisieren Sie *immer* die Teilsammlung `ports-base`. Diese Teilsammlung enthält das Bausystem der Ports. Immer wenn `ports-base` geändert wird, ist es so gut wie sicher, dass diese Änderung auch tatsächlich von einem Port benutzt wird. Der Bau eines Ports, der auf Änderungen im Bausystem angewiesen wird, wird fehlschlagen, wenn das Bausystem noch auf einem alten Stand ist. Aktualisieren Sie vor allen Dingen `ports-base`, wenn Sie bei einem Bau merkwürdige Fehlermeldungen erhalten und kein aktuelles Bausystem benutzen.

> **Wichtig:** Wenn Sie die Datei `ports/INDEX` selbst erzeugen, brauchen Sie unbedingt die Sammlung `ports-all` (den ganzen Ports-Baum). Es ist nicht möglich, `ports/INDEX` nur mit einem Teilbaum zu erstellen. Lesen Sie dazu bitte die FAQ (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/faq/applications.html#MAKE-INDEX).

`ports-accessibility release=cvs`

Werkzeuge für behinderte Benutzer.

`ports-arabic release=cvs`

Arabische Sprachunterstützung.

`ports-archivers release=cvs`

Werkzeuge zum Archivieren.

`ports-astro release=cvs`

Astronomie-Programme.

`ports-audio release=cvs`

Audio-Programme.

`ports-base release=cvs`

Das Bausystem der Ports-Sammlung. Dazu gehören verschiedene Dateien in den Unterverzeichnissen `Mk/` und `Tools/` von `/usr/ports`.

> **Anmerkung:** Aktualisieren Sie diese Teilsammlung *jedes Mal*, wenn Sie einen Teil der Ports-Sammlung aktualisieren. Lesen Sie dazu auch den obigen Hinweis zur Ports-Sammlung.

`ports-benchmarks release=cvs`

Benchmarks.

`ports-biology release=cvs`

Biologie.

`ports-cad release=cvs`

Computer Aided Design Werkzeuge.

`ports-chinese release=cvs`

Chinesische Sprachunterstützung.

`ports-comms release=cvs`

Programme zur Datenkommunikation.

`ports-converters release=cvs`

Zeichensatz Konvertierer.

`ports-databases release=cvs`

Datenbanken.

`ports-deskutils release=cvs`

Sachen, die sich vor dem Computer-Zeitalter auf dem Schreibtisch befanden.

`ports-devel release=cvs`

Werkzeuge für Entwickler.

`ports-dns release=cvs`

Software für DNS.

`ports-editors release=cvs`

Editoren.

`ports-emulators release=cvs`

Programme, die andere Betriebssysteme emulieren.

`ports-finance release=cvs`

Finanz-Anwendungen.

`ports-ftp release=cvs`

Werkzeuge für FTP Clients und Server.

`ports-games release=cvs`

Spiele.

`ports-german release=cvs`

Deutsche Sprachunterstützung.

`ports-graphics release=cvs`

Graphik-Programme.

`ports-hebrew release=cvs`

Hebräische Sprachunterstützung.

`ports-hungarian release=cvs`

Ungarische Sprachunterstützung.

`ports-irc release=cvs`

Internet Relay Chat Werkzeuge.

`ports-japanese release=cvs`

Japanische Sprachunterstützung.

`ports-java release=cvs`

Java Werkzeuge.

`ports-korean release=cvs`

Koreanische Sprachunterstützung.

`ports-lang release=cvs`

Programmiersprachen.

`ports-mail release=cvs`

E-Mail Programme.

`ports-math release=cvs`

Programme zur numerischen Mathematik.

`ports-misc release=cvs`

Verschiedene Werkzeuge.

`ports-multimedia release=cvs`

Multimedia-Anwendungen.

`ports-net release=cvs`

Netzwerk-Programme.

`ports-net-im release=cvs`

Diverse Instant-Messenger.

`ports-net-mgmt release=cvs`

Software zum Verwalten von Netzwerken.

`ports-net-p2p release=cvs`

Software für die Nutzung von Peer-to-Peer-Netzwerken.

`ports-news release=cvs`

USENET News Werkzeuge.

`ports-palm release=cvs`

Programme für den Palm™.

`ports-polish release=cvs`

Polnische Sprachunterstützung.

`ports-ports-mgmt release=cvs`

Werkzeuge zum Management von Ports und Paketen.

`ports-portuguese release=cvs`

Portugiesische Sprachunterstützung.

`ports-print release=cvs`

Druckprogramme.

`ports-russian release=cvs`

Russische Sprachunterstützung.

`ports-science release=cvs`

Wissenschaft.

`ports-security release=cvs`

Werkzeuge zum Thema Sicherheit.

`ports-shells release=cvs`

Kommandozeilen-Shells.

`ports-sysutils release=cvs`

System-Werkzeuge.

`ports-textproc release=cvs`

Programme zur Textverarbeitung (ohne Desktop Publishing).

`ports-ukrainian release=cvs`

Ukrainische Sprachunterstützung.

`ports-vietnamese release=cvs`

Vietnamesische Sprachunterstützung.

`ports-www release=cvs`

Software rund um das World Wide Web.

`ports-x11 release=cvs`

X-Window Programme.

`ports-x11-clocks release=cvs`

X11-Uhren.

`ports-x11-drivers release=cvs`

X11-Treiber.

`ports-x11-fm release=cvs`

X11-Dateiverwalter.

`ports-x11-fonts release=cvs`

X11-Zeichensätze und Werkzeuge dazu.

`ports-x11-toolkits release=cvs`

X11-Werkzeuge.

`ports-x11-servers release=cvs`

X11-Server.

`ports-x11-themes release=cvs`

X11-Themes.

`ports-x11-wm release=cvs`

X11-Fensterverwalter.

`projects-all release=cvs`

Quelltexte der verschiedenen FreeBSD-Projekte.

`src-all release=cvs`

Die FreeBSD-Quellen einschließlich der Kryptographie-Module.

`src-base release=cvs`

Verschiedene Dateien unter `/usr/src`.

`src-bin release=cvs`

Benutzer-Werkzeuge die im Einzelbenutzermodus gebraucht werden (`/usr/src/bin`).

`src-cddl release=cvs`

Werkzeuge und Bibliotheken, die der CDDL-Lizenz unterliegen (`/usr/src/cddl`).

`src-contrib release=cvs`

Werkzeuge und Bibliotheken, die nicht aus dem FreeBSD Project stammen und wenig verändert übernommen werden. (`/usr/src/contrib`).

`src-crypto release=cvs`

Kryptographische Werkzeuge und Bibliotheken, die nicht aus dem FreeBSD Project stammen und wenig verändert übernommen werden. (`/usr/src/crypto`).

`src-eBones release=cvs`

Kerberos und DES (`/usr/src/eBones`). Wird in aktuellen Releases von FreeBSD nicht benutzt.

`src-etc release=cvs`

Konfigurationsdateien des Systems (`/usr/src/etc`).

`src-games release=cvs`

Spiele (`/usr/src/games`).

`src-gnu release=cvs`

Werkzeuge, die unter der GNU Public License stehen (`/usr/src/gnu`).

`src-include release=cvs`

Header Dateien (`/usr/src/include`).

`src-kerberos5 release=cvs`

Kerberos5 (`/usr/src/kerberos5`).

`src-kerberosIV release=cvs`

KerberosIV (`/usr/src/kerberosIV`).

`src-lib release=cvs`

Bibliotheken (`/usr/src/lib`).

`src-libexec release=cvs`

Systemprogramme, die von anderen Programmen ausgeführt werden (`/usr/src/libexec`).

`src-release release=cvs`

Dateien, die zum Erstellen eines FreeBSD Releases notwendig sind (`/usr/src/release`).

`src-rescue release=cvs`

Statisch gelinkte Programme zur Wiederherstellung eines defekten Systems. Lesen Sie dazu auch die Manualpage rescue(8) (`/usr/src/rescue`).

`src-sbin release=cvs`

Werkzeuge für den Einzelbenutzermodus (`/usr/src/sbin`).

`src-secure release=cvs`

Kryptographische Bibliotheken und Befehle (`/usr/src/secure`).

`src-share release=cvs`

Dateien, die von mehreren Systemen gemeinsam benutzt werden können (`/usr/src/share`).

`src-sys release=cvs`

Der Kernel (`/usr/src/sys`).

`src-sys-crypto release=cvs`

Kryptographie Quellen des Kernels (`/usr/src/sys/crypto`).

`src-tools release=cvs`

Verschiedene Werkzeuge zur Pflege von FreeBSD (`/usr/src/tools`).

`src-usrbin release=cvs`

Benutzer-Werkzeuge (`/usr/src/usr.bin`).

`src-usrsbin release=cvs`

System-Werkzeuge (`/usr/src/usr.sbin`).

`www release=cvs`

Die Quellen der FreeBSD-WWW-Seite.

`distrib release=self`

Die Konfigurationsdateien des **CVSup** Servers. Diese werden von den **CVSup** benutzt.

`gnats release=current`

Die GNATS Datenbank, in der Problemberichte verwaltet werden.

`mail-archive release=current`

Das Archiv der FreeBSD-Mailinglisten.

`www release=current`

Die formatierten Dateien der FreeBSD-WWW-Seite (nicht die Quellen). Diese werden von den WWW-Spiegeln benutzt.

A.6.6. Weiterführende Informationen

Die **CVSup** FAQ und weitere Informationen über **CVSup** finden Sie auf The CVSup Home Page (http://www.cvsup.org).

FreeBSD spezifische Diskussionen über **CVSup** finden auf der Mailingliste FreeBSD technical discussions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers) statt. Dort und auf der Liste FreeBSD announcements (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce) werden neue Versionen von **CVSup** angekündigt.

Bei Fragen und Problemberichten zu **CVSup** lesen Sie bitte die CVSup FAQ (http://www.cvsup.org/faq.html#bugreports).

A.6.7. CVSup-Server

Die folgende Aufzählung enthält CVSup Server für FreeBSD:

Hauptserver, Hauptspiegel, Armenien, Australien, Brasilien, Dänemark, Deutschland, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Japan, Korea, Lettland, Litauen, Niederlande, Norwegen, Polen, Russland, Schweden, Schweiz, Slowakische Republik, Slowenien, Spanien, Südafrika, Taiwan, Tschechische Republik, Ukraine, USA.

(aktualisiert am: UTC)

Hauptserver

- cvsup.FreeBSD.org

Hauptspiegel

- cvsup1.FreeBSD.org
- cvsup3.FreeBSD.org
- cvsup4.FreeBSD.org
- cvsup5.FreeBSD.org
- cvsup6.FreeBSD.org
- cvsup7.FreeBSD.org
- cvsup8.FreeBSD.org
- cvsup9.FreeBSD.org
- cvsup10.FreeBSD.org
- cvsup11.FreeBSD.org
- cvsup12.FreeBSD.org
- cvsup14.FreeBSD.org
- cvsup15.FreeBSD.org

- cvsup18.FreeBSD.org

Armenien

- cvsup1.am.FreeBSD.org

Australien

- cvsup.au.FreeBSD.org

Brasilien

- cvsup2.br.FreeBSD.org

Dänemark

- cvsup.dk.FreeBSD.org
- cvsup2.dk.FreeBSD.org

Deutschland

- cvsup.de.FreeBSD.org
- cvsup2.de.FreeBSD.org
- cvsup3.de.FreeBSD.org
- cvsup4.de.FreeBSD.org
- cvsup5.de.FreeBSD.org
- cvsup6.de.FreeBSD.org
- cvsup7.de.FreeBSD.org
- cvsup8.de.FreeBSD.org

Estland

- cvsup.ee.FreeBSD.org

Finnland

- cvsup.fi.FreeBSD.org

Frankreich

- cvsup.fr.FreeBSD.org
- cvsup1.fr.FreeBSD.org
- cvsup3.fr.FreeBSD.org
- cvsup5.fr.FreeBSD.org
- cvsup8.fr.FreeBSD.org

Griechenland

- cvsup.gr.FreeBSD.org

Irland

- cvsup.ie.FreeBSD.org
- cvsup2.ie.FreeBSD.org

Island

- cvsup.is.FreeBSD.org

Italien

- cvsup.it.FreeBSD.org

Japan

- cvsup.jp.FreeBSD.org
- cvsup2.jp.FreeBSD.org
- cvsup3.jp.FreeBSD.org
- cvsup4.jp.FreeBSD.org
- cvsup5.jp.FreeBSD.org
- cvsup6.jp.FreeBSD.org

Korea

- cvsup.kr.FreeBSD.org

Lettland

- cvsup.lv.FreeBSD.org

Litauen

- cvsup.lt.FreeBSD.org

Niederlande

- cvsup.nl.FreeBSD.org
- cvsup2.nl.FreeBSD.org
- cvsup3.nl.FreeBSD.org

Norwegen

- cvsup.no.FreeBSD.org

Polen

- cvsup.pl.FreeBSD.org
- cvsup3.pl.FreeBSD.org

Russland

- cvsup3.ru.FreeBSD.org
- cvsup5.ru.FreeBSD.org
- cvsup6.ru.FreeBSD.org

Schweden

- cvsup.se.FreeBSD.org
- cvsup2.se.FreeBSD.org

Schweiz

- cvsup.ch.FreeBSD.org

Slowakische Republik

- cvsup.sk.FreeBSD.org

Slowenien

- cvsup.si.FreeBSD.org

Spanien

- cvsup.es.FreeBSD.org
- cvsup2.es.FreeBSD.org
- cvsup3.es.FreeBSD.org

Südafrika

- cvsup.za.FreeBSD.org

Taiwan

- cvsup.tw.FreeBSD.org
- cvsup3.tw.FreeBSD.org
- cvsup6.tw.FreeBSD.org
- cvsup10.tw.FreeBSD.org
- cvsup11.tw.FreeBSD.org
- cvsup12.tw.FreeBSD.org
- cvsup13.tw.FreeBSD.org

Tschechische Republik

- cvsup.cz.FreeBSD.org

Ukraine

- cvsup5.ua.FreeBSD.org
- cvsup6.ua.FreeBSD.org

USA

- cvsup1.us.FreeBSD.org
- cvsup3.us.FreeBSD.org
- cvsup4.us.FreeBSD.org
- cvsup5.us.FreeBSD.org
- cvsup6.us.FreeBSD.org
- cvsup7.us.FreeBSD.org
- cvsup8.us.FreeBSD.org

- cvsup9.us.FreeBSD.org
- cvsup11.us.FreeBSD.org
- cvsup12.us.FreeBSD.org
- cvsup13.us.FreeBSD.org
- cvsup14.us.FreeBSD.org
- cvsup15.us.FreeBSD.org
- cvsup18.us.FreeBSD.org

A.7. CVS-Tags

Wenn Sie Quellen mit **CVS** oder **CVSup** erhalten oder aktualisieren wollen, müssen Sie ein Tag angeben. Ein Tag kann einen bestimmten FreeBSD-Zweig oder einen bestimmten Zeitpunkt (Release-Tag) bestimmen.

A.7.1. Tags für Zweige

Mit Ausnahme von `HEAD` (das immer ein gültiges Tag ist), können die folgenden Tags nur im `src/`-Quellbaum verwendet werden. Die Quellbäume `ports/`, `doc/` und `www/` sind nicht verzweigt.

HEAD

Symbolischer Name für den Hauptzweig, auch FreeBSD-CURRENT genannt. Dies ist die Vorgabe, wenn keine Revision angegeben wird.

In **CVSup** wird dieses Tag mit einem `.` (Punkt) bezeichnet.

> **Anmerkung:** In **CVS** ist das die Vorgabe, wenn Sie kein Tag oder eine Revision angeben. Außer Sie wollen einen -STABLE Rechner auf -CURRENT aktualisieren, ist es *nicht* ratsam, die -CURRENT Quellen auf einem -STABLE Rechner einzuspielen.

RELENG_8

Der Entwicklungszweig für FreeBSD-8.X, auch bekannt als FreeBSD 8-STABLE.

RELENG_8_2

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 8.2 durchgeführt werden.

RELENG_8_1

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 8.1 durchgeführt werden.

RELENG_8_0

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 8.0 durchgeführt werden.

RELENG_7

Der Entwicklungszweig für FreeBSD-7.X, auch als FreeBSD 7-STABLE bekannt.

RELENG_7_4

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 7.4 durchgeführt werden.

RELENG_7_3

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 7.3 durchgeführt werden.

RELENG_7_2

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 7.2 durchgeführt werden.

RELENG_7_1

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 7.1 durchgeführt werden.

RELENG_7_0

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 7.0 durchgeführt werden.

RELENG_6

Der Entwicklungszweig für FreeBSD-6.X, auch als FreeBSD 6-STABLE bekannt.

RELENG_6_4

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 6.4 durchgeführt werden.

RELENG_6_3

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 6.3 durchgeführt werden.

RELENG_6_2

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 6.2 durchgeführt werden.

RELENG_6_1

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 6.1 durchgeführt werden.

RELENG_6_0

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 6.0 durchgeführt werden.

RELENG_5

Der FreeBSD 5.X Entwicklungszweig, der auch FreeBSD 5-STABLE genannt wird.

RELENG_5_5

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.5 durchgeführt werden.

RELENG_5_4

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.4 durchgeführt werden.

RELENG_5_3

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.3 durchgeführt werden.

RELENG_5_2

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.2 und FreeBSD 5.2.1 durchgeführt werden.

RELENG_5_1

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.1 durchgeführt werden.

RELENG_5_0

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 5.0 durchgeführt werden.

RELENG_4

Der FreeBSD 4.X Entwicklungszweig, der auch FreeBSD 4-STABLE genannt wird.

RELENG_4_11

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.11 durchgeführt werden.

RELENG_4_10

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.10 durchgeführt werden.

RELENG_4_9

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.9 durchgeführt werden.

RELENG_4_8

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.8 durchgeführt werden.

RELENG_4_7

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.7 durchgeführt werden.

RELENG_4_6

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.6 und FreeBSD 4.6.2 durchgeführt werden.

RELENG_4_5

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.5 durchgeführt werden.

RELENG_4_4

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.4 durchgeführt werden.

RELENG_4_3

Der Zweig, auf dem sicherheitsrelevante oder kritische Fehlerbehebungen für FreeBSD 4.3 durchgeführt werden.

RELENG_3

Der FreeBSD-3.X Entwicklungszweig, der auch 3.X-STABLE genannt wird.

RELENG_2_2

Der FreeBSD-2.2.X Entwicklungszweig, der auch 2.2-STABLE genannt wird.

A.7.2. Release-Tags

Diese Tags geben den Zeitpunkt an, an dem eine bestimme FreeBSD-Version veröffentlicht wurde. Das Erstellen einer Release ist in den Dokumenten Release Engineering Information (http://www.FreeBSD.org/releng/) und Release Process (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/relengrelease-proc.html) beschrieben. Der `src`-Baum benutzt Tags, deren Namen mit `RELENG_` anfangen. Die Bäume `ports` und `doc` benutzen Tags, deren Namen mit `RELEASE` anfangen. Im Baum `www` werden keine Release-Tags verwendet.

RELENG_8_2_0_RELEASE

FreeBSD 8.2

RELENG_8_1_0_RELEASE

FreeBSD 8.1

RELENG_8_0_0_RELEASE

: FreeBSD 8.0

RELENG_7_4_0_RELEASE

: FreeBSD 7.4

RELENG_7_3_0_RELEASE

: FreeBSD 7.3

RELENG_7_2_0_RELEASE

: FreeBSD 7.2

RELENG_7_1_0_RELEASE

: FreeBSD 7.1

RELENG_7_0_0_RELEASE

: FreeBSD 7.0

RELENG_6_4_0_RELEASE

: FreeBSD 6.4

RELENG_6_3_0_RELEASE

: FreeBSD 6.3

RELENG_6_2_0_RELEASE

: FreeBSD 6.2

RELENG_6_1_0_RELEASE

: FreeBSD 6.1

RELENG_6_0_0_RELEASE

: FreeBSD 6.0

RELENG_5_5_0_RELEASE

: FreeBSD 5.5

RELENG_5_4_0_RELEASE

: FreeBSD 5.4

RELENG_4_11_0_RELEASE

: FreeBSD 4.11

RELENG_5_3_0_RELEASE

: FreeBSD 5.3

RELENG_4_10_0_RELEASE

: FreeBSD 4.10

RELENG_5_2_1_RELEASE

: FreeBSD 5.2.1

RELENG_5_2_0_RELEASE

: FreeBSD 5.2

RELENG_4_9_0_RELEASE

: FreeBSD 4.9

RELENG_5_1_0_RELEASE

: FreeBSD 5.1

RELENG_4_8_0_RELEASE

: FreeBSD 4.8

RELENG_5_0_0_RELEASE

: FreeBSD 5.0

RELENG_4_7_0_RELEASE

: FreeBSD 4.7

RELENG_4_6_2_RELEASE

: FreeBSD 4.6.2

RELENG_4_6_1_RELEASE

: FreeBSD 4.6.1

RELENG_4_6_0_RELEASE

: FreeBSD 4.6

RELENG_4_5_0_RELEASE

: FreeBSD 4.5

RELENG_4_4_0_RELEASE

: FreeBSD 4.4

RELENG_4_3_0_RELEASE

: FreeBSD 4.3

RELENG_4_2_0_RELEASE

: FreeBSD 4.2

RELENG_4_1_1_RELEASE

: FreeBSD 4.1.1

RELENG_4_1_0_RELEASE

: FreeBSD 4.1

RELENG_4_0_0_RELEASE

: FreeBSD 4.0

RELENG_3_5_0_RELEASE

: FreeBSD-3.5

RELENG_3_4_0_RELEASE

: FreeBSD-3.4

RELENG_3_3_0_RELEASE

: FreeBSD-3.3

RELENG_3_2_0_RELEASE

: FreeBSD-3.2

RELENG_3_1_0_RELEASE

: FreeBSD-3.1

RELENG_3_0_0_RELEASE

: FreeBSD-3.0

RELENG_2_2_8_RELEASE

: FreeBSD-2.2.8

RELENG_2_2_7_RELEASE

: FreeBSD-2.2.7

RELENG_2_2_6_RELEASE

: FreeBSD-2.2.6

RELENG_2_2_5_RELEASE

: FreeBSD-2.2.5

RELENG_2_2_2_RELEASE

: FreeBSD-2.2.2

RELENG_2_2_1_RELEASE

: FreeBSD-2.2.1

RELENG_2_2_0_RELEASE

FreeBSD-2.2.0

A.8. AFS-Server

Die folgende Aufzählung enthält AFS Server für FreeBSD:

Schweden

Die Dateien sind unter dem Pfad `/afs/stacken.kth.se/ftp/pub/FreeBSD/` erreichbar.

```
stacken.kth.se          # Stacken Computer Club, KTH, Sweden
130.237.234.43          #hot.stacken.kth.se
130.237.237.230         #fishburger.stacken.kth.se
130.237.234.3           #milko.stacken.kth.se
```

Betreuer <`ftp@stacken.kth.se`>

A.9. rsync-Server

rsync wird ähnlich wie rcp(1) verwendet, besitzt aber mehr Optionen und verwendet das “rsync remote-update” Protokoll, das nur geänderte Dateien überträgt und damit viel schneller als ein normaler Kopiervorgang ist. **rsync** ist sehr nützlich, wenn Sie einen FreeBSD-FTP-Spiegel oder einen CVS-Spiegel betreiben. Das Programm ist für viele Betriebssysteme erhältlich, mit FreeBSD können Sie den Port `net/rsync` oder das fertige Paket benutzen. Die folgenden Server stellen FreeBSD über das **rsync** Protokoll zur Verfügung:

Großbritannien

rsync://rsync.mirrorservice.org/

Verfügbare Sammlungen:

- sites/ftp.freebsd.org: Kompletter Spiegel des FreeBSD-FTP-Servers.

Niederlande

rsync://ftp.nl.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD-FTP-Servers.

Russland

rsync://ftp.mtu.ru/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD-FTP-Servers.

- FreeBSD-gnats: Die GNATS-Datenbank zur Verwaltung von Problemberichten.
- FreeBSD-Archive: Ein Spiegel des FreeBSD-Archive-FTP-Servers.

Schweden

rsync://ftp4.se.freebsd.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD-FTP-Servers.

Taiwan

rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD-FTP-Servers.

Tschechische Republik

rsync://ftp.cz.FreeBSD.org/

Verfügbare Sammlungen:

- ftp: Unvollständiger Spiegel des FreeBSD-FTP-Servers.
- FreeBSD: Vollständiger Spiegel des FreeBSD-FTP-Servers.

USA

rsync://ftp-master.FreeBSD.org/

Dieser Server darf nur von primären Spiegeln benutzt werden.

Verfügbare Sammlungen:

- FreeBSD: Das Hauptarchiv des FreeBSD FTP Servers.
- acl: Die primäre ACL-Liste.

rsync://ftp13.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD-FTP-Servers.

Fußnoten

1. Tags sind symbolische Namen, die im Repository vergeben werden.
2. Abkürzung für “CVS Through eMail”

Anhang B. Bibliografie

Übersetzt von Frank Gründer <elwood@mc5sys.in-berlin.de>

Während die Manualpages die endgültige Auskunft über bestimmte Teile des FreeBSD-Betriebssystems geben, so können sie jedoch nicht darstellen, wie man die einzelnen Teile zusammenfügt, um ein vollständig laufendes Betriebssystem herzustellen. Daher gibt es keinen Ersatz für ein gutes Buch über die Administration von UNIX Systemen und ein gutes Benutzerhandbuch.

In der Regel handelt es sich im folgenden Kapitel um englische Ausgaben der genannten Werke. Übersetzungen oder Ausgaben in anderen Sprachen sind mit entsprechenden Hinweisen versehen.

B.1. Bücher und Magazine speziell für FreeBSD

Internationale Bücher und Magazine:

- Using FreeBSD (http://jdli.tw.FreeBSD.org/publication/book/freebsd2/index.htm), herausgegeben von Drmaster (http://www.drmaster.com.tw/), 1997 (in traditionellem Chinesisch). ISBN 9-578-39435-7.
- FreeBSD Unleashed (in vereinfachtem Chinesisch), herausgegeben von China Press (http://www.hzbook.com/). ISBN 7-111-10201-0.
- FreeBSD From Scratch First Edition (in vereinfachtem Chinesisch), herausgegeben von China Press (http://www.hzbook.com/). ISBN 7-111-07482-3.
- FreeBSD From Scratch Second Edition (in vereinfachtem Chinesisch), herausgegeben von China Press (http://www.hzbook.com/). ISBN 7-111-10286-X.
- FreeBSD Handbook Second Edition (in vereinfachtem Chinesisch), herausgegeben von Posts & Telecom Press (http://www.ptpress.com.cn/). ISBN 7-115-10541-3.
- FreeBSD 3.x Internet (in vereinfachtem Chinesisch), herausgegeben von Tsinghua University Press (http://www.tup.tsinghua.edu.cn/). ISBN 7-900625-66-6.
- FreeBSD & Windows (in vereinfachtem Chinesisch), herausgegeben von China Railway Publishing House (http://www.tdpress.com/). ISBN 7-113-03845-X.
- FreeBSD Internet Services HOWTO (in vereinfachtem Chinesisch), herausgegeben von China Railway Publishing House. ISBN 7-113-03423-3.
- FreeBSD for PC 98’ers (in japanischer Sprache), herausgegeben von SHUWA System Co, LTD. ISBN 4-87966-468-5 C3055 P2900E.
- FreeBSD (in japanischer Sprache), herausgegeben von CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (http://www.shoeisha.com/book/Detail.asp?bid=650) (in Japanese), published by Shoeisha Co., Ltd (http://www.shoeisha.co.jp/). ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (http://www.ascii.co.jp/pb/book1/shinkan/detail/1322785.html) (in japanischer Sprache), herausgegeben von ASCII (http://www.ascii.co.jp/). ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (japanische Übersetzung), herausgegeben von ASCII (http://www.ascii.co.jp/). ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (in deutscher Sprache), herausgegeben von Computer und Literatur Verlag (http://www.cul.de) /Vertrieb Hanser, 1998. ISBN 3-932311-31-0.

- FreeBSD 4 - Installieren, Konfigurieren, Administrieren (http://www.cul.de/freebsd.html) (in deutscher Sprache), herausgegeben von Computer und Literatur Verlag (http://www.cul.de), 2001. ISBN 3-932311-88-4.
- FreeBSD 5 – Installieren, Konfigurieren, Administrieren (http://www.cul.de/freebsd.html) (in deutscher Sprache), herausgegeben von Computer und Literatur Verlag (http://www.cul.de), 2001. ISBN 3-936546-06-1.
- FreeBSD de Luxe (http://www.mitp.de/vmi/mitp/detail/pWert/1343/) (in German), published by Verlag Modere Industrie (http://www.mitp.de), 2003. ISBN 3-8266-1343-0.
- FreeBSD Install and Utilization Manual (http://www.pc.mycom.co.jp/FreeBSD/install-manual.html) (in japanischer Sprache), herausgegeben von Mainichi Communications Inc. (http://www.pc.mycom.co.jp/), 1998. ISBN 4-8399-0112-0.
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Building Internet Server with FreeBSD (http://maxwell.itb.ac.id/)* (in indonesischer Sprache), herausgegeben von Elex Media Komputindo (http://www.elexmedia.co.id/).
- Absolute BSD: The Ultimate Guide to FreeBSD (in traditionellem Chinesisch), herausgegeben von GrandTech Press (http://www.grandtech.com.tw/), 2003. ISBN 986-7944-92-5.
- The FreeBSD 6.0 Book (http://www.twbsd.org/cht/book/) (in traditionellem Chinesisch, herausgegeben von Drmaster, 2006. ISBN 9-575-27878-X.

Englischsprachige Bücher und Magazine:

- Absolute FreeBSD, 2nd Edition: The Complete Guide to FreeBSD (http://www.absoluteFreeBSD.com/), herausgegeben von No Starch Press (http://www.nostarch.com/), 2007. ISBN: 978-1-59327-151-0
- The Complete FreeBSD (http://www.freebsdmall.com/cgi-bin/fm/bsdcomp), herausgegeben von O'Reilly (http://www.oreilly.com/), 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide (http://www.freebsd-corp-net-guide.com/), herausgegeben von Addison-Wesley (http://www.awl.com/aw/), 2002. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer (http://andrsn.stanford.edu/FreeBSD/introbook/), herausgegeben von The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, herausgegeben von Sams (http://www.samspublishing.com/), 2002. ISBN: 0672324245
- FreeBSD6 Unleashed, herausgegeben von Sams (http://www.samspublishing.com/), 2006. ISBN: 0672328755
- FreeBSD: The Complete Reference, herausgegeben von McGrawHill (http://books.mcgraw-hill.com), 2003. ISBN: 0072224096
- BSD Magazine (http://www.bsdmag.org), herausgegeben von Software Press Sp. z o.o. SK. ISBN: 1898-9144

B.2. Handbücher

- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-075-9

- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-076-7
- *UNIX in a Nutshell*. O'Reilly & Associates, Inc., 1990. ISBN 093717520X
- Mui, Linda. *What You Need To Know When You Can't Find Your UNIX System Administrator*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-104-6
- Die Ohio State University hat ein UNIX Introductory Course (http://www.cs.duke.edu/csl/docs/unix_course/) veröffentlicht, welcher auch online im HTML- und PostScriptformat verfügbar ist.

  Eine italienische Übersetzung (http://www.FreeBSD.org/doc/it_IT.ISO8859-15/books/unix-introduction/index.html) ist Teil des FreeBSD Italian Documentation Projects.
- Jpman Project, Japan FreeBSD Users Group (http://www.jp.FreeBSD.org/). FreeBSD User's Reference Manual (http://www.pc.mycom.co.jp/FreeBSD/urm.html) (japanische Übersetzung). Mainichi Communications Inc. (http://www.pc.mycom.co.jp/), 1998. ISBN4-8399-0088-4 P3800E.
- Edinburgh University (http://www.ed.ac.uk/) hat einen Online Guide (http://unixhelp.ed.ac.uk/) für Anfänger in Sachen UNIX geschrieben.

B.3. Administrations-Anleitungen

- Albitz, Paul and Liu, Cricket. *DNS and BIND*, 4th Ed. O'Reilly & Associates, Inc., 2001. ISBN 1-59600-158-4
- Computer Systems Research Group, UC Berkeley. *4.4BSD System Manager's Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-080-5
- Costales, Brian, et al. *Sendmail*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-222-0
- Frisch, Æleen. *Essential System Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-127-5
- Hunt, Craig. *TCP/IP Network Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-322-7
- Nemeth, Evi. *UNIX System Administration Handbook*. 3rd Ed. Prentice Hall, 2000. ISBN 0-13-020601-6
- Stern, Hal *Managing NFS and NIS* O'Reilly & Associates, Inc., 1991. ISBN 0-937175-75-7
- Jpman Project, Japan FreeBSD Users Group (http://www.jp.FreeBSD.org/). FreeBSD System Administrator's Manual (http://www.pc.mycom.co.jp/FreeBSD/sam.html) (japanische Übersetzung). Mainichi Communications Inc. (http://www.pc.mycom.co.jp/), 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD (http://www.eyrolles.com/Informatique/Livre/9782212114638/) 2nd Ed. (in French), Eyrolles, 2004. ISBN 2-212-11463-X.

B.4. Programmierhandbücher

- Asente, Paul, Paul, Converse, Diana, and Swick, Ralph. *X Window System Toolkit*. Digital Press, 1998. ISBN 1-55558-178-1
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3

- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed., PTR Prentice Hall, 1988. ISBN 0-13-110362-9
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7
- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9
- Spinellis, Diomidis. *Code Reading: The Open Source Perspective* (http://www.spinellis.gr/codereading/). Addison-Wesley, 2003. ISBN 0-201-79940-5
- Spinellis, Diomidis. *Code Quality: The Open Source Perspective* (http://www.spinellis.gr/codequality/). Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard and Stephen A. Rago. *Advanced Programming in the UNIX Environment*. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X
- Wells, Bill. "Writing Serial Drivers for UNIX". *Dr. Dobb's Journal*. 19(15), December 1994. pp68-71, 97-99.

B.5. Betriebssystem-Interna

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobb's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1

  Kapitel 2 dieses Buchs ist Teil des FreeBSD Documentation Projects und online (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/design-44bsd/book.html) erhältlich.
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4
- Marshall Kirk McKusick, George V. Neville-Neil. *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3

- Vahalia, Uresh. *UNIX Internals -- The New Frontiers.* Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

B.6. Sicherheits-Anleitung

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson and Gene Spafford. *Practical UNIX & Internet Security*. 2nd Ed. O'Reilly & Associates, Inc., 1996. ISBN 1-56592-148-8
- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

B.7. Hardware-Anleitung

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Die Intel Corporation veröffentlicht Dokumentationen Ihrer CPUs, Chipsets und Standards auf ihrer developer web site (http://developer.intel.com/), normalerweise als PDF-Dateien.
- Shanley, Tom. *80486 System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

B.8. UNIX® Geschichte

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Auch bekannt als das Jargon File (http://www.catb.org/~esr/jargon/html/index.html)
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5

- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Online (http://www.simson.net/ref/ugh.pdf) verfügbar.
- Don Libes, Sandy Ressler *Life with UNIX* — special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. http://www.FreeBSD.org/cgi/cvsweb.cgi/src/share/misc/bsd-family-tree oder unter `/usr/share/misc/bsd-family-tree` auf einem FreeBSD-System.
- *Networked Computer Science Technical Reports Library*. http://www.ncstrl.org/
- *Old BSD releases from the Computer Systems Research group (CSRG)*. http://www.mckusick.com/csrg/: Das Paket mit 4 CD-ROMs enthält alle BSD-Versionen von 1BSD bis 4.4BSD und 4.4BSD-Lite2 (nicht aber 2.11BSD). Die letzte CD beinhaltet auch die finalen Sourcen inklusive den SCCS Dateien.

B.9. Magazine und Journale

- *The C/C++ Users Journal*. R&D Publications Inc. ISSN 1075-2838
- *Sys Admin — The Journal for UNIX System Administrators* Miller Freeman, Inc., ISSN 1061-2688
- *freeX – Das Magazin für Linux – BSD – UNIX* (in deutscher Sprache), Computer- und Literaturverlag GmbH, ISSN 1436-7033

Anhang C. Ressourcen im Internet

Gedruckte Medien können mit der schnellen Entwicklung von FreeBSD nicht Schritt halten. Elektronische Medien sind häufig die einzige Möglichkeit, über aktuelle Entwicklungen informiert zu sein. Da FreeBSD ein Projekt von Freiwilligen ist, gibt die Benutzergemeinde selbst auch technische Unterstützung. Die Benutzergemeinde erreichen Sie am besten über E-Mail, Internetforen oder Usenet-News.

Die wichtigsten Wege, auf denen Sie die FreeBSD-Benutzergemeinde erreichen können, sind unten dargestellt. Wenn Sie weitere Ressourcen kennen, die hier fehlen, schicken Sie diese bitte an die Mailingliste des FreeBSD documentation project (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc), damit sie hier aufgenommen werden können.

C.1. Mailinglisten

Die Mailinglisten sind der direkteste Weg, um Fragen an das gesamte FreeBSD Publikum zu stellen oder eine technische Diskussion zu beginnen. Es existiert eine grosse Vielfalt von Listen mit einer Reihe von verschiedenen FreeBSD Themen. Wenn Sie ihre Fragen an die richtige Mailingliste richten können Sie viel eher mit einer passenden Antwort darauf rechnen.

Die Chartas der verschiedenen Listen sind unten wiedergegeben. *Bevor Sie sich einer Mailingliste anschließen oder E-Mails an eine Liste senden, lesen Sie bitte die Charta der Liste.* Die meisten Mitglieder unserer Mailinglisten erhalten Hunderte E-Mails zum Thema FreeBSD pro Tag. Die Chartas und Regeln, die den Gebrauch der Listen beschreiben, garantieren die hohe Qualität der Listen. Die Listen würden ihren hohen Wert für das Projekt verlieren, wenn wir weniger Regeln aufstellen würden.

> **Anmerkung:** *Um zu testen, ob Sie eine Nachricht an eine FreeBSD-Liste senden können, verwenden Sie bitte Die Liste freebsd-test (http://lists.FreeBSD.org/mailman/listinfo/freebsd-test).* Schicken Sie derartige Nachrichten bitte nicht an eine der anderen Listen.

Wenn Sie Sich nicht sicher sind, auf welcher Liste Sie Ihre Frage stellen sollen, sollten Sie den Artikel How to get best results from the FreeBSD-questions mailing list (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/freebsd-questions/index.html) lesen.

Bevor Sie eine Nachricht an eine Mailingliste senden, sollten Sie die korrekte Nutzung der Mailinglisten erlernen. Dazu gehört auch das Vermeiden von sich häufig wiederholenden Diskussionen (lesen Sie deshalb zuerst die Mailing List Frequently Asked Questions (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/mailing-list-faq/index.html)).

Alle Mailinglisten werden archiviert und können auf dem FreeBSD World Wide Web Server (http://www.FreeBSD.org/search/index.html) durchsucht werden. Das nach Schlüsselwörtern durchsuchbare Archiv bietet die hervorragende Möglichkeit, Antworten auf häufig gestellte Fragen zu finden. Nutzen Sie bitte diese Möglichkeit, bevor Sie Fragen auf einer Liste stellen. Beachten Sie auch, dass das zur Folge hat, dass die Nachrichten an die FreeBSD Mailinglisten für die Ewigkeit erhalten bleiben. Wenn Sie am Schutz ihrer Privatsphähre interessiert sind, ziehen Sie die Verwendung einer Wegwerf-E-Mail-Adresse in Betracht und schreiben Sie nur solche Nachrichten, die für die Öffentlichkeit bestimmt sind.

C.1.1. Beschreibung der Mailinglisten

Allgemeine Listen: Jeder kann die folgenden allgemeinen Listen abonnieren (und ist dazu aufgefordert):

Mailingliste	Zweck
freebsd-advocacy (http://lists.FreeBSD.org/mailman/listinfo/freebsd-advocacy)	Verbreitung von FreeBSD
freebsd-announce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce)	Wichtige Ereignisse und Meilensteine des Projekts
freebsd-arch (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch)	Architektur und Design von FreeBSD
freebsd-bugbusters (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters)	Diskussionen über die Pflege der FreeBSD Fehlerberichte-Datenbank und die dazu benutzten Werkzeuge
freebsd-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs)	Fehlerberichte
freebsd-chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat)	Nicht technische Themen, die die FreeBSD-Gemeinschaft betreffen
freebsd-chromium (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chromium)	Diskussionen zum Einsatz von Chromium unter FreeBSD
freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)	Gebrauch von FreeBSD-CURRENT
freebsd-isp (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp)	Für Internet-Service-Provider, die FreeBSD benutzen
freebsd-jobs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs)	Anstellung und Beratung im FreeBSD-Umfeld
freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions)	Benutzerfragen und technische Unterstützung
freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications)	Ankündigungen zum Thema Sicherheit

Mailingliste	Zweck
freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable)	Gebrauch von FreeBSD-STABLE
freebsd-test (http://lists.FreeBSD.org/mailman/listinfo/freebsd-test)	Schicken Sie Testnachrichten an diese Liste anstelle der wirklichen Listen

Technische Listen: Auf den folgenden Listen werden technische Diskussionen geführt. Bevor Sie eine der Listen abonnieren oder Nachrichten an sie schicken, lesen Sie sich bitte die Charta der Liste durch, da der Inhalt und Zweck dieser Listen genau festgelegt ist.

Mailingliste	Zweck
freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi)	Entwicklung von ACPI
freebsd-afs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs)	Portierung von AFS nach FreeBSD
freebsd-aic7xxx (http://lists.FreeBSD.org/mailman/listinfo/aic7xxx)	Entwicklung von Adaptec AIC 7xxx Treibern
freebsd-amd64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-amd64)	Portierung von FreeBSD auf AMD64-Systeme
freebsd-apache (http://lists.FreeBSD.org/mailman/listinfo/freebsd-apache)	Diskussion über Ports, die mit **Apache** zusammenhängen.
freebsd-arm (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arm)	Portierung von FreeBSD auf ARM®-Prozessoren
freebsd-atm (http://lists.FreeBSD.org/mailman/listinfo/freebsd-atm)	Benutzung von ATM-Netzen mit FreeBSD
freebsd-binup (http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup)	Design und Entwicklung eines Systems, das es erlaubt, ein FreeBSD-System mit binären Paketen zu aktualisieren
freebsd-bluetooth (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bluetooth)	Bluetooth unter FreeBSD verwenden
freebsd-cluster (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cluster)	Benutzung von FreeBSD in einem Cluster

Mailingliste	Zweck
freebsd-cvsweb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb)	Pflege von CVSweb
freebsd-database (http://lists.FreeBSD.org/mailman/listinfo/freebsd-database)	Diskussion über Datenbanken und Datenbankprogrammierung unter FreeBSD
freebsd-desktop (http://lists.FreeBSD.org/mailman/listinfo/freebsd-desktop)	FreeBSD als Desktop verwenden und verbessern
freebsd-doc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)	Erstellen der FreeBSD-Dokumentation
freebsd-drivers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-drivers)	Gerätetreiber für FreeBSD schreiben
freebsd-eclipse (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eclipse)	Für FreeBSD-Anwender, die die Eclipse IDE, deren Werkzeuge, Anwendungen und Ports einsetzen
freebsd-embedded (http://lists.FreeBSD.org/mailman/listinfo/freebsd-embedded)	FreeBSD in eingebetteten Anwendungen einsetzen
freebsd-emulation (http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation)	Emulation anderer Systeme wie Linux, MS-DOS oder Windows
freebsd-eol (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eol)	Support für FreeBSD-bezogene Software, die vom FreeBSD Project offiziell nicht mehr unterstützt wird.
freebsd-firewire (http://lists.FreeBSD.org/mailman/listinfo/freebsd-firewire)	Technische Diskussion über FireWire (iLink, IEEE 1394)
freebsd-fs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs)	Dateisysteme
freebsd-gecko (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gecko)	Angelegenheiten zur **Gecko Rendering Engine**

Mailingliste	Zweck
freebsd-geom (http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom)	Diskussion über GEOM
freebsd-gnome (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome)	Portierung von **GNOME** und **GNOME**-Anwendungen
freebsd-hackers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers)	Allgemeine technische Diskussionen
freebsd-hardware (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware)	Allgemeine Diskussion über Hardware, auf der FreeBSD läuft
freebsd-i18n (http://lists.FreeBSD.org/mailman/listinfo/freebsd-i18n)	Internationalisierung von FreeBSD
freebsd-ia32 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia32)	FreeBSD für die IA-32 (Intel x86) Plattform
freebsd-ia64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64)	Portierung von FreeBSD auf Intels neue IA64-Systeme
freebsd-ipfw (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw)	Technische Diskussion über die Neubearbeitung der IP-Firewall Quellen
freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn)	Für Entwickler des ISDN-Systems
freebsd-java (http://lists.FreeBSD.org/mailman/listinfo/freebsd-java)	Für Java Entwickler und Leute, die JDKs nach FreeBSD portieren
freebsd-kde (https://mail.kde.org/mailman/listinfo/kde-freebsd)	Portierung von **KDE** und **KDE**-Anwendungen
freebsd-lfs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-lfs)	Portierung von LFS nach FreeBSD
freebsd-mips (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips)	Portierung von FreeBSD zu MIPS®
freebsd-mobile (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile)	Diskussionen über mobiles Rechnen
freebsd-mono (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mono)	Mono und C# Anwendungen auf FreeBSD

Mailingliste	Zweck
freebsd-mozilla (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mozilla)	Portierung von **Mozilla** nach FreeBSD
freebsd-multimedia (http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia)	Multimedia Anwendungen
freebsd-new-bus (http://lists.FreeBSD.org/mailman/listinfo/freebsd-new-bus)	Technische Diskussionen über die Architektur von Bussen
freebsd-net (http://lists.FreeBSD.org/mailman/listinfo/freebsd-net)	Diskussion über Netzwerke und den TCP/IP Quellcode
freebsd-office (http://lists.FreeBSD.org/mailman/listinfo/freebsd-office)	Office-Anwendungen für FreeBSD
freebsd-performance (http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance)	Fragen zur Optimierung der Leistung stark ausgelasteter Systeme
freebsd-perl (http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance)	Pflege der portierten Perl-Anwendungen.
freebsd-pf (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf)	Diskussionen und Fragen zu *packet filter* als Firewallsystem.
freebsd-platforms (http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms)	Portierungen von FreeBSD auf nicht-Intel Architekturen
freebsd-ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports)	Diskussion über die Ports-Sammlung
freebsd-ports-announce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-announce)	Wichtige Neuigkeiten und Anweisungen zur Ports-Sammlung
freebsd-ports-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs)	Diskussion über Fehler und PRs der Ports
freebsd-ppc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc)	Portierung von FreeBSD auf den PowerPC

Mailingliste	Zweck
freebsd-proliant (http://lists.FreeBSD.org/mailman/listinfo/freebsd-proliant)	Technische Diskussionen zum Einsatz von FreeBSD auf der ProLiant-Serverplattform von HP.
freebsd-python (http://lists.FreeBSD.org/mailman/listinfo/freebsd-python)	FreeBSD-spezifische Diskussionen zu Python
freebsd-rc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-rc)	Diskussion über das `rc.d`-System sowie dessen Weiterentwicklung
freebsd-realtime (http://lists.FreeBSD.org/mailman/listinfo/freebsd-realtime)	Entwicklung von Echtzeiterweiterungen für FreeBSD
freebsd-ruby (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ruby)	FreeBSD-spezifische Diskussionen zu Ruby
freebsd-scsi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi)	Diskussion über das SCSI-Subsystem
freebsd-security (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security)	Sicherheitsthemen
freebsd-small (http://lists.FreeBSD.org/mailman/listinfo/freebsd-small)	Gebrauch von FreeBSD in eingebetteten Systemen (obsolet; verwenden Sie stattdessen freebsd-embedded (http://lists.FreeBSD.org/mailman/listinfo/freebsd-embedded))
freebsd-sparc64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-sparc64)	Portierung von FreeBSD auf SPARC Systeme
freebsd-standards (http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards)	Konformität von FreeBSD mit den C99- und POSIX-Standards
freebsd-sysinstall (http://lists.FreeBSD.org/mailman/listinfo/freebsd-sysinstall)	sysinstall(8) Entwicklung
freebsd-threads (http://lists.FreeBSD.org/mailman/listinfo/freebsd-threads)	Leichgewichtige Prozesse (*Threads*) in FreeBSD

Mailingliste	Zweck
freebsd-tilera (http://lists.FreeBSD.org/mailman/listinfo/freebsd-tilera)	Diskussionen zur Portierung von FreeBSD auf die Tilera-CPU-Familie.
freebsd-tokenring (http://lists.FreeBSD.org/mailman/listinfo/freebsd-tokenring)	Token-Ring Unterstützung in FreeBSD
freebsd-toolchain (http://lists.FreeBSD.org/mailman/listinfo/freebsd-toolchain)	Wartung der FreeBSD-Toolchain
freebsd-usb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb)	USB-Unterstützung in FreeBSD
freebsd-virtualization (http://lists.FreeBSD.org/mailman/listinfo/freebsd-virtualization)	Diskussion über verschiedene Virtualisierungsverfahren, die von FreeBSD unterstützt werden
freebsd-vuxml (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vuxml)	Diskussion über die Infratruktur von VuXML
freebsd-x11 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-x11)	Wartung und Unterstützung von X11 auf FreeBSD
freebsd-xen (http://lists.FreeBSD.org/mailman/listinfo/freebsd-xen)	Diskussionen über die FreeBSD Portierung auf Xen™ - Implementierung und Verwendung
freebsd-xfce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-xfce)	Portierung und Wartung von **XFCE**
freebsd-zope (http://lists.FreeBSD.org/mailman/listinfo/freebsd-zope)	**Zope** für FreeBSD - Portierung und Wartung

Eingeschränkte Listen: Die folgenden Listen wenden sich an Zielgruppen mit speziellen Anforderungen und sind nicht für die Öffentlichkeit gedacht. Bevor Sie eine dieser Listen abonnieren, sollten Sie einige der technischen Listen abonniert haben, um mit den Umgangsformen vertraut zu sein.

Mailingliste	Zweck
freebsd-hubs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs)	Betrieb von FreeBSD-Spiegeln
freebsd-user-groups (http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups)	Koordination von Benutzergruppen
freebsd-vendors (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors)	Koordination von Händlern vor einem Release

Mailingliste	Zweck
freebsd-wip-status (http://lists.FreeBSD.org/mailman/listinfo/freebsd-wip-status)	Status von in Arbeit befindlichen FreeBSD-Tätigkeiten
freebsd-wireless (http://lists.FreeBSD.org/mailman/listinfo/freebsd-wireless)	Diskussionen zum 802.11-Stack sowie zur Entwicklung von Tools und Gerätetreibern
freebsd-www (http://lists.FreeBSD.org/mailman/listinfo/freebsd-www)	Betreuer von www.FreeBSD.org (http://www.FreeBSD.org/index.html)

Zusammenfassungen: Alle eben aufgezählten Listen sind auch in zusammengefasster Form (*digest*) erhältlich. In den Einstellungen Ihres Accounts legen Sie fest, in welcher Form Sie die Listen empfangen.

CVS & SVN Listen: Die folgenden Listen versenden die Log-Einträge zu Änderungen an verschiedenen Teilen des Quellbaums. Diese Listen sollen *nur gelesen* werden, schicken Sie bitte keine Nachrichten an eine der Listen.

Mailingliste	Teil des Quellbaums	Beschreibung
cvs-all (http://lists.FreeBSD.org/mailman/listinfo/cvs-all)	`/usr/(CVSROOT|doc|ports)`	Alle Änderungen im Quellbaum (Obermenge der anderen Commit-Listen)
cvs-doc (http://lists.FreeBSD.org/mailman/listinfo/cvs-doc)	`/usr/(doc|www)`	Änderungen in den `doc`- und `www` Bäumen
cvs-ports (http://lists.FreeBSD.org/mailman/listinfo/cvs-ports)	`/usr/ports`	Änderungen im `ports`-Baum
cvs-projects (http://lists.FreeBSD.org/mailman/listinfo/cvs-projects)	`/usr/projects`	Änderungen im `projects`-Baum
cvs-src (http://lists.FreeBSD.org/mailman/listinfo/cvs-src)	`/usr/src`	Änderungen im src-Baum (generiert aus den svn-zu-cvs Import-Commits
svn-src-all (http://lists.FreeBSD.org/mailman/listinfo/svn-src-all)	`/usr/src`	Änderungen im Subversion Repository (ausser für `user` und `projects`)

Mailingliste	Teil des Quellbaums	Beschreibung
svn-src-head (http://lists.FreeBSD.org/mailman/listinfo/svn-src-head)	`/usr/src`	Änderungen im “head” Zweig des Subversion Repository (der FreeBSD-CURRENT Zweig)
svn-src-projects (http://lists.FreeBSD.org/mailman/listinfo/svn-src-projects)	`/usr/projects`	Änderungen im `projects` Bereich des src Subversion Repository
svn-src-release (http://lists.FreeBSD.org/mailman/listinfo/svn-src-release)	`/usr/src`	Änderungen im `releases` Bereich des src Subversion Repository
svn-src-releng (http://lists.FreeBSD.org/mailman/listinfo/svn-src-releng)	`/usr/src`	Änderungen im `releng` Zweig des src Subversion Repository (der security / release engineering Zweige)
svn-src-stable (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable)	`/usr/src`	Änderungen an allen stable Zweigen des src Subversion Repository
svn-src-stable-6 (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-6)	`/usr/src`	Änderungen im `stable/6` Zweig des src Subversion Repository
svn-src-stable-7 (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-7)	`/usr/src`	Änderungen im `stable/7` Zweig des src Subversion Repository
svn-src stable-8 (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-8)	`/usr/src`	Änderungen im `stable/8` Zweig des src Subversion Repository
svn-src-stable-9 (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-9)	`/usr/src`	Änderungen im `stable/8` Zweig des src Subversion Repository
svn-src-stable-other (http://lists.FreeBSD.org/mailman/listinfo/svn-src-stable-other)	`/usr/src`	Änderungen an älteren `stable` Zweigen des src Subversion Repository
svn-src-svnadmin (http://lists.FreeBSD.org/mailman/listinfo/svn-src-svnadmin)	`/usr/src`	Änderungen an den administrativen Skripten, hooks, and anderen Daten zur Konfiguration des src Subversion Repository

Mailingliste	Teil des Quellbaums	Beschreibung
svn-src-user (http://lists.FreeBSD.org/mailman/listinfo/svn-src-user)	`/usr/src`	Änderungen am experimentellen `user` Bereich des src Subversion Repository
svn-src-vendor (http://lists.FreeBSD.org/mailman/listinfo/svn-src-vendor)	`/usr/src`	Änderungen am Herstellerbereich des src Subversion Repository

C.1.2. Mailinglisten abonnieren

Um eine Liste zu abonnieren, folgen Sie dem oben angegebenen Hyperlink der Liste oder Sie besuchen die Webseite http://lists.FreeBSD.org/mailman/listinfo und klicken dort auf die Liste, die Sie abonnieren wollen. Sie gelangen dann auf die Webseite der Liste, die weitere Anweisungen enthält.

Um eine Nachricht an eine Mailingliste zu schicken, schreiben Sie einfach eine E-Mail an <*Liste*`@FreeBSD.org`>. Die E-Mail wird dann an alle Mitglieder der Mailingliste verteilt.

Wenn Sie das Abonnement aufheben wollen, folgen Sie der URL, die am Ende jeder Mail der Liste angegeben ist. Sie können das Abonnement auch mit einer E-Mail an <*Liste*`-unsubscribe@FreeBSD.org`> aufheben.

Verwenden Sie bitte die technischen Listen ausschließlich für technische Diskussionen. Wenn Sie nur an wichtigen Ankündigungen interessiert sind, abonnieren Sie die Mailingliste FreeBSD announcements (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce), auf der nur wenige Nachrichten versendet werden.

C.1.3. Chartas der Mailinglisten

Alle FreeBSD-Mailinglisten besitzen Grundregeln, die von jedem beachtet werden müssen. Für die ersten beiden Male, in denen ein Absender gegen diese Regeln verstößt, erhält er jeweils eine Warnung vom FreeBSD-Postmaster <`postmaster@FreeBSD.org`>. Ein dritter Verstoß gegen die Regeln führt dazu, dass der Absender in allen FreeBSD-Mailinglisten gesperrt wird und weitere Nachrichten von ihm nicht mehr angenommen werden. Wir bedauern sehr, dass wir solche Maßnahmen ergreifen müssen, aber heutzutage ist das Internet eine recht rauhe Umgebung, in der immer weniger Leute Rücksicht aufeinander nehmen.

Die Regeln:

- Das Thema einer Nachricht soll der Charta der Liste, an die sie gesendet wird, entsprechen. Wenn Sie eine Nachricht an eine technische Liste schicken, sollte die Nachricht auch technische Inhalte haben. Fortwährendes Geschwätz oder Streit mindern den Wert der Liste für alle Mitglieder und wird nicht toleriert. Benutzen Sie FreeBSD chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat) für allgemeine Diskussionen über FreeBSD.
- Eine Nachricht sollte an nicht mehr als zwei Mailinglisten gesendet werden. Schicken Sie eine Nachricht nur dann an zwei Listen, wenn das wirklich notwendig ist. Viele Leute haben mehrere Mailinglisten abonniert und Nachrichten sollten nur zu ungewöhnlichen Kombinationen der Listen, wie “-stable” und “-scsi”, gesendet werden. Wenn Sie eine Nachricht erhalten, die im `Cc`-Feld mehrere Listen enthält, sollten Sie das Feld kürzen, bevor Sie eine Antwort darauf verschicken. *Unabhängig von dem ursprünglichen Verteiler sind Sie für Ihre eigenen Mehrfach-Sendungen selbst verantwortlich.*

- Persönliche Angriffe und Beschimpfungen sind in einer Diskussion nicht erlaubt. Dies gilt gleichermaßen für Benutzer wie Entwickler. Grobe Verletzungen der Netiquette, wie das Verschicken oder Zitieren von privater E-Mail ohne eine entsprechende Genehmigung, werden nicht gebilligt. Die Nachrichten werden aber nicht besonders auf Verletzungen der Netiquette untersucht. Es kann sein, dass eine Verletzung der Netiquette durchaus zu der Charta einer Liste passt, aber der Absender aufgrund der Verletzung eine Warnung erhält oder gesperrt wird.
- Werbung für Produkte oder Dienstleistungen, die nichts mit FreeBSD zu tun haben, sind verboten. Ist die Werbung als Spam verschickt worden, wird der Absender sofort gesperrt.

Chartas einzelner Listen:

freebsd-acpi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi)

Die Entwicklung von ACPI und Energieverwaltungsfunktionen.

freebsd-afs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs)

Andrew File System

Auf dieser Liste wird die Portierung des AFS von CMU/Transarc diskutiert.

freebsd-announce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce)

Wichtige Ereignisse und Meilensteine

Diese Liste ist für Personen, die nur an den wenigen Ankündigungen wichtiger Ereignisse interessiert sind. Die Ankündigungen betreffen Schnappschüsse und Releases, neue Merkmale von FreeBSD und die Suche nach freiwilligen Mitarbeitern. Auf der Liste herrscht wenig Verkehr und sie wird streng moderiert.

freebsd-arch (http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch)

Architektur und Design von FreeBSD

Auf dieser technischen Liste wird die FreeBSD-Architektur diskutiert. Beispiele für angemessene Themen sind:

- Wie das Bausystem zu verändern ist, damit verschiedene Läufe gleichzeitig möglich sind.
- Was am VFS geändert werden muss, damit Heidemann Schichten eingesetzt werden können.
- Wie die Schnittstelle der Gerätetreiber angepasst werden muss, damit derselbe Treiber auf verschiedenen Bussen und Architekturen eingesetzt werden kann.
- Wie ein Netzwerktreiber geschrieben wird.

freebsd-binup (http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup)

FreeBSD Binary Update Project

Auf dieser Liste wird das Design und die Implementierung von **binup** diskutiert. Weitere Themen sind Fehlerbehebungen, Fehlerberichte und Anfragen nach Neuerungen. Die CVS-Logmeldungen des Projekts werden ebenfalls auf diese Liste gesendet.

freebsd-bluetooth (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bluetooth)

Bluetooth unter FreeBSD

Diese Liste diskutiert Probleme der Verwendung von Bluetooth unter FreeBSD. Designprobleme, Implementierungsdetails, Patches, Fehler- und Statusberichte, Verbesserungsvorschläge sowie alle anderen mit Bluetooth zusammenhängenden Themen werden hier behandelt.

freebsd-bugbusters (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters)

Bearbeitung der Fehlerberichte

Auf dieser Liste wird die Bearbeitung der Fehlerberichte (PR, engl. *problem report*) koordiniert. Sie dient dem "Bugmeister" und allen Leuten, die ein Interesse an der Datenbank der Fehlerberichte haben, als Diskussionsforum. Auf dieser Liste werden keine spezifischen Fehler, Fehlerbehebungen oder PRs diskutiert.

freebsd-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs)

Fehlerberichte

Auf dieser Liste werden Fehlerberichte gesammelt. Fehlerberichte sollten immer mit send-pr(1) oder dem Web Formular (http://www.FreeBSD.org/send-pr.html) erstellt werden.

freebsd-chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat)

Nicht technische Themen, die die FreeBSD Gemeinschaft betreffen

Auf dieser Liste werden nicht-technische soziale Themen diskutiert, die nicht auf die anderen Listen passen. Hier kann diskutiert werden, ob Jordan wie ein Frettchen aus einem Zeichentrickfilm aussieht oder nicht, ob grundsätzlich in Großbuchstaben geschrieben werden soll, wer zuviel Kaffee trinkt, wo das beste Bier gebraut wird und wer Bier in seinem Keller braut. Gelegentlich können auf den technischen Listen wichtige Ereignisse wie Feste, Hochzeiten oder Geburten angekündigt werden, aber nachfolgende Nachrichten sollten auf die Liste FreeBSD chat (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat) gesendet werden.

freebsd-chromium (http://lists.FreeBSD.org/mailman/listinfo/freebsd-chromium)

Diskussionen zum Einsatz von Chromium unter FreeBSD

Auf dieser technischen Liste werden Fragen zur Entwicklung, zur Installation sowie zum Einsatz von Chromium unter FreeBSD diskutiert.

freebsd-core

FreeBSD Core Team

Dies ist eine interne Mailingliste des FreeBSD Core Teams. Wenn in einer wichtigen Angelegenheit, die FreeBSD betrifft, entschieden werden muss oder die Angelegenheit einer genauen Prüfung unterzogen werden muss, können Nachrichten an diese Liste gesendet werden.

freebsd-current (http://lists.FreeBSD.org/mailman/listinfo/freebsd-current)

Gebrauch von FreeBSD-CURRENT

Diese Mailingliste ist für die Benutzer von FreeBSD-CURRENT eingerichtet. Auf ihr finden sich Ankündigungen über Besonderheiten von -CURRENT, von denen Benutzer betroffen sind. Sie enthält weiterhin Anweisungen, wie man ein System auf -CURRENT hält. Jeder, der ein -CURRENT System besitzt, muss diese Liste lesen. Die Liste ist nur für technische Inhalte bestimmt.

freebsd-cvsweb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb)

FreeBSD CVSweb Project

Technische Diskussion über den Gebrauch, die Entwicklung und die Pflege von FreeBSD-CVSweb.

freebsd-desktop (http://lists.FreeBSD.org/mailman/listinfo/freebsd-desktop)

FreeBSD als Desktop verwenden und verbessern

Dies ist ein Forum für Diskussionen um FreeBSD auf dem Desktop. Es wird primär von Desktop-Portierern und Nutzern verwendet, um Probleme und Verbesserungen zu FreeBSDs Einsatz auf dem Desktop zu besprechen.

freebsd-doc (http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc)

Documentation Project

Auf dieser Mailingliste werden Themen und Projekte diskutiert, die im Zusammenhang mit der Erstellung der FreeBSD Dokumentation stehen. “The FreeBSD Documentation Project” besteht aus den Mitgliedern dieser Liste. Diese Liste steht jedem offen, Sie sind herzlich eingeladen teilzunehmen und mitzuhelfen.

freebsd-drivers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-drivers)

Gerätetreiber für FreeBSD schreiben

Ein Forum für technische Diskussionen über das Schreiben von Gerätetreibern für FreeBSD. Daher werden hier vor allem Fragen behandelt, die sich um das Schreiben von Treibern, die die APIs des Kernels nutzen, drehen.

freebsd-eclipse (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eclipse)

Für FreeBSD-Anwender, die die Eclipse IDE deren Werkzeuge, Anwendungen und Ports einsetzen

Das Ziel dieser Liste ist es, Unterstützung für all jene zu bieten, die mit der Installation, Verwendung, Entwicklung und Wartung der Eclipse-IDE sowie deren Werkzeugen und Anwendungen unter FreeBSD zu tun haben. Außerdem wird Hilfe bei der Portierung der IDE und deren Plugins auf FreeBSD geboten.

Zusätzlich soll diese Liste einen Informationsaustausch zwischen der Eclipse- und der FreeBSD-Gemeinde ermöglichen, von dem beide Seiten profitieren können.

Obwohl sich diese Liste auf die Anforderungen von Anwendern konzentriert, möchte sie auch Entwickler unterstützen, die an der Entwicklung von FreeBSD-spezifischen Anwendungen unter Nutzung des Eclipse-Frameworks arbeiten.

freebsd-embedded (http://lists.FreeBSD.org/mailman/listinfo/freebsd-embedded)

FreeBSD in eingebetteten Anwendungen einsetzen

Diese Liste diskutiert Themen im Zusammenhang mit dem Einsatz von ungewöhnlich kleinen und eingebettenen FreeBSD-Installationen. Auf dieser Liste werden ausschließlich technische Diskussionen geführt. Unter eingebetteten Systemen versteht diese Liste Systeme, bei denen es sich nicht um Desktopsysteme handelt, und die in der Regel nur einem einzigen Zweck dienen (im Gegensatz zu Desktopsystemen, die für die Bewältigung verschiedenster Aufgaben geeignet sind). In die Gruppe der eingebetteten Systeme gehören beispielsweise Telephone, Netzwerkgeräte wie Router, Switche oder PBX-Systeme, PDAs, Verkaufsautomaten und andere mehr.

freebsd-emulation (http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation)

Emulation anderer Systeme wie Linux, MS-DOS oder Windows

Hier werden technische Diskussionen zum Einsatz von Programmen, die für andere Betriebssysteme geschrieben wurden, geführt.

freebsd-eol (http://lists.FreeBSD.org/mailman/listinfo/freebsd-eol)

Support für FreeBSD-bezogene Software, die vom FreeBSD Project offiziell nicht mehr unterstützt wird.

Diese Liste ist für all jene interessant, die Unterstützung für vom FreeBSD Project offiziell nicht mehr (in Form von Security Advisories oder Patches) unterstützte Programme benötigen oder anbieten wollen.

freebsd-firewire (http://lists.FreeBSD.org/mailman/listinfo/freebsd-firewire)

FireWire (iLink, IEEE 1394)

Auf dieser Liste wird das Design und die Implementierung eines FireWire-Subsystems (auch IEEE 1394 oder iLink) für FreeBSD diskutiert. Relevante Themen sind die Standards, Busse und ihre Protokolle, sowie Adapter, Karten und Chipsätze. Des Weiteren die Architektur und der Quellcode, die nötig sind, diese Geräte zu unterstützen.

freebsd-fs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs)

Dateisysteme

Diskussionen über FreeBSD-Dateisysteme. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

freebsd-gecko (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gecko)

Angelegenheiten zur *Gecko Rendering Engine*

Dies ist ein Forum über **Gecko**-Anwendungen, die FreeBSD verwenden.

Die Diskussion dreht sich um die Portierung von Gecko-Anwendungen, deren Installation, die Entwicklung sowie deren Unterstützung innerhalb von FreeBSD.

freebsd-geom (http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom)

GEOM

Diskussion über GEOM und verwandte Implementierungen. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

freebsd-gnome (http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome)

GNOME

Diskussionen über die grafische Benutzeroberfläche **GNOME**. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

freebsd-ipfw (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw)

IP Firewall

Diskussionen über eine Neubearbeitung des IP-Firewall Quelltexts in FreeBSD. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

freebsd-ia64 (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64)

Portierung von FreeBSD auf die IA64-Plattform

Dies ist eine technische Liste für diejenigen, die FreeBSD auf die IA-64 Plattform von Intel portieren. Themen sind die Probleme bei der Portierung und deren Lösung. Interessierte, die der Diskussion folgen wollen, sind ebenfalls willkommen.

freebsd-isdn (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn)

ISDN Subsystem

Mailingliste für die Entwickler des ISDN Subsystems von FreeBSD.

freebsd-java (http://lists.FreeBSD.org/mailman/listinfo/freebsd-java)

Java Entwicklung

Mailingliste, auf der die Entwicklung von Java Anwendungen für FreeBSD sowie die Portierung und Pflege von JDKs diskutiert wird.

freebsd-jobs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs)

Stellenangebote und Stellengesuche

In diesem Forum können Sie Stellenangebote und Stellengesuche, die mit FreeBSD zu tun haben, aufgeben. Wenn Sie beispielsweise eine Beschäftigung im FreeBSD-Umfeld suchen oder eine freie Stelle haben, die mit FreeBSD zu tun hat, ist dies der richtige Ort. Diese Mailingliste ist *nicht* der Ort, um über allgemeine Beschäftigungsprobleme zu diskutieren; dazu gibt es anderswo geeignete Foren.

Beachten Sie bitte, dass diese Liste, wie die anderen `FreeBSD.org`-Listen, weltweit gelesen wird. Geben Sie daher bitte den Arbeitsort genau an. Geben Sie bitte auch an, ob Telearbeit möglich ist und ob Hilfen für einen Umzug angeboten werden.

Benutzen Sie in der E-Mail bitte nur offene Formate – vorzugsweise nur das Textformat. Andere Formate, wie PDF oder HTML, werden von den Lesern akzeptiert. Nicht offene Formate wie Microsoft Word (`.doc`) werden vom Server der Liste abgelehnt.

freebsd-hackers (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers)

Technische Diskussionen

Dies ist ein Forum für technische Diskussionen über FreeBSD. Leute, die aktiv an FreeBSD arbeiten, können hier Probleme und deren Lösungen diskutieren. Interessierte, die den Diskussionen folgen wollen, steht die Liste ebenfalls offen. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-hardware (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware)

Allgemeine Diskussionen über Hardware

Allgemeine Diskussionen über die Hardware, auf der FreeBSD läuft: Probleme und Ratschläge welche Hardware man kaufen sollte und welche nicht.

freebsd-hubs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs)

FreeBSD-Spiegel

Ankündigungen und Diskussionsforum für Leute, die FreeBSD-Spiegel betreiben.

freebsd-isp (http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp)

Themen für Internet Service Provider

Diese Liste ist für Internet Service Provider (ISP), die FreeBSD benutzen. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-mono (http://lists.FreeBSD.org/mailman/listinfo/freebsd-mono)

Mono und C# Anwendungen auf FreeBSD

Diese Liste beinhaltet Diskussionen über das Mono Entwicklungsframework auf FreeBSD. Dies ist eine technische Mailingliste. Es ist für Personen gedacht, die aktiv an der Portierung von Mono oder C# Anwendungen auf FreeBSD sind, um Probleme oder alternative Lösungen zu beratschlagen. Personen die der technischen Diskussion folgen möchten sind ebenso willkommen.

freebsd-kde (https://mail.kde.org/mailman/listinfo/kde-freebsd)

KDE

Diskussionen über **KDE** auf FreeBSD-Systemen. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

freebsd-performance (http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance)

Diskussionsforum mit dem Ziel, die Leistung von FreeBSD zu verbessern.

Auf dieser Liste diskutieren Hacker, Systemadministratoren und andere Interessierte die Leistung von FreeBSD. Zulässige Themen sind beispielsweise Systeme unter hoher Last, Systeme mit Leistungsproblemen oder Systeme, die Leistungsgrenzen von FreeBSD überwinden. Jeder, der mithelfen will, die Leistung von FreeBSD zu verbessern, sollte diese Liste abonnieren. Die Liste ist technisch anspruchsvoll und geeignet für erfahrene FreeBSD-Benutzer, Hacker oder Administratoren, die FreeBSD schnell, robust und skalierbar halten wollen. Auf der Liste werden Beiträge gesammelt oder Fragen nach ungelösten Problemen beantwortet. Sie ist kein Ersatz für das gründliche Studium der Dokumentation.

freebsd-pf (http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf)

Diskussionen und Fragen zu packet filter als Firewallsystem.

FreeBSD-spezische Diskussionen zur Benutzung von *packet filter* (pf) als Firewallsystem. Sowohl technische Diskussionen als auch Anwenderfragen sind auf dieser Liste willkommen. Fragen zum ALTQ QoS Framework können ebenfalls gestellt werden.

freebsd-platforms (http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms)

Portierung auf nicht-Intel Plattformen

Plattformübergreifende Themen und Vorschläge für die Portierung auf nicht-Intel Plattformen. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-ports (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports)

Diskussion über die Ports-Sammlung

Diskussionen über die FreeBSD-Ports-Sammlung und die Infrastruktur der Sammlung. Die Liste dient auch der allgemeinen Koordination der Dinge, die die Ports-Sammlung betreffen. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-ports-bugs (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs)

Diskussion über Fehler in den Ports

Diskussion über Fehler in der Ports-Sammlung (`/usr/ports`), neue Ports oder Änderungen an bestehenden Ports. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-proliant (http://lists.FreeBSD.org/mailman/listinfo/freebsd-proliant)

Technische Diskussionen zum Einsatz von FreeBSD auf der ProLiant-Serverplattform von HP

Diese Mailingliste bietet technische Diskussionen zum Einsatz von FreeBSD auf der ProLiant-Serverplattform von HP, darunter Fragen zu ProLiant-spezifischen Treibern, Konfigurationswerkzeugen sowie BIOS-Aktualisierungen. Daher ist sie die erste Anlaufstelle, um die Module hpasmd, hpasmcli, sowie hpacucli zu diskutieren.

freebsd-python (http://lists.FreeBSD.org/mailman/listinfo/freebsd-python)

Python unter FreeBSD

Diese technische Liste dient der Verbesserung der Python-Unterstützung unter FreeBSD. Sie wird von Personen gelesen, die an der Portierung von Python, von Python-Modulen Dritter und von **Zope** nach FreeBSD arbeiten. Personen, die diese technischen Diskussion verfolgen wollen, sind ebenfalls willkommen.

freebsd-questions (http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions)

Benutzerfragen

Auf dieser Mailingliste können Fragen zu FreeBSD gestellt werden. Fragen Sie bitte nicht nach Anleitungen, wenn Sie nicht sicher sind, dass Ihre Frage wirklich technischer Natur ist.

freebsd-ruby (http://lists.FreeBSD.org/mailman/listinfo/freebsd-ruby)

Ruby unter FreeBSD

Diese technische Liste dient der Verbesserung der Ruby-Unterstützung unter FreeBSD. Sie wird von Personen gelesen, die an der Portierung von Ruby, von Bibliotheken Dritter und Frameworks arbeiten. Personen, die diese technischen Diskussionen verfolgen wollen, sind ebenfalls willkommen.

freebsd-scsi (http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi)

SCSI Subsystem

Diese Mailingliste ist für die Entwickler des SCSI Subsystems von FreeBSD. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-security (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security)

Sicherheitsthemen

Sicherheitsthemen, die FreeBSD betreffen, wie DES, Kerberos, bekannte Sicherheitslöcher und Fehlerbehebungen. Stellen Sie bitte auf dieser Liste keine allgemeinen Fragen zum Thema Sicherheit. Willkommen sind allerdings Beiträge zur FAQ, das heißt eine Frage mit der passenden Antwort. Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-security-notifications (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications)

Ankündigungen zum Thema Sicherheit

Ankündigungen über Sicherheitsprobleme von FreeBSD und deren Behebungen. Diese Liste ist kein Diskussionsforum, benutzen Sie FreeBSD security (http://lists.FreeBSD.org/mailman/listinfo/freebsd-security), um Sicherheitsthemen zu diskutieren.

freebsd-small (http://lists.FreeBSD.org/mailman/listinfo/freebsd-small)

Gebrauch von FreeBSD in eingebetteten Systemen.

Diese Liste für ungewöhnlich kleine FreeBSD Installation oder den Einsatz von FreeBSD in eingebetteten Systemen gedacht. Auf dieser Liste finden nur technische Diskussionen statt.

> **Anmerkung:** Diese Liste wurde durch freebsd-embedded (http://lists.FreeBSD.org/mailman/listinfo/freebsd-embedded) ersetzt.

freebsd-stable (http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable)

Gebrauch von FreeBSD-STABLE.

Diese Mailingliste ist für die Benutzer von FreeBSD-STABLE eingerichtet. Auf ihr finden sich Ankündigungen über Besonderheiten von -STABLE, von denen Benutzer betroffen sind. Sie enthält weiterhin Anweisungen, wie man ein System auf -STABLE hält. Jeder, der ein -STABLE System besitzt, muss diese Liste lesen. Die Liste ist nur für technische Inhalte bestimmt.

freebsd-standards (http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards)

Konformität von FreeBSD mit den C99- und POSIX Standards

Dieses Forum ist für technische Diskussionen über die Konformität von FreeBSD mit den C99- und POSIX-Standards.

freebsd-toolchain (http://lists.FreeBSD.org/mailman/listinfo/freebsd-toolchain)

Wartung der FreeBSD-Toolchain

Auf dieser Mailingliste werden alle Themen rund um die FreeBSD-Toolchain diskutiert. Dazu gehören der Status von Clang und GCC, aber auch Fragen zu Programmen wie Assemblern, Linkern und Debuggern.

freebsd-usb (http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb)

USB-Unterstützung in FreeBSD.

Auf dieser Liste finden nur technische Diskussionen statt.

freebsd-user-groups (http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups)

Koordination von Benutzergruppen

Diese Liste ist für Koordinatoren lokaler Benutzergruppen und einem ausgesuchten Mitglied des Core Teams eingerichtet worden. Der Inhalt sollte Inhalte von Treffen und die Koordination von Projekten mehrerer Benutzergruppen beschränkt sein.

freebsd-vendors (http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors)

Koordination von Händlern

Koordination zwischen dem FreeBSD Project und Händlern, die Soft- und Hardware für FreeBSD verkaufen.

freebsd-virtualization (http://lists.FreeBSD.org/mailman/listinfo/freebsd-virtualization)

Diskussion über verschiedene Virtualisierungsverfahren, die von FreeBSD unterstützt werden

Eine Liste, auf der die verschiedenen Virtualisierungsverfahren, die von FreeBSD unterstützt werden, diskutiert werden. Auf der einen Seite liegt der Fokus auf der Implementierung der zugrundeliegenden Funktionalitäten,

ebenso wie das Hinzufügen neuer Eigenschaften. Auf der anderen Seite haben die Benutzer ein Forum, um Fragen bei Problemen zu stellen oder um ihre Anwendungsfälle zu besprechen.

freebsd-wip-status (http://lists.FreeBSD.org/mailman/listinfo/freebsd-wip-status)

Status von in Arbeit befindlichen FreeBSD-Tätigkeiten

Diese Mailingliste kann dazu verwendet werden, eigene Kreationen und deren Fortschritt von FreeBSD-verwandten Tätigkeiten anzukündigen. Die Nachrichten werden moderiert. Es wird vorgeschlagen, die Nachricht "An:" eine mehr themenverwandte FreeBSD-Liste zu senden und diese Liste nur in Blindkopie zu setzen. Auf diese Weise kann ihre in Arbeit befindliche Tätigkeit auch auf der themenverwandten Liste diskutiert werden, da auf dieser Liste keine Diskussionen erlaubt sind.

Sehen Sie sich das Archiv der Liste für passende Nachrichten an.

Redaktionelle Auszüge der Nachrichten an diese Liste werden eventuell alle paar Monate auf die FreeBSD Webseite als Teil der Statusberichte [1] gestellt. Weitere Beispiele und zurückliegende Berichte können Sie auch dort finden.

freebsd-wireless (http://lists.FreeBSD.org/mailman/listinfo/freebsd-wireless)

Diskussionen zum 802.11-Stack sowie zur Entwicklung von Tools und Gerätetreibern

Die Mailingliste freebsd-wireless diskutiert Themen rund um den 802.11-Stack (sys/net80211). Besprochen werden die Entwicklung von Tools und Gerätetreibern sowie auftretende Probleme, neue Funktionen sowie die Wartung der existierenden Werkzeuge und Treiber.

freebsd-xen (http://lists.FreeBSD.org/mailman/listinfo/freebsd-xen)

Diskussionen über die FreeBSD Portierung auf Xen - Implementierung und Verwendung

Eine Liste, die die FreeBSD Portierung auf Xen behandelt. Das erwartete Nachrichtenaufkommen ist klein genug, so dass es als Forum für sowohl technische Diskussionen über die Implementierung und Entwurfsdetails, als auch administrative Verteilaspekte ausgelegt ist.

freebsd-xfce (http://lists.FreeBSD.org/mailman/listinfo/freebsd-xfce)

XFCE

Eine Liste, auf der Fragen zum Einsatz von **XFCE** unter FreeBSD diskutiert werden. Es handelt sich um eine technische Mailingliste, die sich primär an Entwickler richtet, die aktiv an der Portierung von **XFCE** nach FreeBSD arbeiten. Aber auch Nutzer, die einfach nur die technischen Diskussionen verfolgen wollen, sind willkommen. Diskutiert werden vor allem bei der Portierung auftretende Probleme und mögliche Lösungswege.

freebsd-zope (http://lists.FreeBSD.org/mailman/listinfo/freebsd-zope)

Zope

Ein Forum für Diskussionen darüber, wie man die **Zope**-Umgebung auf FreeBSD portieren kann. Dies ist eine technische Mailingliste. Sie ist für Leute gedacht, die aktiv an der Portierung von **Zope** auf FreeBSD arbeiten, um aufkommende Probleme oder alternative Lösungsansätze zu besprechen. Personen, die der technischen Diskussion folgen möchten, sind ebenfalls willkommen.

C.1.4. Filter der Mailinglisten

Um die Verbreitung von Spam, Viren und anderen nicht erwünschten E-Mails zu verhindern, werden auf den FreeBSD-Mailinglisten Filter eingesetzt. Dieser Abschnitt beschreibt nur einen Teil der zum Schutz der Listen eingesetzten Filter.

Auf den Mailinglisten sind nur die unten aufgeführten Anhänge erlaubt. Anhänge mit einem anderen MIME-Typ werden entfernt, bevor eine E-Mail an eine Liste verteilt wird.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch

> **Anmerkung:** Einige Mailinglisten erlauben vielleicht Anhänge mit anderem MIME-Typ. Für die meisten Mailinglisten sollte die obige Aufzählung aber richtig sein.

Wenn eine E-Mail sowohl aus einer HTML-Version wie auch aus einer Text-Version besteht, wird die HTML-Version entfernt. Wenn eine E-Mail nur im HTML-Format versendet wurde, wird sie in reinen Text umgewandelt.

C.2. Usenet-News

Neben den Gruppen, die sich ausschließlich mit BSD beschäftigen, gibt es viele weitere in denen über FreeBSD diskutiert wird, oder die für FreeBSD-Benutzer wichtig sind. Warren Toomey <`wkt@cs.adfa.edu.au`> stellte großzügig suchbare Archive (http://minnie.tuhs.org/BSD-info/bsdnews_search.html) einiger dieser Gruppen bereit.

C.2.1. BSD spezifische Gruppen

- comp.unix.bsd.freebsd.announce (news:comp.unix.bsd.freebsd.announce)
- comp.unix.bsd.freebsd.misc (news:comp.unix.bsd.freebsd.misc)
- de.comp.os.unix.bsd (news:de.comp.os.unix.bsd) (deutsch)

- fr.comp.os.bsd (news:fr.comp.os.bsd) (französisch)
- it.comp.os.bsd (news:it.comp.os.bsd) (italienisch)
- tw.bbs.comp.386bsd (news:tw.bbs.comp.386bsd) (Traditionelles Chinesisch)

C.2.2. Weitere UNIX Gruppen

- comp.unix (news:comp.unix)
- comp.unix.questions (news:comp.unix.questions)
- comp.unix.admin (news:comp.unix.admin)
- comp.unix.programmer (news:comp.unix.programmer)
- comp.unix.shell (news:comp.unix.shell)
- comp.unix.user-friendly (news:comp.unix.user-friendly)
- comp.security.unix (news:comp.security.unix)
- comp.sources.unix (news:comp.sources.unix)
- comp.unix.advocacy (news:comp.unix.advocacy)
- comp.unix.misc (news:comp.unix.misc)
- comp.bugs.4bsd (news:comp.bugs.4bsd)
- comp.bugs.4bsd.ucb-fixes (news:comp.bugs.4bsd.ucb-fixes)
- comp.unix.bsd (news:comp.unix.bsd)

C.2.3. X Window System

- comp.windows.x.i386unix (news:comp.windows.x.i386unix)
- comp.windows.x (news:comp.windows.x)
- comp.windows.x.apps (news:comp.windows.x.apps)
- comp.windows.x.announce (news:comp.windows.x.announce)
- comp.windows.x.intrinsics (news:comp.windows.x.intrinsics)
- comp.windows.x.motif (news:comp.windows.x.motif)
- comp.windows.x.pex (news:comp.windows.x.pex)
- comp.emulators.ms-windows.wine (news:comp.emulators.ms-windows.wine)

C.3. World Wide Web Server

C.3.1. Foren, Blogs und soziale Netzwerke

- Die FreeBSD Foren (http://forums.freebsd.org/) dienen als webbasiertes Diskussionsforum für Fragen und technische Diskussionen zu FreeBSD.
- Planet FreeBSD (http://planet.freebsdish.org/) bietet einen gesammelten Feed aus dutzenden von Blogs, die von den FreeBSD Entwicklern geschrieben werden. Viele Entwickler nutzen dies, um schnell Aufzeichnungen darüber zu veröffentlichen, woran sie gerade arbeiten, welche neuen Erweiterungen es gibt und andere Arbeiten, die gerade im Gange sind.
- Der BSDConferences YouTube-Kanal (http://www.youtube.com/bsdconferences) beinhaltet eine Sammlung von qualitativ hochwertigen Videos von BSD Konferenzen aus der ganzen Welt. Dies ist eine ausgezeichnete Art und Weise, den Entwicklern beim Präsentieren von neuen Arbeiten an FreeBSD zuzuschauen.

C.3.2. Official Mirrors

Hauptserver, Armenien, Australien, Dänemark, Deutschland, Finnland, Frankreich, Großbritannien, Hong Kong, Irland, Island, Japan, Kanada, Lettland, Litauen, Niederlande, Norwegen, Österreich, Russland, Schweden, Schweiz, Slowakische Republik, Slowenien, Spanien, Südafrika, Taiwan, Tschechische Republik, Türkei, USA.

(aktualisiert am: UTC)

- Hauptserver
  - http://www.FreeBSD.org/
- Armenien
  - http://www1.am.FreeBSD.org/ (IPv6)
- Australien
  - http://www.au.FreeBSD.org/
  - http://www2.au.FreeBSD.org/
- Dänemark
  - http://www.dk.FreeBSD.org/ (IPv6)

- Deutschland
  - http://www.de.FreeBSD.org/
- Finnland
  - http://www.fi.FreeBSD.org/
- Frankreich
  - http://www1.fr.FreeBSD.org/
- Großbritannien
  - http://www1.uk.FreeBSD.org/
  - http://www3.uk.FreeBSD.org/
- Hong Kong
  - http://www.hk.FreeBSD.org/
- Irland
  - http://www.ie.FreeBSD.org/
- Island
  - http://www.is.FreeBSD.org/
- Japan
  - http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)
- Kanada

- http://www.ca.FreeBSD.org/
- http://www2.ca.FreeBSD.org/

- Lettland
  - http://www.lv.FreeBSD.org/

- Litauen
  - http://www.lt.FreeBSD.org/

- Niederlande
  - http://www.nl.FreeBSD.org/

- Norwegen
  - http://www.no.FreeBSD.org/

- Österreich
  - http://www.at.FreeBSD.org/ (IPv6)

- Russland
  - http://www.ru.FreeBSD.org/
  - http://www2.ru.FreeBSD.org/

- Schweden
  - http://www.se.FreeBSD.org/
  - http://www2.se.FreeBSD.org/

- Schweiz

- http://www.ch.FreeBSD.org/ (IPv6)
- http://www2.ch.FreeBSD.org/ (IPv6)

- Slowakische Republik
  - http://www.sk.FreeBSD.org/

- Slowenien
  - http://www.si.FreeBSD.org/

- Spanien
  - http://www.es.FreeBSD.org/
  - http://www2.es.FreeBSD.org/

- Südafrika
  - http://www.za.FreeBSD.org/
  - http://www2.za.FreeBSD.org/

- Taiwan
  - http://www.tw.FreeBSD.org/ (IPv6)
  - http://www2.tw.FreeBSD.org/
  - http://www4.tw.FreeBSD.org/
  - http://www5.tw.FreeBSD.org/ (IPv6)

- Tschechische Republik
  - http://www.cz.FreeBSD.org/ (IPv6)

- Türkei
  - http://www.tr.FreeBSD.org/

- USA
  - http://www5.us.FreeBSD.org/ (IPv6)

C.4. E-Mail Adressen

Die folgenden Benutzergruppen stellen ihren Mitgliedern für die Arbeit an FreeBSD E-Mail-Adressen zur Verfügung. Der aufgeführte Administrator behält sich das Recht vor, die Adresse zu sperren, wenn sie missbraucht wird.

Domain	Angebot	Benutzergruppe	Administrator
ukug.uk.FreeBSD.org	nur zum Weiterleiten	<ukfreebsd@uk.FreeBSD.org>	Lee Johnston <lee@uk.FreeBSD.org>

Fußnoten

1. http://www.freebsd.org/news/status/

Anhang D. PGP Schlüssel

Verwenden Sie die nachstehenden Schlüssel, wenn Sie eine Signatur überprüfen oder eine verschlüsselte E-Mail an einen Ansprechpartner oder einen Entwickler schicken wollen. Den vollständigen Schlüsselring der Benutzer von `FreeBSD.org` finden Sie unter http://www.FreeBSD.org/doc/pgpkeyring.txt.

D.1. Ansprechpartner

D.1.1. Security Officer Team <security-officer@FreeBSD.org>

```
pub  1024D/CA6CDFB2 2002-08-27 FreeBSD Security Officer <security-officer@FreeBSD.org>
     Key fingerprint = C374 0FC5 69A6 FBB1 4AED  B131 15D6 8804 CA6C DFB2
sub  2048g/A3071809 2002-08-27
```

D.1.2. Core Team Secretary <core-secretary@FreeBSD.org>

```
pub   2048R/2CA49776 2012-07-23
      Key fingerprint = 89F6 C031 B4E3 D472 E4CE  8372 4D58 FDCD 2CA4 9776
uid                  FreeBSD Core Team Secretary <core-secretary@freebsd.org>
sub   2048R/BBAD1C98 2012-07-23
```

D.1.3. Ports Management Team Secretary <portmgr-secretary@FreeBSD.org>

```
pub   2048R/BBC4D7D5 2012-07-24
      Key fingerprint = FB37 45C8 6F15 E8ED AC81  32FC D829 4EC3 BBC4 D7D5
uid                  FreeBSD Ports Management Team Secretary <portmgr-secretary@FreeBSD.org>
sub   2048R/5F65CFE7 2012-07-24
```

D.2. Mitglieder des Core Teams

D.2.1. Thomas Abthorpe <tabthorpe@FreeBSD.org>

```
pub   2048R/A473C990 2010-05-28
      Key fingerprint = D883 2D7C EB78 944A 69FC  36A6 D937 1097 A473 C990
uid                  Thomas Abthorpe (FreeBSD Committer) <tabthorpe@FreeBSD.org>
uid                  Thomas Abthorpe <tabthorpe@abthorpe.org>
uid                  Thomas Abthorpe <tabthorpe@goodking.ca>
uid                  Thomas Abthorpe <tabthorpe@goodking.org>
uid                  Thomas Abthorpe <thomas@goodking.ca>
sub   2048R/8CA60EE0 2010-05-28
```

D.2.2. Gavin Atkinson <gavin@FreeBSD.org>

```
pub   1024D/A093262B 2005-02-18
      Key fingerprint = 313A A79F 697D 3A5C 216A  EDF5 935D EF44 A093 262B
uid                  Gavin Atkinson (FreeBSD key) <gavin@FreeBSD.org>
uid                  Gavin Atkinson (Work e-mail) <ga9@york.ac.uk>
uid                  Gavin Atkinson <gavin@16squared.co.uk>
uid                  Gavin Atkinson <gavin.atkinson@ury.york.ac.uk>
uid                  Gavin Atkinson (Work e-mail) <gavin.atkinson@york.ac.uk>
sub   2048g/58F40B3D 2005-02-18
```

D.2.3. John Baldwin <jhb@FreeBSD.org>

```
pub  1024R/C10A874D 1999-01-13 John Baldwin <jbaldwin@weather.com>
     Key fingerprint = 43 33 1D 37 72 B1 EF 5B  9B 5F 39 F8 BD C1 7C B5
uid                           John Baldwin <john@baldwin.cx>
uid                           John Baldwin <jhb@FreeBSD.org>
uid                           John Baldwin <jobaldwi@vt.edu>
```

D.2.4. Konstantin Belousov <kib@FreeBSD.org>

```
pub   4096R/C1BCAD41 2012-11-17
      Key fingerprint = 7DE0 3388 64AC 53C3 7B88  3A79 90C2 B92B C1BC AD41
uid                  Konstantin Belousov <kib@FreeBSD.org>
uid                  Konstantin Belousov <kostikbel@gmail.com>
uid                  Konstantin Belousov <kib@kib.kiev.ua>
sub   4096R/3BBC8F64 2012-11-17
```

D.2.5. David Chisnall <theraven@FreeBSD.org>

```
pub   4096R/65C4F55D 2012-11-28
      Key fingerprint = 3E8F 5E9F 7586 F090 AC2C  58C2 BA06 FF14 65C4 F55D
uid                  David Chisnall <theraven@FreeBSD.org>
sub   4096R/04B2A21D 2012-11-28
```

D.2.6. Hiroki Sato <hrs@FreeBSD.org>

```
pub   1024D/2793CF2D 2001-06-12
      Key fingerprint = BDB3 443F A5DD B3D0 A530  FFD7 4F2C D3D8 2793 CF2D
uid                  Hiroki Sato <hrs@allbsd.org>
uid                  Hiroki Sato <hrs@eos.ocn.ne.jp>
uid                  Hiroki Sato <hrs@ring.gr.jp>
uid                  Hiroki Sato <hrs@FreeBSD.org>
uid                  Hiroki Sato <hrs@jp.FreeBSD.org>
uid                  Hiroki Sato <hrs@vlsi.ee.noda.tus.ac.jp>
uid                  Hiroki Sato <hrs@jp.NetBSD.org>
```

```
uid                  Hiroki Sato <hrs@NetBSD.org>
uid                  Hiroki Sato <hrs@ec.ss.titech.ac.jp>
uid                  Hiroki Sato <hrs@ieee.org>
uid                  Hiroki Sato <hrs@acm.org>
uid                  Hiroki Sato <hrs@bsdconsulting.co.jp>
uid                  Hiroki Sato <hrs@bsdresearch.org>
uid                  Hiroki Sato <hrs@ec.ce.titech.ac.jp>
sub   1024g/8CD251FF 2001-06-12
```

D.2.7. Peter Wemm <peter@FreeBSD.org>

```
pub  1024D/7277717F 2003-12-14 Peter Wemm <peter@wemm.org>
     Key fingerprint = 622B 2282 E92B 3BAB 57D1  A417 1512 AE52 7277 717F
uid                            Peter Wemm <peter@FreeBSD.ORG>
sub  1024g/8B40D9D1 2003-12-14
pub  1024R/D89CE319 1995-04-02 Peter Wemm <peter@netplex.com.au>
     Key fingerprint = 47 05 04 CA 4C EE F8 93  F6 DB 02 92 6D F5 58 8A
uid                            Peter Wemm <peter@perth.dialix.oz.au>
uid                            Peter Wemm <peter@haywire.dialix.com>
```

D.2.8. Martin Wilke <miwi@FreeBSD.org>

```
pub   1024D/B1E6FCE9 2009-01-31
      Key fingerprint = C022 7D60 F598 8188 2635  0F6E 74B2 4884 B1E6 FCE9
uid                  Martin Wilke <miwi@FreeBSD.org>
sub   4096g/096DA69D 2009-01-31
```

D.3. Entwickler

D.3.1. Ariff Abdullah <ariff@FreeBSD.org>

```
pub   1024D/C5304CDA 2005-10-01
      Key fingerprint = 5C7C 6BF4 8293 DE76 27D9  FD57 96BF 9D78 C530 4CDA
uid                  Ariff Abdullah <skywizard@MyBSD.org.my>
uid                  Ariff Abdullah <ariff@MyBSD.org.my>
uid                  Ariff Abdullah <ariff@FreeBSD.org>
sub   2048g/8958C1D3 2005-10-01
```

D.3.2. Thomas Abthorpe <tabthorpe@FreeBSD.org>

```
pub   2048R/A473C990 2010-05-28
      Key fingerprint = D883 2D7C EB78 944A 69FC  36A6 D937 1097 A473 C990
uid                  Thomas Abthorpe (FreeBSD Committer) <tabthorpe@FreeBSD.org>
uid                  Thomas Abthorpe <tabthorpe@abthorpe.org>
```

```
uid                  Thomas Abthorpe <tabthorpe@goodking.ca>
uid                  Thomas Abthorpe <tabthorpe@goodking.org>
uid                  Thomas Abthorpe <thomas@goodking.ca>
sub   2048R/8CA60EE0 2010-05-28
```

D.3.3. Eitan Adler <eadler@FreeBSD.org>

```
pub   4096R/8FC8196C 2011-02-11
      Key fingerprint = 49C7 29DF E09C 0FC7 A1C4  6ECB A338 A6FC 8FC8 196C
uid                  Eitan Adler <lists@eitanadler.com>
sub   4096R/18763D51 2011-02-11
sub   4096R/DAB9CF9B 2011-02-11
```

D.3.4. Shaun Amott <shaun@FreeBSD.org>

```
pub   1024D/6B387A9A 2001-03-19
      Key fingerprint = B506 E6C7 74A1 CC11 9A23  5C13 9268 5D08 6B38 7A9A
uid                  Shaun Amott <shaun@inerd.com>
uid                  Shaun Amott <shaun@FreeBSD.org>
sub   2048g/26FA8703 2001-03-19
sub   2048R/7FFF5151 2005-11-06
sub   2048R/27C54137 2005-11-06
```

D.3.5. Henrik Brix Andersen <brix@FreeBSD.org>

```
pub   1024D/54E278F8 2003-04-09
      Key fingerprint = 7B63 EF32 7831 A704 220D  7E61 BFE4 387E 54E2 78F8
uid                  Henrik Brix Andersen <henrik@brixandersen.dk>
uid                  Henrik Brix Andersen <brix@FreeBSD.org>
uid                  Henrik Brix Andersen <hbn@terma.com>
uid                  Henrik Brix Andersen <brix@osaa.dk>
sub   1024g/3B13C209 2003-04-09
```

D.3.6. Matthias Andree <mandree@FreeBSD.org>

```
pub   1024D/052E7D95 2003-08-28
      Key fingerprint = FDD0 0C43 6E33 07E1 0758  C6A8 BE61 8339 052E 7D95
uid                  Matthias Andree <mandree@freebsd.org>
uid                  Matthias Andree <matthias.andree@gmx.de>
sub   1536g/E65A83DA 2003-08-28
```

D.3.7. Will Andrews <will@FreeBSD.org>

```
pub  1024D/F81672C5 2000-05-22 Will Andrews (Key for official matters) <will@FreeBSD.org>
     Key fingerprint = 661F BBF7 9F5D 3D02 C862  5F6C 178E E274 F816 72C5
uid                            Will Andrews <will@physics.purdue.edu>
uid                            Will Andrews <will@puck.firepipe.net>
uid                            Will Andrews <will@c-60.org>
uid                            Will Andrews <will@csociety.org>
uid                            Will Andrews <will@csociety.ecn.purdue.edu>
uid                            Will Andrews <will@telperion.openpackages.org>
sub  1024g/55472804 2000-05-22
```

D.3.8. Dimitry Andric <dim@FreeBSD.org>

```
pub   1024D/2E2096A3 1997-11-17
      Key fingerprint = 7AB4 62D2 CE35 FC6D 4239  4FCD B05E A30A 2E20 96A3
uid                  Dimitry Andric <dimitry@andric.com>
uid                  Dimitry Andric <dim@xs4all.nl>
uid                  Dimitry Andric <dimitry.andric@tomtom.com>
uid                  [jpeg image of size 5132]
uid                  Dimitry Andric <dim@nah6.com>
uid                  Dimitry Andric <dim@FreeBSD.org>
sub   4096g/6852A5C5 1997-11-17
```

D.3.9. Eric Anholt <anholt@FreeBSD.org>

```
pub   1024D/6CF0EAF7 2003-09-08
      Key fingerprint = 76FE 2475 820B B75F DCA4  0F3E 1D47 6F60 6CF0 EAF7
uid                  Eric Anholt <eta@lclark.edu>
uid                  Eric Anholt <anholt@FreeBSD.org>
sub   1024g/80B404C1 2003-09-08
```

D.3.10. Marcus von Appen <mva@FreeBSD.org>

```
pub   1024D/B267A647 2009-02-14
      Key fingerprint = C7CC 1853 D8C5 E580 7795  B654 8BAF 3F12 B267 A647
uid                  Marcus von Appen <freebsd@sysfault.org>
uid                  Marcus von Appen <mva@freebsd.org>
sub   2048g/D34A3BAF 2009-02-14
```

D.3.11. Marcelo Araujo <araujo@FreeBSD.org>

```
pub   1024D/53E4CFA8 2007-04-27
      Key fingerprint = 9D6A 2339 925C 4F61 ED88  ED8B A2FC 4977 53E4 CFA8
uid                  Marcelo Araujo (Ports Committer) <araujo@FreeBSD.org>
sub   2048g/63CC012D 2007-04-27
```

D.3.12. Mathieu Arnold <mat@FreeBSD.org>

```
pub   1024D/FE6D850F 2005-04-25
      Key fingerprint = 2771 11F4 0A7E 73F9 ADDD  A542 26A4 7C6A FE6D 850F
uid                  Mathieu Arnold <mat@FreeBSD.org>
uid                  Mathieu Arnold <mat@mat.cc>
uid                  Mathieu Arnold <mat@cpan.org>
uid                  Mathieu Arnold <m@absolight.fr>
uid                  Mathieu Arnold <m@absolight.net>
uid                  Mathieu Arnold <mat@club-internet.fr>
uid                  Mathieu Arnold <marnold@april.org>
uid                  Mathieu Arnold <paypal@mat.cc>
sub   2048g/EAD18BD9 2005-04-25
```

D.3.13. Takuya ASADA <syuu@FreeBSD.org>

```
pub   2048R/43788F78 2012-11-21
      Key fingerprint = 31CE 242E 6F4F F24F EEF4  D9BB 0890 2C5F 4378 8F78
uid                  Takuya ASADA <syuu@freebsd.org>
sub   2048R/A87B0906 2012-11-21
```

D.3.14. Satoshi Asami <asami@FreeBSD.org>

```
pub  1024R/1E08D889 1997-07-23 Satoshi Asami <asami@cs.berkeley.edu>
     Key fingerprint = EB 3C 68 9E FB 6C EB 3F  DB 2E 0F 10 8F CE 79 CA
uid                           Satoshi Asami <asami@FreeBSD.ORG>
```

D.3.15. Gavin Atkinson <gavin@FreeBSD.org>

```
pub   1024D/A093262B 2005-02-18
      Key fingerprint = 313A A79F 697D 3A5C 216A  EDF5 935D EF44 A093 262B
uid                  Gavin Atkinson (FreeBSD key) <gavin@FreeBSD.org>
uid                  Gavin Atkinson (Work e-mail) <ga9@york.ac.uk>
uid                  Gavin Atkinson <gavin@16squared.co.uk>
uid                  Gavin Atkinson <gavin.atkinson@ury.york.ac.uk>
uid                  Gavin Atkinson (Work e-mail) <gavin.atkinson@york.ac.uk>
sub   2048g/58F40B3D 2005-02-18
```

D.3.16. Joseph S. Atkinson <jsa@FreeBSD.org>

```
pub   2048R/21AA7B06 2010-07-14
      Key fingerprint = 5B38 63B0 9CCA 12BE 3919  9412 CC9D FC84 21AA 7B06
uid                  Joseph S. Atkinson <jsa@FreeBSD.org>
uid                  Joseph S. Atkinson <jsa.bsd@gmail.com>
uid                  Joseph S. Atkinson <jsa@wickedmachine.net>
sub   2048R/5601C3E3 2010-07-14
```

D.3.17. Philippe Audeoud <jadawin@FreeBSD.org>

```
pub   1024D/C835D40E 2005-04-13
      Key fingerprint = D090 8C96 3612 15C9 4E3E  7A4A E498 FC2B C835 D40E
uid                  Philippe Audeoud <jadawin@tuxaco.net>
uid                  Philippe Audeoud <philippe@tuxaco.net>
uid                  Philippe Audeoud <philippe.audeoud@sitadelle.com>
uid                  Philippe Audeoud <jadawin@freebsd.org>
sub   2048g/EF8EA329 2005-04-13
```

D.3.18. Timur I. Bakeyev <timur@FreeBSD.org>

```
pub   1024D/60BA1F47 2002-04-27
      Key fingerprint = 84BF EAD1 607D 362F 210E  69B3 0BF0 6412 60BA 1F47
uid                  Timur I. Bakeyev (BaT) <timur@bat.ru>
uid                  Timur I. Bakeyev <timur@gnu.org>
uid                  Timur I. Bakeyev (BaT) <bat@cpan.org>
uid                  Timur I. Bakeyev (BaT) <timur@FreeBSD.org>
uid                  Timur I. Bakeyev (BaT) <timur@gnome.org>
uid                  Timur I. Bakeyev <timur@gnome.org>
sub   2048g/8A5B0042 2002-04-27
```

D.3.19. Glen Barber <gjb@FreeBSD.org>

```
pub   2048R/A0B946A3 2010-08-03 [expires: 2017-04-25]
      Key fingerprint = 78B3 42BA 26C7 B2AC 681E  A7BE 524F 0C37 A0B9 46A3
uid                  Glen Barber <gjb@FreeBSD.org>
uid                  Glen Barber <glen.j.barber@gmail.com>
uid                  Glen Barber <gjb@glenbarber.us>
sub   2048R/6C0527E5 2010-08-03
```

D.3.20. Nick Barkas <snb@FreeBSD.org>

```
pub   2048R/DDADB9DC 2010-07-27
      Key fingerprint = B678 6ECB 303D F580 A050  098F BDFF 4F3D DDAD B9DC
uid                  S. Nicholas Barkas <snb@freebsd.org>
sub   2048R/36E181FB 2010-07-27
sub   2048R/BDA4BED3 2010-07-29
sub   2048R/782A8737 2010-07-29
```

D.3.21. Simon Barner <barner@FreeBSD.org>

```
pub   1024D/EBADA82A 2000-11-10
      Key fingerprint = 67D1 3562 9A2F 3177 E46A  35ED 0A49 FEFD EBAD A82A
uid                  Simon Barner <barner@FreeBSD.org>
uid                  Simon Barner <barner@in.tum.de>
```

```
uid                  Simon Barner <barner@informatik.tu-muenchen.de>
uid                  Simon Barner <barner@gmx.de>
sub   2048g/F63052DE 2000-11-10
```

D.3.22. Artem Belevich <art@FreeBSD.org>

```
pub   2048R/9ED4C836 2011-03-28
      Key fingerprint = 7400 D541 07ED 3DF3 3E97  F2D5 8BDF 101C 9ED4 C836
uid                  Artem Belevich <artemb@gmail.com>
uid                  Artem Belevich <art@freebsd.org>
sub   2048R/55B0E4EB 2011-03-28
```

D.3.23. Anton Berezin <tobez@FreeBSD.org>

```
pub  1024D/7A7BA3C0 2000-05-25 Anton Berezin <tobez@catpipe.net>
     Key fingerprint = CDD8 560C 174B D8E5 0323  83CE 22CA 584C 7A7B A3C0
uid                            Anton Berezin <tobez@tobez.org>
uid                            Anton Berezin <tobez@FreeBSD.org>
sub  1024g/ADC71E87 2000-05-25
```

D.3.24. Damien Bergamini <damien@FreeBSD.org>

```
pub   2048R/D129F093 2005-03-02
      Key fingerprint = D3AB 28C3 1A4A E219 3145  54FE 220A 7486 D129 F093
uid                  Damien Bergamini <damien.bergamini@free.fr>
uid                  Damien Bergamini <damien@FreeBSD.org>
sub   2048R/9FBA73A4 2005-03-02
```

D.3.25. Tim Bishop <tdb@FreeBSD.org>

```
pub   1024D/5AE7D984 2000-10-07
      Key fingerprint = 1453 086E 9376 1A50 ECF6  AE05 7DCE D659 5AE7 D984
uid                  Tim Bishop <tim@bishnet.net>
uid                  Tim Bishop <T.D.Bishop@kent.ac.uk>
uid                  Tim Bishop <tdb@i-scream.org>
uid                  Tim Bishop <tdb@FreeBSD.org>
sub   4096g/7F886031 2000-10-07
```

D.3.26. Grzegorz Blach <gblach@FreeBSD.org>

```
pub   2048R/D25B0682 2012-11-03 [expires: 2014-11-03]
      Key fingerprint = 225B 941C A886 05C6 1C87  9C03 DE72 593D D25B 0682
uid                  Grzegorz Blach <gblach@FreeBSD.org>
sub   2048R/5DE28719 2012-11-03 [expires: 2014-11-03]
```

D.3.27. Martin Blapp <mbr@FreeBSD.org>

```
pub  1024D/D300551E 2001-12-20 Martin Blapp <mb@imp.ch>
     Key fingerprint = B434 53FC C87C FE7B 0A18  B84C 8686 EF22 D300 551E
sub  1024g/998281C8 2001-12-20
```

D.3.28. Warren Block <wblock@FreeBSD.org>

```
pub   2048R/A1F360A3 2011-09-14
      Key fingerprint = 3A44 4DEC B304 5191 8A41  C317 5117 4BB6 A1F3 60A3
uid                  Warren Block <wblock@FreeBSD.org>
uid                  Warren Block <wblock@wonkity.com>
sub   2048R/51F483F3 2011-09-14
```

D.3.29. Vitaly Bogdanov <bvs@FreeBSD.org>

```
pub  1024D/B32017F7 2005-10-02 Vitaly Bogdanov <gad@gad.glazov.net>
     Key fingerprint = 402E B8E4 53CB 22FF BE62  AE35 A0BF B077 B320 17F7
uid                            Vitaly Bogdanov <bvs@freebsd.org>
sub  1024g/0E88C62E 2005-10-02
```

D.3.30. Roman Bogorodskiy <novel@FreeBSD.org>

```
pub   2048R/08C2226A 2010-12-03
      Key fingerprint = 8BA4 DF2A D14F 99B6 37E0  0070 C96D 5FFE 08C2 226A
uid                  Roman Bogorodskiy <bogorodskiy@gmail.com>
uid                  Roman Bogorodskiy <novel@FreeBSD.org>
uid                  Roman Bogorodskiy <rbogorodskiy@apache.org>
uid                  Roman Bogorodskiy <rbogorodskiy@griddynamics.com>
sub   2048R/EC4ED237 2010-12-03
```

D.3.31. Renato Botelho <garga@FreeBSD.org>

```
pub   4096R/9F625790 2012-11-28 [expires: 2017-11-27]
      Key fingerprint = E3DA 9B2A 6160 99CB 4B31  7641 F1F0 E7A1 9F62 5790
uid                  Renato Botelho (FreeBSD) <garga@FreeBSD.org>
uid                  Renato Botelho (Personal) <rbgarga@gmail.com>
uid                  Renato Botelho (FreeBSD) <garga.bsd@gmail.com>
sub   4096R/473CC82A 2012-11-28 [expires: 2017-11-27]
```

D.3.32. Alexander Botero-Lowry <alexbl@FreeBSD.org>

```
pub   1024D/12A95A7B 2006-09-13
      Key fingerprint = D0C3 47F8 AE87 C829 0613  3586 24DF F52B 12A9 5A7B
uid                  Alexander Botero-Lowry <alexbl@FreeBSD.org>
sub   2048g/CA287923 2006-09-13
```

D.3.33. Sofian Brabez <sbz@FreeBSD.org>

```
pub   1024D/2487E57E 2011-03-15 [expires: 2016-03-14]
      Key fingerprint = 05BA DC7E F628 DE3F B241  BFBB 7363 51F4 2487 E57E
uid                  Sofian Brabez <sbrabez@gmail.com>
uid                  Sofian Brabez <sbz@FreeBSD.org>
uid                  Sofian Brabez <sbz@6dev.net>
```

D.3.34. Edson Brandi <ebrandi@FreeBSD.org>

```
pub   3072R/FFD3035B 2012-11-26 [expires: 2017-11-25]
      Key fingerprint = 443B 5363 564F 06C3 EA54  9482 209E 9B54 FFD3 035B
uid                  Edson Brandi <ebrandi@FreeBSD.org>
uid                  Edson Brandi <ebrandi@fugspbr.org>
uid                  Edson Brandi <ebrandi@ebrandi.eti.br>
uid                  Edson Brandi <edson.brandi@gmail.com>
uid                  Edson Brandi <ebrandi@primeirospassos.org>
uid                  Edson Brandi <ebrandi@gmail.com>
uid                  Edson Brandi <ebrandi@fug.com.br>
uid                  Edson Brandi <contato@edsonbrandi.com>
uid                  Edson Brandi (Born 1977-08-14 in S. S. DA GRAMA, SP - Brazil)
sub   3072R/A34B8175 2012-11-26 [expires: 2013-11-26]
sub   3072R/4EB0E0EA 2012-11-26 [expires: 2013-11-26]
sub   3072R/89917E73 2012-11-26 [expires: 2013-11-26]
```

D.3.35. Hartmut Brandt <harti@FreeBSD.org>

```
pub  1024D/5920099F 2003-01-29 Hartmut Brandt <brandt@fokus.fraunhofer.de>
     Key fingerprint = F60D 09A0 76B7 31EE 794B  BB91 082F 291D 5920 099F
uid                           Hartmut Brandt <harti@freebsd.org>
sub  1024g/21D30205 2003-01-29
```

D.3.36. Oliver Braun <obraun@FreeBSD.org>

```
pub  1024D/EF25B1BA 2001-05-06 Oliver Braun <obraun@unsane.org>
     Key fingerprint = 6A3B 042A 732E 17E4 B6E7  3EAF C0B1 6B7D EF25 B1BA
uid                            Oliver Braun <obraun@obraun.net>
uid                            Oliver Braun <obraun@freebsd.org>
uid                            Oliver Braun <obraun@haskell.org>
```

```
sub  1024g/09D28582 2001-05-06
```

D.3.37. Max Brazhnikov <makc@FreeBSD.org>

```
pub   1024D/ACB3CD12 2008-08-18
      Key fingerprint = 4BAA 200E 720A 0BD1 7BB0  9DFD FBD9 08C2 ACB3 CD12
uid                  Max Brazhnikov <makc@FreeBSD.org>
uid                  Max Brazhnikov <makc@issp.ac.ru>
sub   1024g/5FAA4088 2008-08-18
```

D.3.38. Jonathan M. Bresler <jmb@FreeBSD.org>

```
pub  1024R/97E638DD 1996-06-05 Jonathan M. Bresler <jmb@Bresler.org>
     Key fingerprint = 31 57 41 56 06 C1 40 13  C5 1C E3 E5 DC 62 0E FB
uid                            Jonathan M. Bresler <jmb@FreeBSD.ORG>
uid                            Jonathan M. Bresler
uid                            Jonathan M. Bresler <Jonathan.Bresler@USi.net>
uid                            Jonathan M. Bresler <jmb@Frb.GOV>
```

D.3.39. Antoine Brodin <antoine@FreeBSD.org>

```
pub   1024D/50CC2671 2008-02-03
      Key fingerprint = F3F7 72F0 9C4C 9E56 4BE9  44EA 1B80 31F3 50CC 2671
uid                  Antoine Brodin <antoine@FreeBSD.org>
sub   2048g/6F4AFBE5 2008-02-03
```

D.3.40. Diane Bruce <db@FreeBSD.org>

```
pub   2048R/8E9CAA7B 2012-05-16
      Key fingerprint = 8B08 E022 705D 0083 64C4  5E60 5148 0C74 8E9C AA7B
uid                  Diane Bruce <db@db.net>
uid                  Diane Bruce <db@FreeBSD.org>
sub   2048R/932E5985 2012-05-16
```

D.3.41. Christian Brüffer <brueffer@FreeBSD.org>

```
pub  1024D/A0ED982D 2002-10-14 Christian Brueffer <chris@unixpages.org>
     Key fingerprint = A5C8 2099 19FF AACA F41B  B29B 6C76 178C A0ED 982D
uid                            Christian Brueffer <brueffer@hitnet.rwth-aachen.de>
uid                            Christian Brueffer <brueffer@FreeBSD.org>
sub  4096g/1DCC100F 2002-10-14
```

D.3.42. Markus Brüffer <markus@FreeBSD.org>

```
pub   1024D/78F8A8D4 2002-10-21
      Key fingerprint = 3F9B EBE8 F290 E5CC 1447  8760 D48D 1072 78F8 A8D4
uid                  Markus Brueffer <markus@brueffer.de>
uid                  Markus Brueffer <buff@hitnet.rwth-aachen.de>
uid                  Markus Brueffer <mbrueffer@mi.rwth-aachen.de>
uid                  Markus Brueffer <markus@FreeBSD.org>
sub   4096g/B7E5C7B6 2002-10-21
```

D.3.43. Sean Bruno <sbruno@FreeBSD.org>

```
pub   2048R/08E81687 2012-10-15
      Key fingerprint = B9F9 138F 349C D3B2 2AA4  1398 1909 45DC 08E8 1687
uid                  Sean Bruno (clusteradm and developer key) <sbruno@freebsd.org>
sub   2048R/BCC23981 2012-10-15
```

D.3.44. Oleg Bulyzhin <oleg@FreeBSD.org>

```
pub   1024D/78CE105F 2004-02-06
      Key fingerprint = 98CC 3E66 26DE 50A8 DBC4  EB27 AF22 DCEF 78CE 105F
uid                  Oleg Bulyzhin <oleg@FreeBSD.org>
uid                  Oleg Bulyzhin <oleg@rinet.ru>
sub   1024g/F747C159 2004-02-06
```

D.3.45. Michael Bushkov <bushman@FreeBSD.org>

```
pub   1024D/F694C6E4 2007-03-11 [expires: 2008-03-10]
      Key fingerprint = 4278 4392 BF6B 2864 C48E  0FA9 7216 C73C F694 C6E4
uid                  Michael Bushkov <bushman@rsu.ru>
uid                  Michael Bushkov <bushman@freebsd.org>
sub   2048g/5A783997 2007-03-11 [expires: 2008-03-10]
```

D.3.46. Jayachandran C. <jchandra@FreeBSD.org>

```
pub   1024D/3316E465 2010-05-19
      Key fingerprint = 320B DB08 4FE3 BCFD 60AF  E4DB F486 015F 3316 E465
uid                  Jayachandran C. <jchandra@freebsd.org>
sub   2048g/1F7755F9 2010-05-19
```

D.3.47. Jesus R. Camou <jcamou@FreeBSD.org>

```
pub   1024D/C2161947 2005-03-01
      Key fingerprint = 274C B265 48EC 42AE A2CA  47D9 7D98 588A C216 1947
uid                  Jesus R. Camou <jcamou@FreeBSD.org>
sub   2048g/F8D2A8DF 2005-03-01
```

D.3.48. José Alonso Cárdenas Márquez <acm@FreeBSD.org>

```
pub   1024D/9B21BC19 2006-07-18
      Key fingerprint = 4156 2EAC A11C 9651 713B  3FC1 195F D4A8 9B21 BC19
uid                  Jose Alonso Cardenas Marquez <acm@FreeBSD.org>
sub   2048g/ADA16C52 2006-07-18
```

D.3.49. Pietro Cerutti <gahr@FreeBSD.org>

```
pub   1024D/9571F78E 2006-05-17
      Key fingerprint = 1203 92B5 3919 AF84 9B97  28D6 C0C2 6A98 9571 F78E
uid                  Pietro Cerutti <gahr@gahr.ch>
uid                  Pietro Cerutti (The FreeBSD Project) <gahr@FreeBSD.org>
sub   2048g/F24227D5 2006-05-17 [expires: 2011-05-16]
```

D.3.50. Dmitry Chagin <dchagin@FreeBSD.org>

```
pub   1024D/738EFCED 2009-02-27
      Key fingerprint = 3F3F 8B87 CE09 9E10 3606  6ACA D2DD 936F 738E FCED
uid                  Dmitry Chagin <dchagin@freebsd.org>
uid                  Dmitry Chagin (dchagin key) <chagin.dmitry@gmail.com>
sub   2048g/6A3FDFF9 2009-02-27
```

D.3.51. Hye-Shik Chang <perky@FreeBSD.org>

```
pub  1024D/CFDB4BA4 1999-04-23 Hye-Shik Chang <perky@FreeBSD.org>
     Key fingerprint = 09D9 57D6 58BA 44DD CAEC  71CD 0D65 2C59 CFDB 4BA4
uid                           Hye-Shik Chang <hyeshik@gmail.com>
sub  1024g/A94A8ED1 1999-04-23
```

D.3.52. Jonathan Chen <jon@FreeBSD.org>

```
pub  1024D/2539468B 1999-10-11 Jonathan Chen <jon@spock.org>
     Key fingerprint = EE31 CDA1 A105 C8C9 5365  3DB5 C2FC 86AA 2539 468B
uid                            Jonathan Chen <jon@freebsd.org>
uid                            Jonathan Chen <chenj@rpi.edu>
uid                            Jonathan Chen <spock@acm.rpi.edu>
```

```
uid                           Jonathan Chen <jon@cs.rpi.edu>
sub  3072g/B81EF1DB 1999-10-11
```

D.3.53. Jonathan Anderson <jonathan@FreeBSD.org>

```
pub   1024D/E3BBCA48 2006-06-17
      Key fingerprint = D7C6 9096 874F 707E 48F8  FAB7 22A6 6E53 E3BB CA48
uid                  Jonathan Anderson <jonathan@FreeBSD.org>
uid                  Jonathan Anderson <jonathan.anderson@ieee.org>
uid                  Jonathan Anderson <anderson@engr.mun.ca>
uid                  Jonathan Anderson <jonathan.anderson@mun.ca>
sub   2048g/A703650D 2006-06-17
```

D.3.54. Fukang Chen <loader@FreeBSD.org>

```
pub   4096R/6BD4DDE6 2012-10-26
      Key fingerprint = A33E 88AB D358 DA49 59A6  B263 A9A2 599C 6BD4 DDE6
uid                  loader <loader@FreeBSD.org>
uid                  loader <loader@FreeBSDMall.com>
sub   4096R/1036D26C 2012-10-26
```

D.3.55. Luoqi Chen <luoqi@FreeBSD.org>

```
pub  1024D/2926F3BE 2002-02-22 Luoqi Chen <luoqi@FreeBSD.org>
     Key fingerprint = B470 A815 5917 D9F4 37F3  CE2A 4D75 3BD1 2926 F3BE
uid                           Luoqi Chen <luoqi@bricore.com>
uid                           Luoqi Chen <lchen@onetta.com>
sub  1024g/5446EB72 2002-02-22
```

D.3.56. Andrey A. Chernov <ache@FreeBSD.org>

```
pub   1024D/964474DD 2006-12-26
      Key fingerprint = 0F63 1B61 D76D AA23 1591  EA09 560E 582B 9644 74DD
uid                  Andrey Chernov <ache@freebsd.org>
uid                  [jpeg image of size 4092]
sub   2048g/08331894 2006-12-26
```

D.3.57. Alexander V. Chernikov <melifaro@FreeBSD.org>

```
pub   1024D/2675AB69 2008-02-17
      Key fingerprint = 00D2 E063 2FB0 2990 C602  50FD C1C2 7889 2675 AB69
uid                  Alexander V. Chernikov <melifaro@yandex-team.ru>
uid                  Alexander V. Chernikov <melifaro@ipfw.ru>
uid                  Alexander V. Chernikov <melifaro@freebsd.org>
```

```
sub   4096g/BC64F40C 2008-02-17
```

D.3.58. Sean Chittenden <seanc@FreeBSD.org>

```
pub  1024D/EE278A28 2004-02-08 Sean Chittenden <sean@chittenden.org>
     Key fingerprint = E41F F441 7E91 6CBA 1844  65CF B939 3C78 EE27 8A28
sub  2048g/55321853 2004-02-08
```

D.3.59. Junho CHOI <cjh@FreeBSD.org>

```
pub  1024D/E60260F5 2002-10-14 CHOI Junho (Work) <cjh@wdb.co.kr>
     Key fingerprint = 1369 7374 A45F F41A F3C0  07E3 4A01 C020 E602 60F5
uid                            CHOI Junho (Personal) <cjh@kr.FreeBSD.org>
uid                            CHOI Junho (FreeBSD) <cjh@FreeBSD.org>
sub  1024g/04A4FDD8 2002-10-14
```

D.3.60. Crist J. Clark <cjc@FreeBSD.org>

```
pub  1024D/FE886AD3 2002-01-25 Crist J. Clark <cjclark@jhu.edu>
     Key fingerprint = F04E CCD7 3834 72C2 707F  0A8F 259F 8F4B FE88 6AD3
uid                            Crist J. Clark <cjclark@alum.mit.edu>
uid                            Crist J. Clark <cjc@freebsd.org>
sub  1024g/9B6BAB99 2002-01-25
```

D.3.61. Joe Marcus Clarke <marcus@FreeBSD.org>

```
pub  1024D/FE14CF87 2002-03-04 Joe Marcus Clarke (FreeBSD committer address) <marcus@FreeBSD.org>
     Key fingerprint = CC89 6407 73CC 0286 28E4  AFB9 6F68 8F8A FE14 CF87
uid                            Joe Marcus Clarke <marcus@marcuscom.com>
sub  1024g/B9ACE4D2 2002-03-04
```

D.3.62. Nik Clayton <nik@FreeBSD.org>

```
pub  1024D/2C37E375 2000-11-09 Nik Clayton <nik@freebsd.org>
     Key fingerprint = 15B8 3FFC DDB4 34B0 AA5F  94B7 93A8 0764 2C37 E375
uid                            Nik Clayton <nik@slashdot.org>
uid                            Nik Clayton <nik@crf-consulting.co.uk>
uid                            Nik Clayton <nik@ngo.org.uk>
uid                            Nik Clayton <nik@bsdi.com>
sub  1024g/769E298A 2000-11-09
```

D.3.63. Benjamin Close <benjsc@FreeBSD.org>

```
pub   1024D/4842B5B4 2002-04-10
      Key fingerprint = F00D C83D 5F7E 5561 DF91  B74D E602 CAA3 4842 B5B4
uid                  Benjamin Simon Close <Benjamin.Close@clearchain.com>
uid                  Benjamin Simon Close <benjsc@FreeBSD.org>
uid                  Benjamin Simon Close <benjsc@clearchain.com>
sub   2048g/3FA8A57E 2002-04-10
```

D.3.64. Tijl Coosemans <tijl@FreeBSD.org>

```
pub   2048D/20A0B62B 2010-07-13
      Key fingerprint = 39AA F580 6B44 5161 9F86  ED49 7E80 92D8 20A0 B62B
uid                  Tijl Coosemans <tijl@coosemans.org>
uid                  Tijl Coosemans <tijl@freebsd.org>
sub   2048g/7D71BA74 2010-07-13
```

D.3.65. Raphael Kubo da Costa <rakuco@FreeBSD.org>

```
pub   4096R/18DCEED6 2011-10-03
      Key fingerprint = 6911 54FE BA6E 6106 5789  7099 8DD0 7D21 18DC EED6
uid                  Raphael Kubo da Costa (Personal key) <rakuco@FreeBSD.org>
```

D.3.66. Alan L. Cox <alc@FreeBSD.org>

```
pub   2048R/33E2893B 2013-06-15
      Key fingerprint = FC7C 93FD 2C2C ABA5 C1D1  3E74 8513 043C 33E2 893B
uid                  Alan Cox <alc@FreeBSD.org>
uid                  Alan Cox <alc@cs.rice.edu>
uid                  Alan Cox <alc@rice.edu>
sub   2048R/693757AA 2013-06-15
```

D.3.67. Bruce Cran <brucec@FreeBSD.org>

```
pub   2048R/6AF6F99E 2010-01-29
      Key fingerprint = 9A3C AE57 2706 B0E3 4B8A  8374 5787 A72B 6AF6 F99E
uid                  Bruce Cran <brucec@FreeBSD.org>
uid                  Bruce Cran <bruce@cran.org.uk>
sub   2048R/1D665CEE 2010-01-29
```

D.3.68. Frederic Culot <culot@FreeBSD.org>

```
pub   1024D/34876C5B 2006-08-26
      Key fingerprint = 50EE CE94 E43E BA85 CB67  262B B739 1A26 3487 6C5B
uid                  Frederic Culot <culot@FreeBSD.org>
uid                  Frederic Culot <frederic@culot.org>
sub   2048g/F1EF901F 2006-08-26
```

D.3.69. Aaron Dalton <aaron@FreeBSD.org>

```
pub   1024D/8811D2A4 2006-06-21 [expires: 2011-06-20]
      Key fingerprint = 8DE0 3CBB 3692 992F 53EF  ACC7 BE56 0A4D 8811 D2A4
uid                  Aaron Dalton <aaron@freebsd.org>
sub   2048g/304EE8E5 2006-06-21 [expires: 2011-06-20]
```

D.3.70. Baptiste Daroussin <bapt@FreeBSD.org>

```
pub   1024D/49A4E84C 2008-11-19
      Key fingerprint = A14B A5FC B860 86DE 73E2  B24C F244 ED31 49A4 E84C
uid                  Baptiste Daroussin <bapt@etoilebsd.net>
uid                  Baptiste Daroussin <baptiste.daroussin@gmail.com>
uid                  Baptiste Daroussin <bapt@FreeBSD.org>
sub   2048g/54AB46B4 2008-11-19
```

D.3.71. Ceri Davies <ceri@FreeBSD.org>

```
pub   1024D/34B7245F 2002-03-08
      Key fingerprint = 9C88 EB05 A908 1058 A4AE  9959 A1C7 DCC1 34B7 245F
uid                  Ceri Davies <ceri@submonkey.net>
uid                  Ceri Davies <ceri@FreeBSD.org>
uid                  Ceri Davies <ceri@opensolaris.org>
sub   1024g/0C482CBC 2002-03-08
```

D.3.72. Brad Davis <brd@FreeBSD.org>

```
pub   1024D/ED0A754D 2005-05-14 [expires: 2014-02-21]
      Key fingerprint = 5DFD D1A6 BEEE A6D4 B3F5  4236 D362 3291 ED0A 754D
uid                  Brad Davis <so14k@so14k.com>
uid                  Brad Davis <brd@FreeBSD.org>
sub   2048g/1F29D404 2005-05-14 [expires: 2014-02-21]
```

D.3.73. Pawel Jakub Dawidek <pjd@FreeBSD.org>

```
pub  1024D/B1293F34 2004-02-02 Pawel Jakub Dawidek <Pawel@Dawidek.net>
     Key fingerprint = A3A3 5B4D 9CF9 2312 0783  1B1D 168A EF5D B129 3F34
uid                            Pawel Jakub Dawidek <pjd@FreeBSD.org>
uid                            Pawel Jakub Dawidek <pjd@FreeBSD.pl>
sub  2048g/3EEC50A7 2004-02-02 [expires: 2006-02-01]
```

D.3.74. Brian S. Dean <bsd@FreeBSD.org>

```
pub  1024D/723BDEE9 2002-01-23 Brian S. Dean <bsd@FreeBSD.org>
     Key fingerprint = EF49 7ABE 47ED 91B3 FC3D  7EA5 4D90 2FF7 723B DEE9
sub  1024g/4B02F876 2002-01-23
```

D.3.75. Carl Delsey <carl@FreeBSD.org>

```
pub   4096R/FB3B5D38 2013-01-15
      Key fingerprint = F0E5 3849 C6C3 668B 68A3  BCC7 6031 E963 FB3B 5D38
uid                  Carl Delsey <carl@FreeBSD.org>
sub   4096R/256F29D3 2013-01-15
```

D.3.76. Vasil Dimov <vd@FreeBSD.org>

```
pub   1024D/F6C1A420 2004-12-08
      Key fingerprint = B1D5 04C6 26CC 0D20 9525  14B8 170E 923F F6C1 A420
uid                  Vasil Dimov <vd@FreeBSD.org>
uid                  Vasil Dimov <vd@datamax.bg>
sub   4096g/A0148C94 2004-12-08
```

D.3.77. Roman Divacky <rdivacky@FreeBSD.org>

```
pub   1024D/3DC2044C 2006-11-15
      Key fingerprint = 6B61 25CA 49BC AAC5 21A9  FA7A 2D51 23E8 3DC2 044C
uid                  Roman Divacky <rdivacky@freebsd.org>
sub   2048g/39BDCE16 2006-11-15
```

D.3.78. Alexey Dokuchaev <danfe@FreeBSD.org>

```
pub  1024D/3C060B44 2004-08-23 Alexey Dokuchaev <danfe@FreeBSD.org>
     Key fingerprint = D970 08A4 922C 8D63 0C19  8D27 F421 76EE 3C06 0B44
sub  1024g/70BAE967 2004-08-23
```

D.3.79. Dima Dorfman <dd@FreeBSD.org>

```
pub   1024D/69FAE582 2001-09-04
      Key fingerprint = B340 8338 7DA3 4D61 7632  098E 0730 055B 69FA E582
uid                  Dima Dorfman <dima@trit.org>
uid                  Dima Dorfman <dima@unixfreak.org>
uid                  Dima Dorfman <dd@freebsd.org>
sub   2048g/65AF3B89 2003-08-19 [expires: 2005-08-18]
sub   2048g/8DB0CF2C 2005-05-29 [expires: 2007-05-29]
```

D.3.80. Bryan Drewery <bdrewery@FreeBSD.org>

```
pub   4096R/3C9B0CF9 2012-04-06 [expires: 2017-04-05]
      Key fingerprint = 36FE BE99 2F52 80DF 4811  362A 6E78 2AC0 3C9B 0CF9
uid                  Bryan Drewery <bryan@shatow.net>
uid                  Bryan Drewery <bdrewery@gmail.com>
uid                  Bryan Drewery <bryan@xzibition.com>
uid                  Bryan Drewery <bdrewery@FreeBSD.org>
sub   4096R/9E2CE2D3 2012-04-06 [expires: 2017-04-05]
```

D.3.81. Olivier Duchateau <olivierd@FreeBSD.org>

```
pub   2048R/22431859 2012-05-28 [expires: 2017-05-27]
      Key fingerprint = C057 112A 4A27 B5F2 CD8F  6C9A FC5A 0167 2243 1859
uid                  Olivier Duchateau <duchateau.olivier@gmail.com>
sub   2048R/63A85BDF 2012-05-28 [expires: 2017-05-27]
```

D.3.82. Bruno Ducrot <bruno@FreeBSD.org>

```
pub   1024D/7F463187 2000-12-29
      Key fingerprint = 7B79 E1D6 F5A1 6614 792F  D906 899B 4D28 7F46 3187
uid                  Ducrot Bruno (Poup Master) <ducrot@poupinou.org>
sub   1024g/40282874 2000-12-29
```

D.3.83. Alex Dupre <ale@FreeBSD.org>

```
pub  1024D/CE5F554D 1999-06-27 Alex Dupre <sysadmin@alexdupre.com>
     Key fingerprint = DE23 02EA 5927 D5A9 D793  2BA2 8115 E9D8 CE5F 554D
uid                           Alex Dupre <ale@FreeBSD.org>
uid                           [jpeg image of size 5544]
uid                           Alex Dupre <ICQ:5431856>
sub  2048g/FD5E2D21 1999-06-27
```

D.3.84. Peter Edwards <peadar@FreeBSD.org>

```
pub  1024D/D80B4B3F 2004-03-01 Peter Edwards <peadar@FreeBSD.org>
     Key fingerprint = 7A8A 9756 903E BEF2 4D9E  3C94 EE52 52F7 D80B 4B3F
uid                           Peter Edwards <pmedwards@eircom.net>
```

D.3.85. Daniel Eischen <deischen@FreeBSD.org>

```
pub   4096R/7D15560B 2012-11-17
      Key fingerprint = 0039 2133 69CA 14D3 236A  E331 361A 68B2 7D15 560B
uid                  Daniel Eischen <deischen@FreeBSD.org>
sub   4096R/A51F81F7 2012-11-17
```

D.3.86. Josef El-Rayes <josef@FreeBSD.org>

```
pub  2048R/A79DB53C 2004-01-04 Josef El-Rayes <josef@FreeBSD.org>
     Key fingerprint = 58EB F5B7 2AB9 37FE 33C8  716B 59C5 22D9 A79D B53C
uid                           Josef El-Rayes <josef@daemon.li>
```

D.3.87. Lars Engels <lme@FreeBSD.org>

```
pub   1024D/C0F769F8 2004-08-27
      Key fingerprint = 17FC 08E1 5E09 BD21 489E  2050 29CE 75DA C0F7 69F8
uid                  Lars Engels <lars.engels@0x20.net>
sub   1024g/8AD5BF9D 2004-08-27
```

D.3.88. Udo Erdelhoff <ue@FreeBSD.org>

```
pub  1024R/E74FA871 1994-07-19 Udo Erdelhoff <uer@de.uu.net>
     Key fingerprint = 8C B1 80 CA 2C 52 73 81  FB A7 B4 03 C5 32 C8 67
uid                           Udo Erdelhoff <ue@nathan.ruhr.de>
uid                           Udo Erdelhoff <ue@freebsd.org>
uid                           Udo Erdelhoff <uerdelho@eu.uu.net>
uid                           Udo Erdelhoff <uerdelho@uu.net>
```

D.3.89. Ruslan Ermilov <ru@FreeBSD.org>

```
pub  1024D/996E145E 2004-06-02 Ruslan Ermilov (FreeBSD) <ru@FreeBSD.org>
     Key fingerprint = 274E D201 71ED 11F6 9CCB  0194 A917 E9CC 996E 145E
uid                           Ruslan Ermilov (FreeBSD Ukraine) <ru@FreeBSD.org.ua>
uid                           Ruslan Ermilov (IPNet) <ru@ip.net.ua>
sub  1024g/557E3390 2004-06-02 [expires: 2007-06-02]
```

D.3.90. Lukas Ertl <le@FreeBSD.org>

```
pub  1024D/F10D06CB 2000-11-23 Lukas Ertl <le@FreeBSD.org>
     Key fingerprint = 20CD C5B3 3A1D 974E 065A  B524 5588 79A9 F10D 06CB
uid                            Lukas Ertl <a9404849@unet.univie.ac.at>
uid                            Lukas Ertl <l.ertl@univie.ac.at>
uid                            Lukas Ertl <le@univie.ac.at>
sub  1024g/5960CE8E 2000-11-23
```

D.3.91. Brendan Fabeny <bf@FreeBSD.org>

```
pub   2048R/9806EBC1 2010-06-08 [expires: 2012-06-07]
      Key fingerprint = 2075 ADD3 7634 A4F9 5357  D934 08E7 06D9 9806 EBC1
uid                  b. f. <bf@freebsd.org>
sub   2048R/1CD0AD79 2010-06-08 [expires: 2012-06-07]
```

D.3.92. Guido Falsi <madpilot@FreeBSD.org>

```
pub   2048R/56CBD293 2012-04-12
      Key fingerprint = F317 2057 E17E 4E3A 3DA5  9E1D 1AE6 860E 56CB D293
uid                  Guido Falsi <madpilot@FreeBSD.org>
uid                  Guido Falsi <mad@madpilot.net>
sub   2048R/1F9772C5 2012-04-12
```

D.3.93. Rong-En Fan <rafan@FreeBSD.org>

```
pub   1024D/86FD8C68 2004-06-04
      Key fingerprint = DC9E 5B4D 2DDA D5C7 B6F8  6E69 D78E 1091 86FD 8C68
uid                  Rong-En Fan <rafan@infor.org>
uid                  Rong-En Fan <rafan@csie.org>
uid                  Rong-En Fan <rafan@FreeBSD.org>
sub   2048g/42A8637E 2009-01-25 [expires: 2012-07-08]
```

D.3.94. Stefan Farfeleder <stefanf@FreeBSD.org>

```
pub  1024D/8BEFD15F 2004-03-14 Stefan Farfeleder <stefan@fafoe.narf.at>
     Key fingerprint = 4220 FE60 A4A1 A490 5213  27A6 319F 8B28 8BEF D15F
uid                            Stefan Farfeleder <stefanf@complang.tuwien.ac.at>
uid                            Stefan Farfeleder <stefanf@FreeBSD.org>
uid                            Stefan Farfeleder <stefanf@ten15.org>
sub  2048g/418753E9 2004-03-14 [expires: 2007-03-14]
```

D.3.95. Babak Farrokhi <farrokhi@FreeBSD.org>

```
pub   1024D/7C810476 2005-12-22
      Key fingerprint = AABD 388F A207 58B4 2EE3  5DFD 4FC1 32C3 7C81 0476
uid                  Babak Farrokhi <farrokhi@FreeBSD.org>
uid                  Babak Farrokhi <babak@farrokhi.net>
sub   2048g/2A5F93C7 2005-12-22
```

D.3.96. Chris D. Faulhaber <jedgar@FreeBSD.org>

```
pub  1024D/FE817A50 2000-12-20 Chris D. Faulhaber <jedgar@FreeBSD.org>
     Key fingerprint = A47D A838 9216 F921 A456  54FF 39B6 86E0 FE81 7A50
uid                            Chris D. Faulhaber <jedgar@fxp.org>
sub  2048g/93452698 2000-12-20
```

D.3.97. Mark Felder <feld@FreeBSD.org>

```
pub   2048R/E64C94FE 2013-06-25
      Key fingerprint = 71ED 6A7F F4D7 430A BDF3  A180 BF01 619F E64C 94FE
uid                  Mark Felder <feld@freebsd.org>
sub   2048R/FDC20CA9 2013-06-25
```

D.3.98. Brian F. Feldman <green@FreeBSD.org>

```
pub  1024D/41C13DE3 2000-01-11 Brian Fundakowski Feldman <green@FreeBSD.org>
     Key fingerprint = 6A32 733A 1BF6 E07B 5B8D  AE14 CC9D DCA2 41C1 3DE3
sub  1024g/A98B9FCC 2000-01-11 [expires: 2001-01-10]

pub  1024D/773905D6 2000-09-02 Brian Fundakowski Feldman <green@FreeBSD.org>
     Key fingerprint = FE23 7481 91EA 5E58 45EA  6A01 B552 B043 7739 05D6
sub  2048g/D2009B98 2000-09-02
```

D.3.99. Mário Sérgio Fujikawa Ferreira <lioux@FreeBSD.org>

```
pub   1024D/75A63712 2006-02-23 [expires: 2007-02-23]
      Key fingerprint = 42F2 2F74 8EF9 5296 898F  C981 E9CF 463B 75A6 3712
uid                  Mario Sergio Fujikawa Ferreira (lioux) <lioux@FreeBSD.org>
uid                  Mario Sergio Fujikawa Ferreira <lioux@uol.com.br>
sub   4096g/BB7D80F2 2006-02-23 [expires: 2007-02-23]
```

D.3.100. Matthew Fleming <mdf@FreeBSD.org>

```
pub   2048R/A783DAA2 2012-11-22 [expires: 2016-11-22]
      Key fingerprint = 773F E069 BE98 CE96 4AC6  B8AB 1A1B 255E A783 DAA2
uid                  Matthew D Fleming <mdf356@gmail.com>
uid                  Matthew D Fleming <mdf@FreeBSD.org>
sub   2048R/4015B7AA 2012-11-22 [expires: 2016-11-22]
```

D.3.101. Tony Finch <fanf@FreeBSD.org>

```
pub  1024D/84C71B6E 2002-05-03 Tony Finch <dot@dotat.at>
     Key fingerprint = 199C F25B 2679 6D04 63C5  2159 FFC0 F14C 84C7 1B6E
uid                            Tony Finch <fanf@FreeBSD.org>
uid                            Tony Finch <fanf@apache.org>
uid                            Tony Finch <fanf2@cam.ac.uk>
sub  2048g/FD101E8B 2002-05-03
```

D.3.102. Marc Fonvieille <blackend@FreeBSD.org>

```
pub  1024D/4F8E74E8 2004-12-25 Marc Fonvieille <blackend@FreeBSD.org>
     Key fingerprint = 55D3 4883 4A04 828A A139  A5CF CD0F 51C0 4F8E 74E8
uid                            Marc Fonvieille <marc@blackend.org>
uid                            Marc Fonvieille <marc@freebsd-fr.org>
sub  1024g/37AD4E7D 2004-12-25
```

D.3.103. Pete Fritchman <petef@FreeBSD.org>

```
pub  1024D/74B91CFD 2001-01-30 Pete Fritchman <petef@FreeBSD.org>
     Key fingerprint = 9A9F 8A13 DB0D 7777 8D8E  1CB2 C5C9 A08F 74B9 1CFD
uid                            Pete Fritchman <petef@databits.net>
uid                            Pete Fritchman <petef@csh.rit.edu>
sub  1024g/0C02AF0C 2001-01-30
```

D.3.104. Bernhard Fröhlich <decke@FreeBSD.org>

```
pub   1024D/CF5840D4 2008-01-07 [expires: 2015-05-05]
      Key fingerprint = 47F6 BDF1 DF9E 81E2 2C54  8A06 E796 7A5A CF58 40D4
uid                  Bernhard Fröhlich <decke@FreeBSD.org>
uid                  Bernhard Fröhlich <decke@bluelife.at>
sub   2048g/4E51CE79 2008-01-07
```

D.3.105. Bill Fumerola <billf@FreeBSD.org>

```
pub  1024D/7F868268 2000-12-07 Bill Fumerola (FreeBSD Developer) <billf@FreeBSD.org>
     Key fingerprint = 5B2D 908E 4C2B F253 DAEB  FC01 8436 B70B 7F86 8268
uid                            Bill Fumerola (Security Yahoo) <fumerola@yahoo-inc.com>
sub  1024g/43980DA9 2000-12-07
```

D.3.106. Andriy Gapon <avg@FreeBSD.org>

```
pub   2048R/A651FE2F 2009-02-16
      Key fingerprint = F234 4D58 DEFF 5E3A 4E0F  13BC 74A5 2D27 A651 FE2F
uid                  Andriy Gapon (FreeBSD) <avg@FreeBSD.org>
uid                  Andriy Gapon (FreeBSD) <avg@freebsd.org>
uid                  Andriy Gapon (FreeBSD) <avg@icyb.net.ua>
sub   4096R/F9A4D312 2009-02-16
```

D.3.107. Beat Gätzi <beat@FreeBSD.org>

```
pub   1024D/774249DB 2009-01-28 [expires: 2014-01-27]
      Key fingerprint = C410 3187 5B29 DD02 745F  0890 40C5 BCF7 7742 49DB
uid                  Beat Gaetzi <beat@FreeBSD.org>
sub   2048g/173CFFCA 2009-01-28 [expires: 2014-01-27]
```

D.3.108. Daniel Geržo <danger@FreeBSD.org>

```
pub   1024D/DA913352 2007-08-30 [expires: 2008-08-29]
      Key fingerprint = 7372 3F15 F839 AFF5 4052  CAC7 1ADA C204 DA91 3352
uid                  Daniel Gerzo <gerzo@rulez.sk>
uid                  Daniel Gerzo <danger@rulez.sk>
uid                  Daniel Gerzo (The FreeBSD Project) <danger@FreeBSD.org>
uid                  Daniel Gerzo (Micronet, a.s.) <gerzo@micronet.sk>
sub   2048g/C5D57BDC 2007-08-30 [expires: 2008-08-29]
```

D.3.109. Simon J. Gerraty <sjg@FreeBSD.org>

```
pub   1024D/B6CC76BF 2002-06-12
      Key fingerprint = F3BA D6CB E1F8 02EA 705F  BCAD 6125 F840 B6CC 76BF
uid                  Simon J. Gerraty <sjg@crufty.net>
uid                  Simon J. Gerraty <sjg@juniper.net>
uid                  Simon J. Gerraty <sjg@NetBSD.org>
uid                  Simon J. Gerraty <sjg@FreeBSD.org>
sub   1024g/D94B72B9 2002-06-12
```

D.3.110. Justin T. Gibbs <gibbs@FreeBSD.org>

```
pub   2048R/45A4FC2F 2012-02-10
      Key fingerprint = B98A C3AB 412B 094B D6FE  E713 FA5A 1E30 45A4 FC2F
uid                  Justin T. Gibbs <gibbs@FreeBSD.org>
uid                  Justin T. Gibbs <gibbs@FreeBSDFoundation.org>
uid                  Justin T. Gibbs <gibbs@scsiguy.com>
sub   2048R/AF6927F8 2012-02-10
```

D.3.111. Pedro Giffuni <pfg@FreeBSD.org>

```
pub   2048D/422BDFE4 2011-12-06
      Key fingerprint = A12B 7C6B 54C0 921B C64F  7B35 58DF 6813 422B DFE4
uid                  Pedro Giffuni (FreeBSD key signature) <pfg@FreeBSD.org>
sub   2048g/43A91DE0 2011-12-06
```

D.3.112. Palle Girgensohn <girgen@FreeBSD.org>

```
pub   2048R/4A6BAAAD 2012-02-23 [expires: 2016-02-23]
      Key fingerprint = BD8C 332C E630 31D6 2FDB  80BD 5FF2 A161 4A6B AAAD
uid                  Palle Girgensohn <girgen@pingpong.net>
uid                  [jpeg image of size 8260]
uid                  Palle Girgensohn <girgen@FreeBSD.org>
sub   2048R/6BC41243 2012-02-23 [expires: 2016-02-23]
```

D.3.113. Philip M. Gollucci <pgollucci@FreeBSD.org>

```
pub   1024D/DB9B8C1C 2008-04-15
      Key fingerprint = B90B FBC3 A3A1 C71A 8E70  3F8C 75B8 8FFB DB9B 8C1C
uid                  Philip M. Gollucci (FreeBSD Foundation) <pgollucci@freebsd.org>
uid                  Philip M. Gollucci (Riderway Inc.) <pgollucci@riderway.com>
uid                  Philip M. Gollucci <pgollucci@p6m7g8.com>
uid                  Philip M. Gollucci (ASF) <pgollucci@apache.org>
sub   2048g/73943732 2008-04-15
```

D.3.114. Daichi GOTO <daichi@FreeBSD.org>

```
pub  1024D/09EBADD6 2002-09-25 Daichi GOTO <daichi@freebsd.org>
     Key fingerprint = 620A 9A34 57FB 5E93 0828  28C7 C360 C6ED 09EB ADD6
sub  1024g/F0B1F1CA 2002-09-25
```

D.3.115. Marcus Alves Grando <mnag@FreeBSD.org>

```
pub   1024D/CDCC273F 2005-09-15 [expires: 2010-09-14]
      Key fingerprint = 57F9 DEC1 5BBF 06DE 44A5  9A4A 8BEE 5F3A CDCC 273F
uid                  Marcus Alves Grando <marcus@sbh.eng.br>
uid                  Marcus Alves Grando <marcus@corp.grupos.com.br>
uid                  Marcus Alves Grando <mnag@FreeBSD.org>
sub   2048g/698AC00C 2005-09-15 [expires: 2010-09-14]
```

D.3.116. Peter Grehan <grehan@FreeBSD.org>

```
pub  1024D/EA45EA7D 2004-07-13 Peter Grehan <grehan@freebsd.org>
     Key fingerprint = 84AD 73DC 370E 15CA 7556  43C8 F5C8 4450 EA45 EA7D
sub  2048g/0E122D70 2004-07-13
```

D.3.117. Jamie Gritton <jamie@FreeBSD.org>

```
pub   1024D/8832CB7F 2009-01-29
      Key fingerprint = 34F8 1E62 C7A5 7CB9 A91F  7864 8C5A F85E 8832 CB7F
uid                  James Gritton <jamie@FreeBSD.org>
sub   2048g/94E3594D 2009-01-29
```

D.3.118. William Grzybowski <wg@FreeBSD.org>

```
pub   2048R/CFC460C5 2012-09-28
      Key fingerprint = FC40 5CD8 0879 7F50 0036  D924 D9F7 8B27 CFC4 60C5
uid                  William Grzybowski (FreeBSD) <wg@freebsd.org>
uid                  William Grzybowski <william88@gmail.com>
sub   2048R/05577997 2012-09-28
```

D.3.119. Barbara Guida <bar@FreeBSD.org>

```
pub   2048R/3DF5F750 2012-11-13
      Key fingerprint = D367 F6C8 2A5F 2921 70D2  B446 27DD 6FD6 3DF5 F750
uid                  Barbara Guida <bar@FreeBSD.org>
uid                  Barbara Guida <barbara.freebsd@gmail.com>
sub   2048R/1DF7506C 2012-11-13
```

D.3.120. John-Mark Gurney <jmg@FreeBSD.org>

```
pub   1024D/6D3FA396 2011-03-03 [expires: 2016-03-01]
      Key fingerprint = 54BA 873B 6515 3F10 9E88  9322 9CB1 8F74 6D3F A396
uid                  John-Mark Gurney <jmg@FreeBSD.org>
uid                  John-Mark Gurney <jmg@funkthat.com>
```

```
sub   4096g/0A4C095E 2011-03-03 [expires: 2016-03-01]
```

D.3.121. Mateusz Guzik <mjg@FreeBSD.org>

```
pub   2048R/21489259 2012-06-03
      Key fingerprint = 3A9F 25FF ABF6 BB23 5C70  C61B 96D3 5178 2148 9259
uid                  Mateusz Guzik <mjg@freebsd.org>
sub   2048R/EA19FE8D 2012-06-03
```

D.3.122. Jason E. Hale <jhale@FreeBSD.org>

```
pub   3072D/8F2E5907 2012-09-07
      Key fingerprint = 009C 54BF 32D0 F373 8126  C8A1 D8DD 2CA4 8F2E 5907
uid                  Jason E. Hale <jhale@FreeBSD.org>
uid                  Jason E. Hale <bsdkaffee@gmail.com>
sub   4096g/7081A001 2012-09-07
```

D.3.123. Daniel Harris <dannyboy@FreeBSD.org>

```
pub  1024D/84D0D7E7 2001-01-15 Daniel Harris <dannyboy@worksforfood.com>
     Key fingerprint = 3C61 B8A1 3F09 D194 3259  7173 6C63 DA04 84D0 D7E7
uid                            Daniel Harris <dannyboy@freebsd.org>
uid                            Daniel Harris <dh@askdh.com>
uid                            Daniel Harris <dh@wordassault.com>
sub  1024g/9DF0231A 2001-01-15
```

D.3.124. Daniel Hartmeier <dhartmei@FreeBSD.org>

```
pub  1024R/6A3A7409 1994-08-15 Daniel Hartmeier <dhartmei@freebsd.org>
     Key fingerprint = 13 7E 9A F3 36 82 09 FE  FD 57 B8 5C 2B 81 7E 1F
```

D.3.125. Oliver Hauer <ohauer@FreeBSD.org>

```
pub   2048R/5D008F1A 2010-07-26
      Key fingerprint = E9EE C9A5 EB4C BD29 74D7  9178 E56E 06B3 5D00 8F1A
uid                  olli hauer <ohauer@FreeBSD.org>
uid                  olli hauer <ohauer@gmx.de>
sub   2048R/5E25776E 2010-07-26
```

D.3.126. Emanuel Haupt <ehaupt@FreeBSD.org>

```
pub   3072D/329A273C 2012-11-17 [expires: 2013-11-17]
      Key fingerprint = 920C A49A 5A23 F9E3 4EB0  4387 AB90 5C56 329A 273C
uid                  Emanuel Haupt <ehaupt@FreeBSD.org>
sub   3072g/70183B96 2012-11-17 [expires: 2013-11-17]
```

D.3.127. John Hay <jhay@FreeBSD.org>

```
pub  2048R/A9275B93 2000-05-10 John Hay <jhay@icomtek.csir.co.za>
     Key fingerprint = E7 95 F4 B9 D4 A7 49 6A  83 B9 77 49 28 9E 37 70
uid                            John Hay <jhay@mikom.csir.co.za>
uid                            Thawte Freemail Member <jhay@mikom.csir.co.za>
uid                            John Hay <jhay@csir.co.za>
uid                            John Hay <jhay@FreeBSD.ORG>
```

D.3.128. Sheldon Hearn <sheldonh@FreeBSD.org>

```
pub  1024D/74A06ACD 2002-06-20 Sheldon Hearn <sheldonh@starjuice.net>
     Key fingerprint = 01A3 EF91 9C5A 3633 4E01  8085 A462 57F1 74A0 6ACD
sub  1536g/C42F8AC8 2002-06-20
```

D.3.129. Mike Heffner <mikeh@FreeBSD.org>

```
pub  1024D/CDECBF99 2001-02-02 Michael Heffner <mheffner@novacoxmail.com>
     Key fingerprint = AFAB CCEB 68C7 573F 5110  9285 1689 1942 CDEC BF99
uid                            Michael Heffner <mheffner@vt.edu>
uid                            Michael Heffner <mikeh@FreeBSD.org>
uid                            Michael Heffner <spock@techfour.net>
uid                            Michael Heffner (ACM sysadmin) <mheffner@acm.vt.edu>
sub  1024g/3FE83FB5 2001-02-02
```

D.3.130. Martin Heinen <mheinen@FreeBSD.org>

```
pub  1024D/116C5C85 2002-06-17 Martin Heinen <mheinen@freebsd.org>
     Key fingerprint = C898 3FCD EEA0 17ED BEA9  564D E5A6 AFF2 116C 5C85
uid                            Martin Heinen <martin@sumuk.de>
sub  1024g/EA67506B 2002-06-17
```

D.3.131. Niels Heinen <niels@FreeBSD.org>

```
pub  1024D/5FE39B80 2004-12-06 Niels Heinen <niels.heinen@ubizen.com>
     Key fingerprint = 75D8 4100 CF5B 3280 543F  930C 613E 71AA 5FE3 9B80
uid                            Niels Heinen <niels@defaced.be>
```

```
uid                          Niels Heinen <niels@heinen.ws>
uid                          Niels Heinen <niels@FreeBSD.org>
sub  2048g/057F4DA7 2004-12-06
```

D.3.132. Jaakko Heinonen <jh@FreeBSD.org>

```
pub   1024D/53CCB781 2009-10-01 [expires: 2014-09-30]
      Key fingerprint = 3AED A2B6 B63D D771 1AFD  25FA DFDF 5B89 53CC B781
uid                  Jaakko Heinonen (FreeBSD) <jh@FreeBSD.org>
sub   4096g/BB97397E 2009-10-01 [expires: 2014-09-30]
```

D.3.133. Jason Helfman <jgh@FreeBSD.org>

```
pub   2048R/4150D3DC 2011-12-18 [expires: 2021-12-15]
      Key fingerprint = 8E0D C457 9A0F C91C 23F3  0454 2059 9A63 4150 D3DC
uid                  Jason Helfman <jgh@FreeBSD.org>
sub   2048R/695B1B92 2011-12-18 [expires: 2021-12-15]
```

D.3.134. Guy Helmer <ghelmer@FreeBSD.org>

```
pub   2048R/8F1CEBC4 2012-05-22
      Key fingerprint = 483E 9E6C C644 2520 C9FE  4E87 9989 CCAF 8F1C EBC4
uid                  Guy Helmer <guy.helmer@palisadesystems.com>
uid                  Guy Helmer <guy.helmer@gmail.com>
uid                  Guy Helmer <ghelmer@freebsd.org>
sub   2048R/2073E3F8 2012-05-22

pub  1024R/35F4ED2D 1997-01-26 Guy G. Helmer <ghelmer@freebsd.org>
     Key fingerprint = A2 59 4B 92 02 5B 9E B1  B9 4E 2E 03 29 D5 DC 3A
uid                          Guy G. Helmer <ghelmer@cs.iastate.edu>
uid                          Guy G. Helmer <ghelmer@palisadesys.com>
```

D.3.135. Maxime Henrion <mux@FreeBSD.org>

```
pub  1024D/881D4806 2003-01-09 Maxime Henrion <mux@FreeBSD.org>
     Key fingerprint = 81F1 BE2D 12F1 184A 77E4  ACD0 5563 7614 881D 4806
sub  2048g/D0B510C0 2003-01-09
```

D.3.136. Wen Heping <wen@FreeBSD.org>

```
pub   2048R/A03F07DA 2012-12-10
      Key fingerprint = 0258 F2C7 C123 E627 9E14  B4BA 270F 30AA A03F 07DA
uid                  Wen Heping (wen) <wen@FreeBSD.org>
sub   2048R/CFC8D6A9 2012-12-10
```

D.3.137. Dennis Herrmann <dhn@FreeBSD.org>

```
pub   4096R/F7CDCAA1 2012-08-26
      Key fingerprint = 0587 E730 68A6 2646 A991  505D CD9B 3A87 F7CD CAA1
uid                  Dennis ’dhn’ Herrmann (Everybody wants to go to heaven, but nobody wants to
sub   4096R/0A6D554F 2012-08-26
```

D.3.138. Justin Hibbits <jhibbits@FreeBSD.org>

```
pub   2048R/37BE2DB9 2011-12-01
      Key fingerprint = 8A12 7064 4F3D 339A 191D  AD52 30C7 858E 37BE 2DB9
uid                  Justin Hibbits <chmeeedalf@gmail.com>
uid                  Justin Hibbits <jhibbits@freebsd.org>
uid                  Justin Hibbits <jrh29@alumni.cwru.edu>
sub   2048R/A8DA156F 2011-12-01
```

D.3.139. Peter Holm <pho@FreeBSD.org>

```
pub   1024D/CF244E81 2008-11-17
      Key fingerprint = BE9B 32D8 89F1 F285 00E4  E4C5 EF3F B4B5 CF24 4E81
uid                  Peter Holm <pho@FreeBSD.org>
sub   2048g/E20A409F 2008-11-17
```

D.3.140. Michael L. Hostbaek <mich@FreeBSD.org>

```
pub  1024D/0F55F6BE 2001-08-07 Michael L. Hostbaek <mich@freebsdcluster.org>
     Key fingerprint = 4D62 9396 B19F 38D3 5C99  1663 7B0A 5212 0F55 F6BE
uid                           Michael L. Hostbaek <mich@freebsdcluster.dk>
uid                           Michael L. Hostbaek <mich@icommerce-france.com>
uid                           Micahel L. Hostbaek <mich@freebsd.dk>
uid                           Michael L. Hostbaek <mich@the-lab.org>
uid                           Michael L. Hostbaek <mich@freebsd.org>
sub  1024g/8BE4E30F 2001-08-07
```

D.3.141. Po-Chuan Hsieh <sunpoet@FreeBSD.org>

```
pub   4096R/CC57E36B 2010-09-21
      Key fingerprint = 8AD8 68F2 7D2B 0A10 7E9B  8CC0 DC44 247E CC57 E36B
uid                  Po-Chuan Hsieh (FreeBSD) <sunpoet@FreeBSD.org>
uid                  Po-Chuan Hsieh (sunpoet) <sunpoet@sunpoet.net>
sub   4096R/ADE9E203 2010-09-21
```

D.3.142. Li-Wen Hsu <lwhsu@FreeBSD.org>

```
pub   1024D/2897B228 2005-01-16
      Key fingerprint = B6F7 170A 6DC6 5D1A BD4B  D86A 416B 0E39 2897 B228
uid                  Li-wen Hsu <lwhsu@lwhsu.org>
uid                  Li-wen Hsu <lwhsu@lwhsu.ckefgisc.org>
uid                  Li-wen Hsu <lwhsu@lwhsu.csie.net>
uid                  Li-wen Hsu <lwhsu@ckefgisc.org>
uid                  Li-wen Hsu <lwhsu@csie.nctu.edu.tw>
uid                  Li-wen Hsu <lwhsu@ccca.nctu.edu.tw>
uid                  Li-wen Hsu <lwhsu@iis.sinica.edu.tw>
uid                  Li-wen Hsu <lwhsu@cs.nctu.edu.tw>
uid                  Li-Wen Hsu <lwhsu@FreeBSD.org>
sub   2048g/16F82238 2005-01-16
```

D.3.143. Howard F. Hu <foxfair@FreeBSD.org>

```
pub  1024D/4E9BCA59 2003-09-01 Foxfair Hu <foxfair@FreeBSD.org>
     Key fingerprint = 280C A846 CA1B CAC9 DDCF  F4CB D553 4BD5 4E9B CA59
uid                           Foxfair Hu <foxfair@drago.fomokka.net>
uid                           Howard Hu <howardhu@yahoo-inc.com>
sub  1024g/3356D8C1 2003-09-01
```

D.3.144. Chin-San Huang <chinsan@FreeBSD.org>

```
pub   1024D/350EECFA 2006-10-04
      Key fingerprint = 1C4D 0C9E 0E68 DB74 0688  CE43 D2A5 3F82 350E ECFA
uid                  Chin-San Huang (lab) <chinsan@chinsan2.twbbs.org>
uid                  Chin-San Huang (FreeBSD committer) <chinsan@FreeBSD.org>
uid                  Chin-San Huang (Gmail) <chinsan.tw@gmail.com>
sub   2048g/35F75A30 2006-10-04
```

D.3.145. Davide Italiano <davide@FreeBSD.org>

```
pub   2048R/4CB47484 2012-01-17
      Key fingerprint = B5C9 77F5 1E67 D110 8D19  7587 EB95 EA82 4CB4 7484
uid                  Davide Italiano <davide@FreeBSD.org>
sub   2048R/91F7443D 2012-01-17
```

D.3.146. Jordan K. Hubbard <jkh@FreeBSD.org>

```
pub  1024R/8E542D5D 1996-04-04 Jordan K. Hubbard <jkh@FreeBSD.org>
     Key fingerprint = 3C F2 27 7E 4A 6C 09 0A  4B C9 47 CD 4F 4D 0B 20
```

D.3.147. Konrad Jankowski <versus@FreeBSD.org>

```
pub   1024D/A01C218A 2008-10-28
      Key fingerprint = A805 21DC 859F E941 D2EA  9986 2264 8E5D A01C 218A
uid                  Konrad Jankowski <versus@freebsd.org>
sub   2048g/56AE1959 2008-10-28
```

D.3.148. Weongyo Jeong <weongyo@FreeBSD.org>

```
pub   1024D/22354D7A 2007-12-28
      Key fingerprint = 138E 7115 A86F AA40 B509  5883 B387 DCE9 2235 4D7A
uid                  Weongyo Jeong <weongyo.jeong@gmail.com>
uid                  Weongyo Jeong <weongyo@freebsd.org>
sub   2048g/9AE6DAEE 2007-12-28
```

D.3.149. Peter Jeremy <peterj@FreeBSD.org>

```
pub   1024D/F00FB887 2005-10-20
      Key fingerprint = 0BF7 7A72 5894 EBE6 4F4D  7EEE FE8A 47BF F00F B887
uid                  Peter Jeremy <peterjeremy@acm.org>
uid                  [jpeg image of size 4413]
uid                  Peter Jeremy <peter.jeremy@auug.org.au>
uid                  Peter Jeremy <peterjeremy@optusnet.com.au>
uid                  Peter Jeremy (preferred) <peter@rulingia.com>
uid                  Peter Jeremy <peterj@freebsd.org>
sub   2048g/7E0B423B 2005-10-20
```

D.3.150. Tatuya JINMEI <jinmei@FreeBSD.org>

```
pub   1024D/ABA82228 2002-08-15
      Key fingerprint = BB70 3050 EE39 BE00 48BB  A5F3 5892 F203 ABA8 2228
uid                  JINMEI Tatuya <jinmei@FreeBSD.org>
uid                  JINMEI Tatuya <jinmei@jinmei.org>
uid                  JINMEI Tatuya (the KAME project) <jinmei@isl.rdc.toshiba.co.jp>
sub   1024g/8B43CF66 2002-08-15
```

D.3.151. Michael Johnson <ahze@FreeBSD.org>

```
pub  1024D/3C046FD6 2004-10-29 Michael Johnson (FreeBSD key) <ahze@FreeBSD.org>
     Key fingerprint = 363C 6ABA ED24 C23B 5F0C  3AB4 9F8B AA7D 3C04 6FD6
uid                           Michael Johnson (pgp key) <ahze@ahze.net>
sub  2048g/FA334AE3 2004-10-29
```

D.3.152. Mark Johnston <markj@FreeBSD.org>

```
pub   2048R/80A62628 2012-12-19
      Key fingerprint = AFEF AD33 1C4E FFE5 141E  0157 05A4 DA8B 80A6 2628
uid                  Mark Johnston <markj@freebsd.org>
sub   2048R/47C7D3C2 2012-12-19
```

D.3.153. Trevor Johnson <trevor@FreeBSD.org>

```
pub  1024D/3A3EA137 2000-04-20 Trevor Johnson <trevor@jpj.net>
     Key fingerprint = 7ED1 5A92 76C1 FFCB E5E3  A998 F037 5A0B 3A3E A137
sub  1024g/46C24F1E 2000-04-20
```

D.3.154. Tom Judge <tj@FreeBSD.org>

```
pub   2048R/81E22216 2012-05-27 [expires: 2017-05-26]
      Key fingerprint = 8EF8 36C8 44A6 9576 6ADB  EB0E 4252 33DC 81E2 2216
uid                  Tom Judge <tom@tomjudge.com>
uid                  Tom Judge <tjudge@sourcefire.com>
uid                  Tom Judge <tj@freebsd.org>
sub   2048R/2CA4AA0D 2012-05-27 [expires: 2017-05-26]
```

D.3.155. Alexander Kabaev <kan@FreeBSD.org>

```
pub   1024D/C9BE5D96 2002-07-01
      Key fingerprint = 7474 A847 DBF5 50A5 FC3E  F223 43AC F58C C9BE 5D96
uid                  Alexander Kabaev <kabaev@gmail.com>
uid                  Alexander Kabaev (FreeBSD committer account ID) <kan@FreeBSD.ORG>
sub   1024g/534D9E06 2002-07-01
```

D.3.156. Benjamin Kaduk <bjk@FreeBSD.org>

```
pub   4096R/8302FE9F 2011-08-20 [expires: 2013-07-21]
      Key fingerprint = 9FD9 F966 D914 5101 BE59  FE13 2D29 EEED 8302 FE9F
uid                  Benjamin Kaduk <bjk@FreeBSD.org>
sub   4096R/28698ABE 2011-08-20 [expires: 2013-08-19]
```

D.3.157. Poul-Henning Kamp <phk@FreeBSD.org>

```
pub  1024R/0358FCBD 1995-08-01 Poul-Henning Kamp <phk@FreeBSD.org>
     Key fingerprint = A3 F3 88 28 2F 9B 99 A2  49 F4 E2 FA 5A 78 8B 3E
```

D.3.158. Sergey Kandaurov <pluknet@FreeBSD.org>

```
pub   2048R/10607419 2010-10-04
      Key fingerprint = 020B EC25 7E1F 8BC5 C42C  513B 3F4E 97BA 1060 7419
uid                  Sergey Kandaurov (freebsd) <pluknet@freebsd.org>
uid                  Sergey Kandaurov <pluknet@gmail.com>
sub   2048R/5711F73B 2010-10-04
```

D.3.159. Coleman Kane <cokane@FreeBSD.org>

```
pub   1024D/C5DAB797 2007-07-22
      Key fingerprint = FC09 F326 4318 E714 DE45  6CB0 70C4 B141 C5DA B797
uid                  Coleman Kane (Personal PGP Key) <cokane@cokane.org>
uid                  Coleman Kane (Personal PGP Key) <cokane@FreeBSD.org>
sub   2048g/5C680129 2007-07-22
```

D.3.160. Takenori KATO <kato@FreeBSD.org>

```
pub   4096R/3CF9ACE7 2012-10-02
      Key fingerprint = 5B72 AEF9 B2F9 069D 54FE  CF60 444F 91C8 3CF9 ACE7
uid                  KATO Takenori <kato@FreeBSD.org>
uid                  KATO Takenori <kato@nendai.nagoya-u.ac.jp>
sub   4096R/1C593356 2012-10-02
```

D.3.161. Josef Karthauser <joe@FreeBSD.org>

```
pub  1024D/E6B15016 2000-10-19 Josef Karthauser <joe@FreeBSD.org>
     Key fingerprint = 7266 8EAF 82C2 D439 5642  AC26 5D52 1C8C E6B1 5016
uid                            Josef Karthauser <joe@tao.org.uk>
uid                            Josef Karthauser <joe@uk.FreeBSD.org>
uid                            [revoked] Josef Karthauser <josef@bsdi.com>
uid                            [revoked] Josef Karthauser <joe@pavilion.net>
sub  2048g/1178B692 2000-10-19
```

D.3.162. Vinod Kashyap <vkashyap@FreeBSD.org>

```
pub  1024R/04FCCDD3 2004-02-19 Vinod Kashyap (gnupg key) <vkashyap@freebsd.org>
     Key fingerprint = 9B83 0B55 604F E491 B7D2  759D DF92 DAA0 04FC CDD3
```

D.3.163. Kris Kennaway <kris@FreeBSD.org>

```
pub  1024D/68E840A5 2000-01-14 Kris Kennaway <kris@citusc.usc.edu>
     Key fingerprint = E65D 0E7D 7E16 B212 1BD6  39EE 5ABC B405 68E8 40A5
uid                            Kris Kennaway <kris@FreeBSD.org>
```

```
uid                          Kris Kennaway <kris@obsecurity.org>
sub  2048g/03A41C45 2000-01-14 [expires: 2006-01-14]
```

D.3.164. Giorgos Keramidas <keramida@FreeBSD.org>

```
pub   1024D/318603B6 2001-09-21
      Key fingerprint = C1EB 0653 DB8B A557 3829  00F9 D60F 941A 3186 03B6
uid                  Giorgos Keramidas <keramida@FreeBSD.org>
uid                  Giorgos Keramidas <keramida@ceid.upatras.gr>
uid                  Giorgos Keramidas <keramida@hellug.gr>
uid                  Giorgos Keramidas <keramida@linux.gr>
uid                  Giorgos Keramidas <gkeramidas@gmail.com>
sub   1024g/50FDBAD1 2001-09-21
```

D.3.165. Max Khon <fjoe@FreeBSD.org>

```
pub   1024D/6B87E212 2009-02-17
      Key fingerprint = 124D EC6C 6365 D41A 497A  9C3E FCF3 8708 6B87 E212
uid                  Max Khon <fjoe@FreeBSD.org>
uid                  Max Khon <fjoe@samodelkin.net>
sub   2048g/CB71491D 2009-02-17
```

D.3.166. Manolis Kiagias <manolis@FreeBSD.org>

```
pub   1024D/6E0FB494 2006-08-22
      Key fingerprint = F820 5AAF 7112 2CDD 23D8  3BDF 67F3 311A 6E0F B494
uid                  Manolis Kiagias <manolis@FreeBSD.org>
uid                  Manolis Kiagias <sonicy@otenet.gr>
uid                  Manolis Kiagias (A.K.A. sonic, sonicy, sonic2000gr) <sonic@diktia.dyndns.org
sub   2048g/EB94B411 2006-08-22
```

D.3.167. Jung-uk Kim <jkim@FreeBSD.org>

```
pub   2048R/D932A1CE 2012-11-19
      Key fingerprint = 2202 B5FB 78B7 A303 4919  B7C7 25E9 69B1 D932 A1CE
uid                  Jung-uk Kim <jkim@FreeBSD.org>
sub   2048R/41858FC6 2012-11-19
```

D.3.168. Zack Kirsch <zack@FreeBSD.org>

```
pub  1024D/1A725562 2010-11-05 Zack Kirsch <zack@freebsd.org>
     Key fingerprint = A8CC AA5E FB47 A386 E757  A2B8 BDD2 0684 1A72 5562
sub  1024g/6BFE2C06 2010-11-05
```

D.3.169. Jakub Klama <jceel@FreeBSD.org>

```
pub   2048R/2AAEA67D 2011-09-27
      Key fingerprint = 40D6 097A 174F 511B 80EB  F3A3 0946 4193 2AAE A67D
uid                  Jakub Klama <jceel@FreeBSD.org>
sub   2048R/5291BC4D 2011-09-27
```

D.3.170. Andreas Klemm <andreas@FreeBSD.org>

```
pub  1024D/6C6F6CBA 2001-01-06 Andreas Klemm <andreas.klemm@eu.didata.com>
     Key fingerprint = F028 D51A 0D42 DD67 4109  19A3 777A 3E94 6C6F 6CBA
uid                            Andreas Klemm <andreas@klemm.gtn.com>
uid                            Andreas Klemm <andreas@FreeBSD.org>
uid                            Andreas Klemm <andreas@apsfilter.org>
sub  2048g/FE23F866 2001-01-06
```

D.3.171. Johann Kois <jkois@FreeBSD.org>

```
pub  1024D/DD61C2D8 2004-06-27 Johann Kois <J.Kois@web.de>
     Key fingerprint = 8B70 03DB 3C45 E71D 0ED4  4825 FEB0 EBEF DD61 C2D8
uid                            Johann Kois <jkois@freebsd.org>
sub  1024g/568307CB 2004-06-27
```

D.3.172. Sergei Kolobov <sergei@FreeBSD.org>

```
pub  1024D/3BA53401 2003-10-10 Sergei Kolobov <sergei@FreeBSD.org>
     Key fingerprint = A2F4 5F34 0586 CC9C 493A  347C 14EC 6E69 3BA5 3401
uid                            Sergei Kolobov <sergei@kolobov.com>
sub  2048g/F8243671 2003-10-10
```

D.3.173. Maxim Konovalov <maxim@FreeBSD.org>

```
pub  1024D/2C172083 2002-05-21 Maxim Konovalov <maxim@FreeBSD.org>
     Key fingerprint = 6550 6C02 EFC2 50F1 B7A3  D694 ECF0 E90B 2C17 2083
uid                            Maxim Konovalov <maxim@macomnet.ru>
sub  1024g/F305DDCA 2002-05-21
```

D.3.174. Taras Korenko <taras@FreeBSD.org>

```
pub   1024D/8ACCC68B 2010-03-30
      Key fingerprint = 5128 2A8B 9BC1 A664 21E0  1E61 D838 54D3 8ACC C68B
uid                   Taras Korenko <taras@freebsd.org>
uid                   Taras Korenko <ds@ukrhub.net>
uid                   Taras Korenko <tarasishche@gmail.com>
```

```
sub   2048g/8D7CC0FA 2010-03-30 [expires: 2015-03-29]
```

D.3.175. Joseph Koshy <jkoshy@FreeBSD.org>

```
pub  1024D/D93798B6 2001-12-21 Joseph Koshy (FreeBSD) <jkoshy@freebsd.org>
     Key fingerprint = 0DE3 62F3 EF24 939F 62AA  2E3D ABB8 6ED3 D937 98B6
sub  1024g/43FD68E9 2001-12-21
```

D.3.176. Wojciech A. Koszek <wkoszek@FreeBSD.org>

```
pub   1024D/C9F25145 2006-02-15
      Key fingerprint = 6E56 C571 9D33 D23E 9A61  8E50 623C AD62 C9F2 5145
uid                  Wojciech A. Koszek <dunstan@FreeBSD.czest.pl>
uid                  Wojciech A. Koszek <wkoszek@FreeBSD.org>
sub   4096g/3BBD20A5 2006-02-15
```

D.3.177. Alex Kozlov <ak@FreeBSD.org>

```
pub   2048R/0D1D29A0 2012-03-01 [expires: 2024-02-27]
      Key fingerprint = 7774 4FCF 6AC9 126B BD0E  DBF3 5EBF 4968 0D1D 29A0
uid                  Alex Kozlov <ak@freebsd.org>
sub   2048R/2DD82C65 2012-03-01 [expires: 2024-02-27]
```

D.3.178. Steven Kreuzer <skreuzer@FreeBSD.org>

```
pub   1024D/E0D6F907 2009-03-16 [expires: 2013-04-25]
      Key fingerprint = 8D8F 14D6 ED9F 6BD0 7756  7A46 66BA B4B6 E0D6 F907
uid                  Steven Kreuzer <skreuzer@exit2shell.com>
uid                  Steven Kreuzer <skreuzer@freebsd.org>
```

D.3.179. Gábor Kövesdán <gabor@FreeBSD.org>

```
pub   1024D/2373A6B1 2006-12-05
      Key fingerprint = A42A 10D6 834B BEC0 26F0  29B1 902D D04F 2373 A6B1
uid                  Gabor Kovesdan <gabor@FreeBSD.org>
sub   2048g/92B0A104 2006-12-05
```

D.3.180. Ana Kukec <anchie@FreeBSD.org>

```
pub   2048R/510D23BB 2010-04-18
      Key fingerprint = 0A9B 0ABB 0E1C B5A4 3408  398F 778A C3B4 510D 23BB
uid                  Ana Kukec <anchie@FreeBSD.org>
```

```
sub   2048R/699E4DDA 2010-04-18
```

D.3.181. Roman Kurakin <rik@FreeBSD.org>

```
pub   1024D/C8550F4C 2005-12-16 [expires: 2008-12-15]
      Key fingerprint = 25BB 789A 6E07 E654 8E59  0FA9 42B1 937C C855 0F4C
uid                  Roman Kurakin <rik@FreeBSD.org>
sub   2048g/D15F2AB6 2005-12-16 [expires: 2008-12-15]
```

D.3.182. Hideyuki KURASHINA <rushani@FreeBSD.org>

```
pub  1024D/439ADC57 2002-03-22 Hideyuki KURASHINA <rushani@bl.mmtr.or.jp>
     Key fingerprint = A052 6F98 6146 6FE3 91E2  DA6B F2FA 2088 439A DC57
uid                            Hideyuki KURASHINA <rushani@FreeBSD.org>
uid                            Hideyuki KURASHINA <rushani@jp.FreeBSD.org>
sub  1024g/64764D16 2002-03-22
```

D.3.183. Jun Kuriyama <kuriyama@FreeBSD.org>

```
pub  1024D/FE3B59CD 1998-11-23 Jun Kuriyama <kuriyama@imgsrc.co.jp>
     Key fingerprint = 5219 55CE AC84 C296 3A3B  B076 EE3C 4DBB FE3B 59CD
uid                            Jun Kuriyama <kuriyama@FreeBSD.org>
uid                            Jun Kuriyama <kuriyama@jp.FreeBSD.org>
sub  2048g/1CF20D27 1998-11-23
```

D.3.184. René Ladan <rene@FreeBSD.org>

```
pub   4096R/0A3789B7 2012-11-18
      Key fingerprint = 101A 716B 162B 00E5 5BED  EA05 ADBB F861 0A37 89B7
uid                  René Ladan <rene@freebsd.org>
sub   4096R/B67184C6 2012-11-18
```

D.3.185. Julien Laffaye <jlaffaye@FreeBSD.org>

```
pub   2048R/6AEBE420 2011-06-06
      Key fingerprint = 031A B449 B383 5C3B B618  E2F4 BAD0 0F0E 6AEB E420
uid                  Julien Laffaye <jlaffaye@FreeBSD.org>
sub   2048R/538B8D5B 2011-06-06
```

D.3.186. Clement Laforet <clement@FreeBSD.org>

```
pub  1024D/0723BA1D 2003-12-13 Clement Laforet (FreeBSD committer address) <clement@FreeBSD.org>
     Key fingerprint = 3638 4B14 8463 A67B DC7E  641C B118 5F8F 0723 BA1D
uid                           Clement Laforet <sheepkiller@cultdeadsheep.org>
uid                           Clement Laforet <clement.laforet@cotds.org>
sub  2048g/23D57658 2003-12-13
```

D.3.187. Max Laier <mlaier@FreeBSD.org>

```
pub   1024D/3EB6046D 2004-02-09
      Key fingerprint = 917E 7F25 E90F 77A4 F746  2E8D 5F2C 84A1 3EB6 046D
uid                  Max Laier <max@love2party.net>
uid                  Max Laier <max.laier@ira.uka.de>
uid                  Max Laier <mlaier@freebsd.org>
uid                  Max Laier <max.laier@tm.uka.de>
sub   4096g/EDD08B9B 2005-06-28
```

D.3.188. Erwin Lansing <erwin@FreeBSD.org>

```
pub   1024D/15256990 1998-07-03
      Key fingerprint = FB58 9797 299A F18E 2D3E  73D6 AB2F 5A5B 1525 6990
uid                  Erwin Lansing <erwin@lansing.dk>
uid                  Erwin Lansing <erwin@FreeBSD.org>
uid                  Erwin Lansing <erwin@droso.dk>
uid                  Erwin Lansing <erwin@droso.org>
uid                  Erwin Lansing <erwin@aauug.dk>
sub   2048g/7C64013D 1998-07-03
```

D.3.189. Ganael Laplanche <martymac@FreeBSD.org>

```
pub   1024D/10B87391 2006-01-13
      Key fingerprint = D59D 984D 8988 7BB9 DA37  BA77 757E D5F0 10B8 7391
uid                  Ganael LAPLANCHE <ganael.laplanche@martymac.org>
uid                  Ganael LAPLANCHE <martymac@martymac.com>
uid                  Ganael LAPLANCHE <ganael.laplanche@martymac.com>
uid                  Ganael LAPLANCHE <martymac@martymac.org>
uid                  Ganael LAPLANCHE <martymac@pasteur.fr>
uid                  Ganael LAPLANCHE <ganael.laplanche@pasteur.fr>
uid                  Ganael LAPLANCHE <martymac@FreeBSD.org>
sub   2048g/D65069D5 2006-01-13
```

D.3.190. Greg Larkin <glarkin@FreeBSD.org>

```
pub   1024D/1C940290 2003-10-09
      Key fingerprint = 8A4A 80AA F26C 8C2C D01B  94C6 D2C4 68B8 1C94 0290
uid                  Greg Larkin (The FreeBSD Project) <glarkin@FreeBSD.org>
uid                  Gregory C. Larkin (SourceHosting.Net, LLC) <glarkin@sourcehosting.net>
uid                  [jpeg image of size 6695]
sub   2048g/47674316 2003-10-09
```

D.3.191. Frank J. Laszlo <laszlof@FreeBSD.org>

```
pub   4096R/012360EC 2006-11-06 [expires: 2011-11-05]
      Key fingerprint = 3D93 21DB B5CC 1339 E4B4  1BC4 AD50 C17C 0123 60EC
uid                  Frank J. Laszlo <laszlof@FreeBSD.org>
```

D.3.192. Dru Lavigne <dru@FreeBSD.org>

```
pub   1024D/C6AA2E94 2013-01-22
      Key fingerprint = 6CC4 2180 F27C 29B6 5A9C  EC0D A454 DC05 C6AA 2E94
uid                  Dru Lavigne <dru@freebsd.org>
sub   1024g/7FAC82EA 2013-01-22
```

D.3.193. Sam Lawrance <lawrance@FreeBSD.org>

```
pub   1024D/32708C59 2003-08-14
      Key fingerprint = 1056 2A02 5247 64D4 538D  6975 8851 7134 3270 8C59
uid                  Sam Lawrance <lawrance@FreeBSD.org>
uid                  Sam Lawrance <boris@brooknet.com.au>
sub   2048g/0F9CCF92 2003-08-14
```

D.3.194. Nate Lawson <njl@FreeBSD.org>

```
pub   1024D/60E5AC11 2007-02-07
      Key fingerprint = 18E2 7E5A FD6A 199B B08B  E9FB 73C8 DB67 60E5 AC11
uid                  Nate Lawson <nate@root.org>
sub   2048g/CDBC7E1B 2007-02-07
```

D.3.195. Jeremie Le Hen <jlh@FreeBSD.org>

```
pub   2048D/8BF6CF92 2012-04-18
      Key fingerprint = 66C9 B361 16CA BFF6 5C07  DA0A 28DE 3702 8BF6 CF92
uid                  Jeremie Le Hen <jeremie@le-hen.org>
uid                  Jeremie Le Hen <jeremie@lehen.org>
uid                  Jeremie Le Hen <ttz@chchile.org>
```

```
uid                  Jeremie Le Hen <jlh@FreeBSD.org>
sub   2048g/045479A3 2012-04-18
```

D.3.196. Yen-Ming Lee <leeym@FreeBSD.org>

```
pub   1024D/93FA8BD6 2007-05-21
      Key fingerprint = DEC4 6E7F 69C0 4AC3 21ED  EE65 6C0E 9257 93FA 8BD6
uid                  Yen-Ming Lee <leeym@leeym.com>
sub   2048g/899A3931 2007-05-21
```

D.3.197. Sam Leffler <sam@FreeBSD.org>

```
pub   1024D/BD147743 2005-03-28
      Key fingerprint = F618 F2FC 176B D201 D91C  67C6 2E33 A957 BD14 7743
uid                  Samuel J. Leffler <sam@freebsd.org>
sub   2048g/8BA91D05 2005-03-28
```

D.3.198. Jean-Yves Lefort <jylefort@FreeBSD.org>

```
pub   1024D/A3B8006A 2002-09-07
      Key fingerprint = CC99 D1B0 8E44 293D 32F7  D92E CB30 FB51 A3B8 006A
uid                  Jean-Yves Lefort <jylefort@FreeBSD.org>
uid                  Jean-Yves Lefort <jylefort@brutele.be>
sub   4096g/C9271AFC 2002-09-07
```

D.3.199. Alexander Leidinger <netchild@FreeBSD.org>

```
pub   1024D/72077137 2002-01-31
      Key fingerprint = AA3A 8F69 B214 6BBD 5E73  C9A0 C604 3C56 7207 7137
uid                  Alexander Leidinger <netchild@FreeBSD.org>
uid                  [jpeg image of size 19667]
sub   2048g/8C9828D3 2002-01-31
```

D.3.200. Andrey V. Elsukov <ae@FreeBSD.org>

```
pub   2048R/10C8A17A 2010-05-29
      Key fingerprint = E659 1E1B 41DA 1516 F0C9  BC00 01C5 EA04 10C8 A17A
uid                  Andrey V. Elsukov <ae@freebsd.org>
uid                  Andrey V. Elsukov <bu7cher@yandex.ru>
sub   2048R/0F6D64C5 2010-05-29
```

D.3.201. Dejan Lesjak <lesi@FreeBSD.org>

```
pub  1024D/96C5221F 2004-08-18 Dejan Lesjak <lesi@FreeBSD.org>
     Key fingerprint = 2C5C 02EA 1060 1D6D 9982  38C0 1DA7 DBC4 96C5 221F
uid                            Dejan Lesjak <dejan.lesjak@ijs.si>
sub  1024g/E0A69278 2004-08-18
```

D.3.202. Achim Leubner <achim@FreeBSD.org>

```
pub   2048R/2E15B3C1 2013-01-22
      Key fingerprint = 2A48 0317 D477 2A07 2AD9  CF1C 7C1D 832E 2E15 B3C1
uid                  Achim Leubner <achim@freebsd.org>
sub   2048R/E275EF01 2013-01-22
```

D.3.203. Chuck Lever <cel@FreeBSD.org>

```
pub   1024D/8FFC2B87 2006-02-13
      Key fingerprint = 6872 923F 5012 F88B 394C  2F69 37B4 8171 8FFC 2B87
uid                  Charles E. Lever <cel@freebsd.org>
sub   2048g/9BCE0459 2006-02-13
```

D.3.204. Greg Lewis <glewis@FreeBSD.org>

```
pub  1024D/1BB6D9E0 2002-03-05 Greg Lewis (FreeBSD) <glewis@FreeBSD.org>
     Key fingerprint = 2410 DA6D 5A3C D801 65FE  C8DB DEEA 9923 1BB6 D9E0
uid                            Greg Lewis <glewis@eyesbeyond.com>
sub  2048g/45E67D60 2002-03-05
```

D.3.205. Qing Li <qingli@FreeBSD.org>

```
pub   2048R/A3CA4C13 2013-06-12 [expires: 2017-06-12]
      Key fingerprint = E37B CB18 35D1 F01B 7D7B  1000 0EAF 4BEA A3CA 4C13
uid                  Qing Li <qingli@freebsd.org>
sub   2048R/EF3A9370 2013-06-12 [expires: 2017-06-12]
```

D.3.206. Xin Li <delphij@FreeBSD.org>

```
pub   1024D/CAEEB8C0 2004-01-28
      Key fingerprint = 43B8 B703 B8DD 0231 B333  DC28 39FB 93A0 CAEE B8C0
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@frontfree.net>
uid                  Xin LI <delphij@delphij.net>
uid                  Xin LI <delphij@geekcn.org>
```

```
pub   1024D/42EA8A4B 2006-01-27 [expired: 2008-01-01]
      Key fingerprint = F19C 2616 FA97 9C13 2581  C6F3 85C5 1CCE 42EA 8A4B
uid                  Xin LI <delphij@geekcn.org>
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@delphij.net>

pub   1024D/18EDEBA0 2008-01-02 [expired: 2010-01-02]
      Key fingerprint = 79A6 CF42 F917 DDCA F1C2  C926 8BEB DB04 18ED EBA0
uid                  Xin LI <delphij@geekcn.org>
uid                  Xin LI <delphij@FreeBSD.org>
uid                  Xin LI <delphij@delphij.net>

pub   2048R/3FCA37C1 2010-01-10 [expired: 2012-01-10]
      Key fingerprint = 27EA 5D6C 9398 BA7F B205  8F70 04CE F812 3FCA 37C1
uid                  Xin LI <delphij@delphij.net>
uid                  Xin LI <delphij@gmail.com>
uid                  Xin LI <delphij@geekcn.org>
uid                  Xin LI <delphij@FreeBSD.org>

pub   4096R/2E54AB2C 2011-12-05
      Key fingerprint = D95C D3C3 8FA8 25C2 C62B  9FEA 0887 6D93 2E54 AB2C
uid                  Xin Li <delphij@geekcn.org>
uid                  Xin Li <delphij@delphij.net>
uid                  Xin Li <delphij@FreeBSD.org>
sub   4096R/7832B740 2011-12-05
sub   2048R/BC50FBB3 2011-12-05 [expires: 2013-12-05]
sub   2048R/C894647D 2011-12-05 [expires: 2013-12-05]
```

D.3.207. Tai-hwa Liang <avatar@FreeBSD.org>

```
pub  1024R/F4013AB1 1998-05-13 Tai-hwa Liang <avatar@FreeBSD.org>
     Key fingerprint = 5B 05 1D 37 7F 35 31 4E  5D 38 BD 07 10 32 B9 D0
uid                           Tai-hwa Liang <avatar@mmlab.cse.yzu.edu.tw>
```

D.3.208. Ying-Chieh Liao <ijliao@FreeBSD.org>

```
pub  1024D/11C02382 2001-01-09 Ying-Chieh Liao <ijliao@CCCA.NCTU.edu.tw>
     Key fingerprint = 4E98 55CC 2866 7A90 EFD7  9DA5 ACC6 0165 11C0 2382
uid                            Ying-Chieh Liao <ijliao@FreeBSD.org>
uid                            Ying-Chieh Liao <ijliao@csie.nctu.edu.tw>
uid                            Ying-Chieh Liao <ijliao@dragon2.net>
uid                            Ying-Chieh Liao <ijliao@tw.FreeBSD.org>
sub  4096g/C1E16E89 2001-01-09
```

D.3.209. Ulf Lilleengen <lulf@FreeBSD.org>

```
pub   1024D/ADE1B837 2009-08-19 [expires: 2014-08-18]
      Key fingerprint = 3822 B4E6 6D1C 6F71 4AA8  7A27 ADDF C400 ADE1 B837
uid                  Ulf Lilleengen <ulf.lilleengen@gmail.com>
uid                  Ulf Lilleengen <lulf@pvv.ntnu.no>
uid                  Ulf Lilleengen <lulf@stud.ntnu.no>
uid                  Ulf Lilleengen <lulf@FreeBSD.org>
uid                  Ulf Lilleengen <lulf@idi.ntnu.no>
sub   2048g/B5409122 2009-08-19 [expires: 2014-08-18]
```

D.3.210. Clive Lin <clive@FreeBSD.org>

```
pub  1024D/A008C03E 2001-07-30 Clive Lin <clive@tongi.org>
     Key fingerprint = FA3F 20B6 A77A 6CEC 1856  09B0 7455 2805 A008 C03E
uid                           Clive Lin <clive@CirX.ORG>
uid                           Clive Lin <clive@FreeBSD.org>
sub  1024g/03C2DC87 2001-07-30 [expires: 2005-08-25]
```

D.3.211. Po-Chien Lin <pclin@FreeBSD.org>

```
pub   4096R/865C427F 2013-02-05
      Key fingerprint = CF3B AB13 4C94 6388 B047  B599 8B28 1692 865C 427F
uid                  Po-Chien Lin <pclin@FreeBSD.org>
uid                  Po-Chien Lin <linpc@cs.nctu.edu.tw>
sub   4096R/F31280BA 2013-02-05
```

D.3.212. Yi-Jheng Lin <yzlin@FreeBSD.org>

```
pub   2048R/A34C6A8A 2009-07-20
      Key fingerprint = 7E3A E981 BB7C 5D73 9534  ED39 0222 04D3 A34C 6A8A
uid                  Yi-Jheng Lin (FreeBSD) <yzlin@FreeBSD.org>
sub   2048R/B4D776FE 2009-07-20
```

D.3.213. Mark Linimon <linimon@FreeBSD.org>

```
pub   1024D/84C83473 2003-10-09
      Key fingerprint = 8D43 1B55 D127 0BFC 842E  1C96 803C 5A34 84C8 3473
uid                  Mark Linimon <linimon@FreeBSD.org>
uid                  Mark Linimon <linimon@lonesome.com>
sub   1024g/24BFF840 2003-10-09
```

D.3.214. Tilman Keskinöz <arved@FreeBSD.org>

```
pub   1024D/807AC53A 2002-06-03 [expires: 2013-09-07]
      Key fingerprint = A92F 344F 31A8 B8DE DDFA  7FB4 7C22 C39F 807A C53A
uid                  Tilman Keskinöz <arved@arved.at>
uid                  Tilman Keskinöz <arved@FreeBSD.org>
sub   1024g/FA351986 2002-06-03 [expires: 2013-09-07]
```

D.3.215. Dryice Liu <dryice@FreeBSD.org>

```
pub   1024D/77B67874 2005-01-28
      Key fingerprint = 8D7C F82D D28D 07E5 EF7F  CD25 6B5B 78A8 77B6 7874
uid                  Dryice Dong Liu (Dryice) <dryice@FreeBSD.org>
uid                  Dryice Dong Liu (Dryice) <dryice@liu.com.cn>
uid                  Dryice Dong Liu (Dryice) <dryice@hotpop.com>
uid                  Dryice Dong Liu (Dryice) <dryiceliu@gmail.com>
uid                  Dryice Dong Liu (Dryice) <dryice@dryice.name>
sub   2048g/ECFA49E4 2005-01-28
```

D.3.216. Tong Liu <nemoliu@FreeBSD.org>

```
pub   1024D/ECC7C907 2007-07-10
      Key fingerprint = B62E 3109 896B B283 E2FA  60FE A1BA F92E ECC7 C907
uid                  Tong LIU <nemoliu@FreeBSD.org>
sub   4096g/B6D7B15D 2007-07-10
```

D.3.217. Zachary Loafman <zml@FreeBSD.org>

```
pub   1024D/4D65492D 2009-05-26
      Key fingerprint = E513 4AE9 5D6D 8BF9 1CD3  4389 4860 D79B 4D65 492D
uid                  Zachary Loafman <zml@FreeBSD.org>
sub   2048g/1AD659F0 2009-05-26
```

D.3.218. Juergen Lock <nox@FreeBSD.org>

```
pub   1024D/1B6BFBFD 2006-12-22
      Key fingerprint = 33A7 7FAE 51AF 00BC F0D3  ECCE FAFD 34C1 1B6B FBFD
uid                  Juergen Lock <nox@FreeBSD.org>
sub   2048g/251229D1 2006-12-22
```

D.3.219. Remko Lodder <remko@FreeBSD.org>

```
pub   4096R/3F774079 2012-11-11 [expires: 2016-11-11]
      Key fingerprint = 7EE4 C4AF DCA3 E0B4 479B  A344 7135 8ED6 3F77 4079
uid                  Remko Lodder <remko@FreeBSD.org>
sub   4096R/59F38CB0 2012-11-11 [expires: 2016-11-11]
```

D.3.220. Alexander Logvinov <avl@FreeBSD.org>

```
pub   1024D/1C47D5C0 2009-05-28
      Key fingerprint = 8B5F 880A 382B 075E E707  9DB2 E135 4176 1C47 D5C0
uid                  Alexander Logvinov <alexander@logvinov.com>
uid                  Alexander Logvinov (FreeBSD Ports Committer) <avl@FreeBSD.org>
uid                  Alexander Logvinov <ports@logvinov.com>
uid                  Alexander Logvinov <logvinov@gmail.com>
uid                  Alexander Logvinov <logvinov@yandex.ru>
sub   2048g/60BDD4BB 2009-05-28
```

D.3.221. Isabell Long <issyl0@FreeBSD.org>

```
pub   4096R/EB83C2BD 2009-09-26
      Key fingerprint = D55A 42E7 0974 EFD9 3939  56B9 6E6B E425 EB83 C2BD
uid                  Isabell Long <isabell@issyl0.co.uk>
uid                  Isabell Long <me@issyl0.co.uk>
uid                  Isabell Long <isabell121@gmail.com>
uid                  Isabell Long (BitFolk Ltd.) <isabell@bitfolk.com>
uid                  Isabell Long (College) <IL18685@woking.ac.uk>
uid                  Isabell Long (The Open University) <il948@my.open.ac.uk>
uid                  Isabell Long (Mailing lists address.) <lists@issyl0.co.uk>
uid                  Isabell Long (YRS) <isabell@youngrewiredstate.org>
uid                  Isabell Long (FreeBSD) <issyl0@FreeBSD.org>
```

D.3.222. Scott Long <scottl@FreeBSD.org>

```
pub  1024D/017C5EBF 2003-01-18 Scott A. Long (This is my official FreeBSD key) <scottl@freebsd.org
     Key fingerprint = 34EA BD06 44F7 F8C3 22BC  B52C 1D3A F6D1 017C 5EBF
sub  1024g/F61C8F91 2003-01-18
```

D.3.223. Rick Macklem <rmacklem@FreeBSD.org>

```
pub   1024D/7FB9C5F1 2009-04-05
      Key fingerprint = B9EA 767A F6F3 3786 E0C7  434A 05C6 70D6 7FB9 C5F1
uid                  Rick Macklem <rmacklem@freebsd.org>
sub   1024g/D0B20E8A 2009-04-05
```

D.3.224. Bruce A. Mah <bmah@FreeBSD.org>

```
pub   1024D/5BA052C3 1997-12-08
      Key fingerprint = F829 B805 207D 14C7 7197  7832 D8CA 3171 5BA0 52C3
uid                  Bruce A. Mah <bmah@acm.org>
uid                  Bruce A. Mah <bmah@ca.sandia.gov>
uid                  Bruce A. Mah <bmah@ieee.org>
uid                  Bruce A. Mah <bmah@cisco.com>
uid                  Bruce A. Mah <bmah@employees.org>
uid                  Bruce A. Mah <bmah@freebsd.org>
uid                  Bruce A. Mah <bmah@packetdesign.com>
uid                  Bruce A. Mah <bmah@kitchenlab.org>
sub   2048g/B4E60EA1 1997-12-08
```

D.3.225. Ruslan Makhmatkhanov <rm@FreeBSD.org>

```
pub   2048R/F60D756F 2011-11-10
      Key fingerprint = 9D18 8A88 304C B78B 8003  0379 4574 0BAF F60D 756F
uid                  Ruslan Makhmatkhanov <rm@FreeBSD.org>
sub   2048R/B658C269 2011-11-10
```

D.3.226. Mike Makonnen <mtm@FreeBSD.org>

```
pub  1024D/7CD41F55 2004-02-06 Michael Telahun Makonnen <mtm@FreeBSD.Org>
     Key fingerprint = AC7B 5672 2D11 F4D0 EBF8  5279 5359 2B82 7CD4 1F55
uid                           Michael Telahun Makonnen <mtm@tmsa-inc.com>
uid                           Mike Makonnen <mtm@identd.net>
uid                           Michael Telahun Makonnen <mtm@acs-et.com>
sub  2048g/E7DC936B 2004-02-06
```

D.3.227. David Malone <dwmalone@FreeBSD.org>

```
pub   512/40378991 1994/04/21 David Malone <dwmalone@maths.tcd.ie>
            Key fingerprint = 86 A7 F4 86 39 2C 47 2C  C1 C2 35 78 8E 2F B8 F5
```

D.3.228. Dmitry Marakasov <amdmi3@FreeBSD.org>

```
pub   1024D/F9D2F77D 2008-06-15 [expires: 2010-06-15]
      Key fingerprint = 55B5 0596 FF1E 8D84 5F56  9510 D35A 80DD F9D2 F77D
uid                  Dmitry Marakasov <amdmi3@amdmi3.ru>
uid                  Dmitry Marakasov <amdmi3@FreeBSD.org>
sub   2048g/2042CDD8 2008-06-15
```

D.3.229. John Marino <marino@FreeBSD.org>

```
pub   2048R/A0AE6229 2011-07-19
      Key fingerprint = EE48 4F90 C861 3A5F E39E  AB9E 33CF 4190 A0AE 6229
uid                  John Marino (DragonFly) <draco@marino.st>
uid                  John R. Marino <john.secure@marino.st>
uid                  John Marino (NetBSD) <marino@netbsd.org>
sub   2048R/71D9FB68 2011-07-19
```

D.3.230. Koop Mast <kwm@FreeBSD.org>

```
pub  1024D/F95426DA 2004-09-10 Koop Mast <kwm@rainbow-runner.nl>
     Key fingerprint = C66F 1835 0548 3440 8576  0FFE 6879 B7CD F954 26DA
uid                           Koop Mast <kwm@FreeBSD.org>
sub  1024g/A782EEDD 2004-09-10
```

D.3.231. Ed Maste <emaste@FreeBSD.org>

```
pub   2048R/50A17BF4 2012-12-18
      Key fingerprint = 0C08 ECC9 3A0A 8500 AB95  B553 49C4 7851 50A1 7BF4
uid                  Ed Maste <emaste@freebsd.org>
sub   2048R/08FA5F72 2012-12-18
```

D.3.232. Cherry G. Mathew <cherry@FreeBSD.org>

```
pub   2048R/2D066FE1 2007-05-22
      Key fingerprint = FBF1 89FF 81BB E1C7 6C1B  378D 3438 20E9 2D06 6FE1
uid                  Cherry G. Mathew (FreeBSD email) <cherry@FreeBSD.org>
uid                  "Cherry G. Mathew" (NetBSD email) <cherry@NetBSD.org>
sub   2048R/7B2C4166 2007-05-22
```

D.3.233. Makoto Matsushita <matusita@FreeBSD.org>

```
pub   1024D/20544576 1999-04-18
      Key fingerprint = 71B6 13BF B262 2DD8 2B7C  6CD0 EB2D 4147 2054 4576
uid                  Makoto Matsushita <matusita@matatabi.or.jp>
uid                  Makoto Matsushita <matusita@FreeBSD.org>
uid                  Makoto Matsushita <matusita@jp.FreeBSD.ORG>
uid                  Makoto Matsushita <matusita@ist.osaka-u.ac.jp>
sub   1024g/F1F3C94D 1999-04-18
```

D.3.234. Martin Matuska <mm@FreeBSD.org>

```
pub   1024D/4261B0D1 2007-02-05
      Key fingerprint = 17C4 3F32 B3DE 3ED7 E84E  5592 A76B 8B03 4261 B0D1
uid                  Martin Matuska <martin@matuska.org>
uid                  Martin Matuska <mm@FreeBSD.org>
uid                  Martin Matuska <martin.matuska@wu-wien.ac.at>
sub   2048g/3AC9A5A6 2007-02-05
```

D.3.235. Sergey Matveychuk <sem@FreeBSD.org>

```
pub   1024D/B71F605D 1999-10-13
      Key fingerprint = 4704 F374 DB28 BEC6 51C8  1322 4DC9 4BD8 B71F 605D
uid                  Sergey Matveychuk <sem@FreeBSD.org>
uid                  Sergey Matveychuk <sem@ciam.ru>
uid                  Sergey Matveychuk <sem@core.inec.ru>
sub   2048g/DEAF9D91 1999-10-13
```

D.3.236. Tom McLaughlin <tmclaugh@FreeBSD.org>

```
pub   1024D/E2F7B3D8 2005-05-24
      Key fingerprint = 7692 B222 8D23 CF94 1993  0138 E339 E225 E2F7 B3D8
uid                  Tom McLaughlin (Personal email address) <tmclaugh@sdf.lonestar.org>
uid                  Tom McLaughlin (Work email address) <tmclaughlin@meditech.com>
uid                  Tom McLaughlin (FreeBSD email address) <tmclaugh@FreeBSD.org>
sub   2048g/16838F62 2005-05-24
```

D.3.237. Jean Milanez Melo <jmelo@FreeBSD.org>

```
pub   1024D/AA5114BF 2006-03-03
      Key fingerprint = 826D C2AA 6CF2 E29A EBE7  4776 D38A AB83 AA51 14BF
uid                  Jean Milanez Melo <jmelo@FreeBSD.org>
uid                  Jean Milanez Melo <jmelo@freebsdbrasil.com.br>
sub   4096g/E9E1CBD9 2006-03-03
```

D.3.238. Kenneth D. Merry <ken@FreeBSD.org>

```
pub  1024D/54C745B5 2000-05-15 Kenneth D. Merry <ken@FreeBSD.org>
     Key fingerprint = D25E EBC5 F17A 9E52 84B4  BF14 9248 F0DA 54C7 45B5
uid                           Kenneth D. Merry <ken@kdm.org>
sub  2048g/89D0F797 2000-05-15

pub  1024R/2FA0A505 1995-10-30 Kenneth D. Merry <ken@plutotech.com>
     Key fingerprint = FD FA 85 85 95 C4 8E E8  98 1A CA 18 56 F0 00 1F
```

D.3.239. Dirk Meyer <dinoex@FreeBSD.org>

```
pub  1024R/331CDA5D 1995-06-04 Dirk Meyer <dinoex@FreeBSD.org>
     Key fingerprint = 44 16 EC 0A D3 3A 4F 28  8A 8A 47 93 F1 CF 2F 12
uid                            Dirk Meyer <dirk.meyer@dinoex.sub.org>
uid                            Dirk Meyer <dirk.meyer@guug.de>
```

D.3.240. Yoshiro Sanpei MIHIRA <sanpei@FreeBSD.org>

```
pub  1024R/391C5D69 1996-11-21 sanpei@SEAPLE.ICC.NE.JP
     Key fingerprint = EC 04 30 24 B0 6C 1E 63  5F 5D 25 59 3E 83 64 51
uid                            MIHIRA Yoshiro <sanpei@sanpei.org>
uid                            Yoshiro MIHIRA <sanpei@FreeBSD.org>
uid                            MIHIRA Yoshiro <sanpei@yy.cs.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@cc.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@educ.cc.keio.ac.jp>
uid                            MIHIRA Yoshiro <sanpei@st.keio.ac.jp>
```

D.3.241. Robert Millan <rmh@FreeBSD.org>

```
pub   4096R/DEA2C38E 2009-08-14
      Key fingerprint = A537 F029 AAAE 0E9C 39A7  C22C BB9D 98D9 DEA2 C38E
uid                  Robert Millan <rmh@debian.org>
uid                  Robert Millan <rmh@freebsd.org>
uid                  Robert Millan <rmh@gnu.org>
sub   4096R/65A0A9CE 2009-08-14
sub   4096R/41F37946 2009-08-14
```

D.3.242. Stephen Montgomery-Smith <stephen@FreeBSD.org>

```
pub   2048R/9A92D807 2011-06-14
      Key fingerprint = 2B61 D82E 168E F08B 6E08  712E 2DF1 2BD1 9A92 D807
uid                  Stephen Montgomery-Smith <stephen@freebsd.org>
sub   2048R/A4BA6560 2011-06-14
```

D.3.243. Marcel Moolenaar <marcel@FreeBSD.org>

```
pub  1024D/61EE89F6 2002-02-09 Marcel Moolenaar <marcel@xcllnt.net>
     Key fingerprint = 68BB E2B7 49AA FF69 CA3A  DF71 A605 A52D 61EE 89F6
sub  1024g/6EAAB456 2002-02-09
```

D.3.244. Kris Moore <kmoore@FreeBSD.org>

```
pub   1024D/6294612C 2009-05-26
      Key fingerprint = 8B70 9876 346F 1F97 5687  6950 4C92 D789 6294 612C
uid                  Kris Moore <kmoore@freebsd.org>
sub   2048g/A7FFE8FB 2009-05-26
```

D.3.245. Dmitry Morozovsky <marck@FreeBSD.org>

```
pub   1024D/6B691B03 2001-07-20
      Key fingerprint = 39AC E336 F03D C0F8 5305  B725 85D4 5045 6B69 1B03
uid                  Dmitry Morozovsky <marck@rinet.ru>
uid                  Dmitry Morozovsky <marck@FreeBSD.org>
sub   2048g/44D656F8 2001-07-20
```

D.3.246. Alexander Motin <mav@FreeBSD.org>

```
pub   1024D/0577BACA 2007-04-20 [expires: 2012-04-18]
      Key fingerprint = 0E84 B263 E97D 3E48 161B  98A2 D240 A09E 0577 BACA
uid                  Alexander Motin <mav@freebsd.org>
uid                  Alexander Motin <mav@mavhome.dp.ua>
uid                  Alexander Motin <mav@alkar.net>
sub   2048g/4D59D1C2 2007-04-20 [expires: 2012-04-18]
```

D.3.247. Felippe de Meirelles Motta <lippe@FreeBSD.org>

```
pub   1024D/F2CF7DAE 2008-09-02 [expires: 2010-09-02]
      Key fingerprint = 0532 A900 286D DAFD 099D  394D 231B AF20 F2CF 7DAE
uid                  Felippe de Meirelles Motta (FreeBSD Ports Committer) <lippe@FreeBSD.org>
sub   2048g/38E8EEF3 2008-09-02 [expires: 2010-09-02]
```

D.3.248. Rich Murphey <rich@FreeBSD.org>

```
pub  1024R/583443A9 1995-03-31 Rich Murphey <rich@lamprey.utmb.edu>
     Key fingerprint = AF A0 60 C4 84 D6 0C 73  D1 EF C0 E9 9D 21 DB E4
```

D.3.249. Akinori MUSHA <knu@FreeBSD.org>

```
pub  1024D/9FD9E1EE 2000-03-21 Akinori MUSHA <knu@and.or.jp>
     Key fingerprint = 081D 099C 1705 861D 4B70  B04A 920B EFC7 9FD9 E1EE
uid                            Akinori MUSHA <knu@FreeBSD.org>
uid                            Akinori MUSHA <knu@idaemons.org>
uid                            Akinori MUSHA <knu@ruby-lang.org>
sub  1024g/71BA9D45 2000-03-21
```

D.3.250. Thomas Möstl <tmm@FreeBSD.org>

```
pub  1024D/419C776C 2000-11-28 Thomas Moestl <tmm@FreeBSD.org>
     Key fingerprint = 1C97 A604 2BD0 E492 51D0  9C0F 1FE6 4F1D 419C 776C
uid                            Thomas Moestl <tmoestl@gmx.net>
uid                            Thomas Moestl <t.moestl@tu-bs.de>
sub  2048g/ECE63CE6 2000-11-28
```

D.3.251. Masafumi NAKANE <max@FreeBSD.org>

```
pub  1024D/CE356B59 2000-02-19 Masafumi NAKANE <max@wide.ad.jp>
     Key fingerprint = EB40 BCAB 4CE5 0764 9942  378C 9596 159E CE35 6B59
uid                            Masafumi NAKANE <max@FreeBSD.org>
uid                            Masafumi NAKANE <max@accessibility.org>
uid                            Masafumi NAKANE <kd5pdi@qsl.net>
sub  1024g/FA9BD48B 2000-02-19
```

D.3.252. Maho Nakata <maho@FreeBSD.org>

```
pub   1024D/F28B4069 2009-02-09
      Key fingerprint = 3FE4 99A9 6F41 8161 4F5F  240C 8615 A60C F28B 4069
uid                  Maho NAKATA (NAKATA's FreeBSD.org alias) <maho@FreeBSD.org>
sub   2048g/6B49098E 2009-02-09
```

D.3.253. Yoichi NAKAYAMA <yoichi@FreeBSD.org>

```
pub  1024D/E0788E46 2000-12-28 Yoichi NAKAYAMA <yoichi@assist.media.nagoya-u.ac.jp>
     Key fingerprint = 1550 2662 46B3 096C 0460  BC03 800D 0C8A E078 8E46
uid                            Yoichi NAKAYAMA <yoichi@eken.phys.nagoya-u.ac.jp>
uid                            Yoichi NAKAYAMA <yoichi@FreeBSD.org>
sub  1024g/B987A394 2000-12-28
```

D.3.254. Edward Tomasz Napierala <trasz@FreeBSD.org>

```
pub   1024D/8E53F00E 2007-04-13
      Key fingerprint = DD8F 91B0 12D9 6237 42D9  DBE1 AFC8 CDE9 8E53 F00E
uid                  Edward Tomasz Napierala <trasz@FreeBSD.org>
sub   2048g/7C1F5D67 2007-04-13
```

D.3.255. David Naylor <dbn@FreeBSD.org>

```
pub   1024D/FF6916B2 2008-04-09
      Key fingerprint = 6540 B47C 54AA 3EBA B23B  58AC 51A6 8580 FF69 16B2
uid                  David Naylor <dbn@freebsd.org>
```

```
uid                  David Naylor <naylor.b.david@gmail.com>
sub   4096g/77FA885C 2008-04-09
```

D.3.256. Alexander Nedotsukov <bland@FreeBSD.org>

```
pub  1024D/D004116C 2003-08-14 Alexander Nedotsukov <bland@FreeBSD.org>
     Key fingerprint = 35E2 5020 55FC 2071 4ADD  1A4A 86B6 8A5D D004 116C
sub  1024g/1CCA8D46 2003-08-14
```

D.3.257. George V. Neville-Neil <gnn@FreeBSD.org>

```
pub   1024D/440A33D2 2002-09-17
      Key fingerprint = AF66 410F CC8D 1FC9 17DB  6225 61D8 76C1 440A 33D2
uid                  George V. Neville-Neil <gnn@freebsd.org>
uid                  George V. Neville-Neil <gnn@neville-neil.com>
sub   2048g/95A74F6E 2002-09-17
```

D.3.258. Simon L. B. Nielsen <simon@FreeBSD.org>

```
pub   1024D/FF7490AB 2007-01-14
      Key fingerprint = 4E92 BA8D E45E 85E2 0380  B264 049C 7480 FF74 90AB
uid                  Simon L. Nielsen <simon@FreeBSD.org>
uid                  Simon L. Nielsen <simon@nitro.dk>
sub   2048g/E3F5A76E 2007-01-14
```

D.3.259. Robert Noland <rnoland@FreeBSD.org>

```
pub   1024D/8A9F44E3 2007-07-24
      Key fingerprint = 107A 0C87 E9D0 E581 677B  2A28 3384 EB43 8A9F 44E3
uid                  Robert C. Noland III <rnoland@FreeBSD.org>
uid                  Robert C. Noland III (Personal Key) <rnoland@2hip.net>
sub   2048g/76C3CF00 2007-07-24
```

D.3.260. Anders Nordby <anders@FreeBSD.org>

```
pub  1024D/00835956 2000-08-13 Anders Nordby <anders@fix.no>
     Key fingerprint = 1E0F C53C D8DF 6A8F EAAD  19C5 D12A BC9F 0083 5956
uid                            Anders Nordby <anders@FreeBSD.org>
sub  2048g/4B160901 2000-08-13
```

D.3.261. Michael Nottebrock <lofi@FreeBSD.org>

```
pub  1024D/6B2974B0 2002-06-06 Michael Nottebrock <michaelnottebrock@gmx.net>
     Key fingerprint = 1079 3C72 0726 F300 B8EC  60F9 5E17 3AF1 6B29 74B0
uid                            Michael Nottebrock <lofi@freebsd.org>
uid                            Michael Nottebrock <lofi@tigress.com>
uid                            Michael Nottebrock <lofi@lofi.dyndns.org>
uid                            Michael Nottebrock <michaelnottebrock@web.de>
uid                            Michael Nottebrock <michaelnottebrock@meitner.wh.uni-dortmund.de>
sub  1024g/EF652E04 2002-06-06 [expires: 2004-06-15]
```

D.3.262. David O’Brien <obrien@FreeBSD.org>

```
pub  1024R/34F9F9D5 1995-04-23 David E. O’Brien <defunct - obrien@Sea.Legent.com>
     Key fingerprint = B7 4D 3E E9 11 39 5F A3  90 76 5D 69 58 D9 98 7A
uid                            David E. O’Brien <obrien@NUXI.com>
uid                            deobrien@ucdavis.edu
uid                            David E. O’Brien <whois Do38>
uid                            David E. O’Brien <obrien@FreeBSD.org>
uid                            David E. O’Brien <dobrien@seas.gwu.edu>
uid                            David E. O’Brien <obrien@cs.ucdavis.edu>
uid                            David E. O’Brien <defunct - obrien@media.sra.com>
uid                            David E. O’Brien <obrien@elsewhere.roanoke.va.us>
uid                            David E. O’Brien <obrien@Nuxi.com>

pub  1024D/7F9A9BA2 1998-06-10 "David E. O’Brien" <obrien@cs.ucdavis.edu>
     Key fingerprint = 02FD 495F D03C 9AF2 5DB7  F496 6FC8 DABD 7F9A 9BA2
uid                            "David E. O’Brien" <obrien@NUXI.com>
uid                            "David E. O’Brien" <obrien@FreeBSD.org>
sub  3072g/BA32C20D 1998-06-10
```

D.3.263. Jimmy Olgeni <olgeni@FreeBSD.org>

```
pub   2048R/6450AE47 2012-11-01
      Key fingerprint = 7133 AB4D DFC8 0A0D F891  B0D2 90B7 A98E 6450 AE47
uid                  Giacomo Olgeni <olgeni@olgeni.com>
uid                  Jimmy Olgeni <olgeni@FreeBSD.org>
uid                  Giacomo Olgeni <olgeni@moviereading.com>
uid                  Giacomo Olgeni <olgeni@unimaccess.com>
uid                  Giacomo Olgeni <olgeni@colby.it>
uid                  Giacomo Olgeni <olgeni@colby.eu>
uid                  Giacomo Olgeni <olgeni@colby.tv>
sub   2048R/1988BB4B 2012-11-01
```

D.3.264. Philip Paeps <philip@FreeBSD.org>

```
pub   4096R/C5D34D05 2006-10-22
      Key fingerprint = 356B AE02 4763 F739 2FA2  E438 2649 E628 C5D3 4D05
uid                  Philip Paeps <philip@paeps.cx>
uid                  Philip Paeps <philip@nixsys.be>
uid                  Philip Paeps <philip@fosdem.org>
uid                  Philip Paeps <philip@freebsd.org>
uid                  Philip Paeps <philip@pub.telenet.be>
sub   1024D/035EFC58 2006-10-22
sub   2048g/6E5FD7D6 2006-10-22
```

D.3.265. Josh Paetzel <jpaetzel@FreeBSD.org>

```
pub   2048D/F6F63F01 2012-09-21
      Key fingerprint = 1D8D 506E B58C BD10 DC8C  97E1 D6AD 8621 F6F6 3F01
uid                  Josh Paetzel <josh@tcbug.org>
uid                  Josh Paetzel <josh@ixsystems.com>
uid                  Josh Paetzel <jpaetzel@FreeBSD.org>
sub   2048R/F32EF801 2012-09-21
sub   2048R/51F1335D 2012-09-21
sub   2048g/9BC280CD 2012-09-21
sub   2048g/CC793500 2012-09-21
```

D.3.266. Gábor Páli <pgj@FreeBSD.org>

```
pub   4096R/6D7E445C 2013-06-14 [expires: 2018-06-13]
      Key fingerprint = 7AD5 76BA AF2D 14B9 6D45  440B C013 309D 6D7E 445C
uid                  Páli Gábor János (Primary identity) <pali.gabor@gmail.com>
uid                  Páli Gábor János (Eötvös Loránd University) <pgj@inf.elte.hu>
uid                  Gabor Pali (FreeBSD committer) <pgj@FreeBSD.org>
uid                  Páli Gábor János (Magyar BSD Egyesület) <pgj@bsd.hu>
uid                  Páli Gábor János (Eötvös Loránd University) <pgj@elte.hu>
sub   4096R/A57B06AB 2013-06-14 [expires: 2018-06-13]
```

D.3.267. Hiren Panchasara <hiren@FreeBSD.org>

```
pub   4096R/61913185 2013-04-13 [expires: 2014-04-13]
      Key fingerprint = 3336 8104 8D15 B238 2465  136B 4A61 462F 6191 3185
uid                  hiren panchasara <hiren@freebsd.org>
```

D.3.268. Hiten Pandya <hmp@FreeBSD.org>

```
pub  1024D/938CACA8 2004-02-13 Hiten Pandya (FreeBSD) <hmp@FreeBSD.org>
     Key fingerprint = 84EB C75E C75A 50ED 304E  E446 D974 7842 938C ACA8
uid                           Hiten Pandya <hmp@backplane.com>
```

```
sub  2048g/783874B5 2004-02-13
```

D.3.269. Dima Panov <fluffy@FreeBSD.org>

```
pub   1024D/93E3B018 2006-11-08
      Key fingerprint = C73E 2B72 1FFD 61BD E206  1234 A626 76ED 93E3 B018
uid                  Dima Panov (FreeBSD.ORG Committer) <fluffy@FreeBSD.ORG>
uid                  Dima Panov (at home) <Fluffy@Fluffy.Khv.RU>
uid                  Dima Panov (at home) <fluffy.khv@gmail.com>
sub   2048g/89047419 2006-11-08

pub   4096R/D5398F29 2009-08-09
      Key fingerprint = 2D30 2CCB 9984 130C 6F87  BAFC FB8B A09D D539 8F29
uid                  Dima Panov (FreeBSD.ORG Committer) <fluffy@FreeBSD.ORG>
uid                  Dima Panov (at Home) <fluffy@Fluffy.Khv.RU>
uid                  Dima Panov (at GMail) <fluffy.khv@gmail.com>
sub   4096R/915A7785 2009-08-09
```

D.3.270. Andrew Pantyukhin <sat@FreeBSD.org>

```
pub   1024D/6F38A569 2006-05-06
      Key fingerprint = 4E94 994A C2EF CB86 C144  3B04 3381 67C0 6F38 A569
uid                  Andrew Pantyukhin <infofarmer@gubkin.ru>
uid                  Andrew Pantyukhin <sat@FreeBSD.org>
uid                  Andrew Pantyukhin <infofarmer@gmail.com>
uid                  Andrew Pantyukhin <infofarmer@mail.ru>
sub   2048g/5BD4D469 2006-05-06
```

D.3.271. Navdeep Parhar <np@FreeBSD.org>

```
pub   1024D/ACAB8812 2009-06-08
      Key fingerprint = C897 7AFB AFC0 4DA9 7B76  D991 CAB2 2B93 ACAB 8812
uid                  Navdeep Parhar <np@FreeBSD.org>
sub   2048g/AB61D2DC 2009-06-08
```

D.3.272. Rui Paulo <rpaulo@FreeBSD.org>

```
pub   4096R/39CB4153 2010-02-03
      Key fingerprint = ABE8 8465 DE8F F04D E9C8  3FF6 AF89 B2E6 39CB 4153
uid                  Rui Paulo <rpaulo@FreeBSD.org>
uid                  Rui Paulo <rpaulo@gmail.com>
sub   4096R/F87D2F34 2010-02-03
```

D.3.273. Mark Peek <mp@FreeBSD.org>

```
pub  1024D/330D4D01 2002-01-27 Mark Peek <mp@FreeBSD.org>
     Key fingerprint = 510C 96EE B4FB 1B0A 2CF8  A0AF 74B0 0B0E 330D 4D01
sub  1024g/9C6CAC09 2002-01-27
```

D.3.274. Peter Pentchev <roam@FreeBSD.org>

```
pub   1024D/16194553 2002-02-01
      Key fingerprint = FDBA FD79 C26F 3C51 C95E  DF9E ED18 B68D 1619 4553
uid                  Peter Pentchev <roam@ringlet.net>
uid                  Peter Pentchev <roam@cnsys.bg>
uid                  Peter Pentchev <roam@sbnd.net>
uid                  Peter Pentchev <roam@online.bg>
uid                  Peter Pentchev <roam@orbitel.bg>
uid                  Peter Pentchev <roam@FreeBSD.org>
uid                  Peter Pentchev <roam@techlab.office1.bg>
uid                  Peter Pentchev <roam@hoster.bg>
uid                  Peter Pentchev <roam@space.bg>
sub   1024g/7074473C 2002-02-01

pub   4096R/2527DF13 2009-10-16
      Key fingerprint = 2EE7 A7A5 17FC 124C F115  C354 651E EFB0 2527 DF13
uid                  Peter Pentchev <roam@ringlet.net>
uid                  Peter Pentchev <roamer@users.sourceforge.net>
uid                  Peter Pentchev <roam@cpan.org>
uid                  Peter Pentchev <roam@cnsys.bg>
uid                  Peter Pentchev <roam@sbnd.net>
uid                  Peter Pentchev <roam@online.bg>
uid                  Peter Pentchev <roam@orbitel.bg>
uid                  Peter Pentchev <roam@FreeBSD.org>
uid                  Peter Pentchev <roam@techlab.office1.bg>
uid                  Peter Pentchev <roam@hoster.bg>
uid                  Peter Pentchev <roam@space.bg>
uid                  Peter Pentchev <roam-guest@alioth.debian.org>
uid                  Peter Pentchev <ppentchev@alumni.princeton.edu>
sub   4096R/D0B337AA 2009-10-16
```

D.3.275. Denis Peplin <den@FreeBSD.org>

```
pub  1024D/485DDDF5 2003-09-11 Denis Peplin <den@FreeBSD.org>
     Key fingerprint = 495D 158C 8EC9 C2C1 80F5  EA96 6F72 7C1C 485D DDF5
sub  1024g/E70BA158 2003-09-11
```

D.3.276. Christian S.J. Peron <csjp@FreeBSD.org>

```
pub   1024D/033FA33C 2009-05-16
      Key fingerprint = 74AA 6040 89A7 936E D970  DDC0 CC71 6954 033F A33C
uid                  Christian S.J. Peron <csjp@FreeBSD.ORG>
sub   2048g/856B194A 2009-05-16
```

D.3.277. Gerald Pfeifer <gerald@FreeBSD.org>

```
pub  1024D/745C015A 1999-11-09 Gerald Pfeifer <gerald@pfeifer.com>
     Key fingerprint = B215 C163 3BCA 0477 615F  1B35 A5B3 A004 745C 015A
uid                            Gerald Pfeifer <Gerald.Pfeifer@vibe.at>
uid                            Gerald Pfeifer <pfeifer@dbai.tuwien.ac.at>
uid                            Gerald Pfeifer <gerald@pfeifer.at>
uid                            Gerald Pfeifer <gerald@FreeBSD.org>
sub  1536g/F0156927 1999-11-09
```

D.3.278. Giuseppe Pilichi <jacula@FreeBSD.org>

```
pub   4096R/8B9F4B8B 2006-03-08
      Key fingerprint = 31AD 73AE 0EC0 16E5 4108  8391 D942 5F20 8B9F 4B8B
uid                  Giuseppe Pilichi (Jacula Modyun) <jacula@FreeBSD.org>
uid                  Giuseppe Pilichi (Jacula Modyun) <jaculamodyun@gmail.com>
uid                  Giuseppe Pilichi (Jacula Modyun) <gpilch@gmail.com>
uid                  Giuseppe Pilichi (Jacula Modyun) <jacula@gmail.com>
sub   4096R/FB4D05A3 2006-03-08
```

D.3.279. John Polstra <jdp@FreeBSD.org>

```
pub  1024R/BFBCF449 1997-02-14 John D. Polstra <jdp@polstra.com>
     Key fingerprint = 54 3A 90 59 6B A4 9D 61  BF 1D 03 09 35 8D F6 0D
```

D.3.280. Kirill Ponomarew <krion@FreeBSD.org>

```
pub   1024D/AEB426E5 2002-04-07
      Key fingerprint = 58E7 B953 57A2 D9DD 4960  2A2D 402D 46E9 AEB4 26E5
uid                  Kirill Ponomarew <krion@voodoo.bawue.com>
uid                  Kirill Ponomarew <krion@guug.de>
uid                  Kirill Ponomarew <krion@FreeBSD.org>
sub   1024D/05AC7CA0 2006-01-30 [expires: 2008-01-30]
sub   2048g/C3EE5537 2006-01-30 [expires: 2008-01-30]
```

D.3.281. Stephane E. Potvin <sepotvin@FreeBSD.org>

```
pub   1024D/3097FE7B 2002-08-06
      Key fingerprint = 6B56 62FA ADE1 6F46 BB62  8B1C 99D3 97B5 3097 FE7B
uid                  Stephane E. Potvin <sepotvin@videotron.ca>
uid                  Stephane E. Potvin <stephane.potvin@telcobridges.com>
uid                  Stephane E. Potvin <stephane_potvin@telcobridges.com>
uid                  Stephane E. Potvin <sepotvin@FreeBSD.org>
sub   2048g/0C427BC9 2002-08-06
```

D.3.282. Mark Pulford <markp@FreeBSD.org>

```
pub  1024D/182C368F 2000-05-10 Mark Pulford <markp@FreeBSD.org>
     Key fingerprint = 58C9 C9BF C758 D8D4 7022  8EF5 559F 7F7B 182C 368F
uid                           Mark Pulford <mark@kyne.com.au>
sub  2048g/380573E8 2000-05-10
```

D.3.283. Alejandro Pulver <alepulver@FreeBSD.org>

```
pub   1024D/945C3F61 2005-11-13
      Key fingerprint = 085F E8A2 4896 4B19 42A4  4179 895D 3912 945C 3F61
uid                  Alejandro Pulver (Ale's GPG key pair) <alepulver@FreeBSD.org>
uid                  Alejandro Pulver (Ale's GPG key pair) <alejandro@varnet.biz>
sub   2048g/6890C6CA 2005-11-13
```

D.3.284. Thomas Quinot <thomas@FreeBSD.org>

```
pub  1024D/393D2469 1999-09-23 Thomas Quinot <thomas@cuivre.fr.eu.org>
 Empreinte de la clé = 4737 A0AD E596 6D30 4356  29B8 004D 54B8 393D 2469
uid                           Thomas Quinot <thomas@debian.org>
uid                           Thomas Quinot <thomas@FreeBSD.org>
sub  1024g/8DE13BB2 1999-09-23
```

D.3.285. Herve Quiroz <hq@FreeBSD.org>

```
pub  1024D/85AC8A80 2004-07-22 Herve Quiroz <hq@FreeBSD.org>
     Key fingerprint = 14F5 BC56 D736 102D 41AF  A07B 1D97 CE6C 85AC 8A80
uid                           Herve Quiroz <herve.quiroz@esil.univ-mrs.fr>
sub  1024g/8ECCAFED 2004-07-22
```

D.3.286. Doug Rabson <dfr@FreeBSD.org>

```
pub   1024D/59F57821 2004-02-07
      Key fingerprint = 9451 C4FE 1A7E 117B B95F  1F8F B123 456E 59F5 7821
uid                  Doug Rabson <dfr@nlsystems.com>
sub   1024g/6207AA32 2004-02-07
```

D.3.287. Lars Balker Rasmussen <lbr@FreeBSD.org>

```
pub   1024D/9EF6F27F 2006-04-30
      Key fingerprint = F251 28B7 897C 293E 04F8  71EE 4697 F477 9EF6 F27F
uid                  Lars Balker Rasmussen <lbr@FreeBSD.org>
sub   2048g/A8C1CFD4 2006-04-30
```

D.3.288. Chris Rees <crees@FreeBSD.org>

```
pub   2048R/1E12E96A 2012-08-26
      Key fingerprint = 8C57 BE3B D320 5FFC C4C3  C0B0 900F 45A6 1E12 E96A
uid                  Chris Rees <crees@FreeBSD.org>
sub   2048R/C10740CD 2012-08-26 [expires: 2013-08-26]
```

D.3.289. Jim Rees <rees@FreeBSD.org>

```
pub   512/B623C791 1995/02/21 Jim Rees <rees@umich.edu>
          Key fingerprint =  02 5F 1B 15 B4 6E F1 3E  F1 C5 E0 1D EA CC 17 88
```

D.3.290. Benedict Reuschling <bcr@FreeBSD.org>

```
pub   1024D/4A819348 2009-05-24
      Key fingerprint = 2D8C BDF9 30FA 75A5 A0DF  D724 4D26 502E 4A81 9348
uid                  Benedict Reuschling <bcr@FreeBSD.org>
sub   2048g/8DA16EDD 2009-05-24
```

D.3.291. Tom Rhodes <trhodes@FreeBSD.org>

```
pub   1024D/FB7D88E1 2008-05-07
      Key fingerprint = 8279 3100 2DF2 F00E 7FDD  AC2C 5776 23AB FB7D 88E1
uid                  Tom Rhodes (trhodes) <trhodes@FreeBSD.org>
sub   4096g/7B0CD79F 2008-05-07
```

D.3.292. Benno Rice <benno@FreeBSD.org>

```
pub   4096R/C5F10BED 2013-05-21 [expires: 2017-05-21]
      Key fingerprint = 77EB 5A9E 97C7 2D2D 6D0A  1B6C C619 4C61 C5F1 0BED
uid                  Benno Rice <benno@FreeBSD.org>
uid                  Benno Rice <benno@jeamland.net>
sub   4096R/408068BC 2013-05-21 [expires: 2017-05-21]
```

D.3.293. Beech Rintoul <beech@FreeBSD.org>

```
pub   2048D/68DFAE1F 2013-02-26
      Key fingerprint = D58B 3E9D B0E3 E081 EC6F  69D9 CDA3 51DD 68DF AE1F
uid                  Beech Rintoul <beech@freebsd.org>
sub   2048g/960F45D9 2013-02-26
```

D.3.294. Matteo Riondato <matteo@FreeBSD.org>

```
pub   1024D/1EC56BEC 2003-01-05 [expires: 2009-09-07]
      Key fingerprint = F0F3 1B43 035D 65B1 08E9  4D66 D8CA 78A5 1EC5 6BEC
uid                  Matteo Riondato (Rionda) <matteo@FreeBSD.ORG>
uid                  Matteo Riondato (Rionda) <rionda@riondabsd.net>
uid                  Matteo Riondato (Rionda) <rionda@gufi.org>
uid                  Matteo Riondato (Rionda) <matteo@riondato.com>
uid                  Matteo Riondato (Rionda) <rionda@riondato.com>
uid                  Matteo Riondato (Rionda) <rionda@FreeSBIE.ORG>
uid                  Matteo Riondato (Rionda) <rionda@autistici.org>
sub   2048g/87C44A55 2008-09-23 [expires: 2009-09-23]
```

D.3.295. Ollivier Robert <roberto@FreeBSD.org>

```
pub   1024D/7DCAE9D3 1997-08-21
      Key fingerprint = 2945 61E7 D4E5 1D32 C100  DBEC A04F FB1B 7DCA E9D3
uid                  Ollivier Robert <roberto@keltia.freenix.fr>
uid                  Ollivier Robert <roberto@FreeBSD.org>
sub   2048g/C267084D 1997-08-21
```

D.3.296. Craig Rodrigues <rodrigc@FreeBSD.org>

```
pub   1024D/3998479D 2005-05-20
      Key fingerprint = F01F EBE6 F5C8 6DC2 954F  098F D20A 8A2A 3998 479D
uid                  Craig Rodrigues <rodrigc@freebsd.org>
uid                  Craig Rodrigues <rodrigc@crodrigues.org>
sub   2048g/AA77E09B 2005-05-20
```

D.3.297. Guido van Rooij <guido@FreeBSD.org>

```
pub  1024R/599F323D 1996-05-18 Guido van Rooij <guido@gvr.org>
     Key fingerprint = 16 79 09 F3 C0 E4 28 A7  32 62 FA F6 60 31 C0 ED
uid                            Guido van Rooij <guido@gvr.win.tue.nl>

pub  1024D/A95102C1 2000-10-25 Guido van Rooij <guido@madison-gurkha.nl>
     Key fingerprint = 5B3E 51B7 0E7A D170 0574  1E51 2471 117F A951 02C1
uid                            Guido van Rooij <guido@madison-gurkha.com>
sub  1024g/A5F20553 2000-10-25
```

D.3.298. Eygene Ryabinkin <rea@FreeBSD.org>

```
pub   3072D/8152ECFB 2010-10-27
      Key fingerprint = 82FE 06BC D497 C0DE 49EC  4FF0 16AF 9EAE 8152 ECFB
uid                  Eygene Ryabinkin <rea-fbsd@codelabs.ru>
uid                  Eygene Ryabinkin <rea@freebsd.org>
uid                  Eygene Ryabinkin <rea@codelabs.ru>
sub   3072g/5FC03749 2010-10-27
```

D.3.299. Aleksandr Rybalko <ray@FreeBSD.org>

```
pub   2048R/4B7B7A4E 2011-05-24
      Key fingerprint = BB9F D01D 7327 0B33 B2F5  6C72 EC49 E6ED 4B7B 7A4E
uid                  Aleksandr Rybalko (Aleksandr Rybalko FreeBSD project identification) <ray@fr
sub   2048R/99F9F9EF 2011-05-24
```

D.3.300. Niklas Saers <niklas@FreeBSD.org>

```
pub  1024D/C822A476 2004-03-09 Niklas Saers <niklas@saers.com>
     Key fingerprint = C41E F734 AF0E 3D21 7499  9EB1 9A31 2E7E C822 A476
sub  1024g/81E2FF36 2004-03-09
```

D.3.301. Boris Samorodov <bsam@FreeBSD.org>

```
pub   1024D/ADFD5C9A 2006-06-21
      Key fingerprint = 81AA FED0 6050 208C 0303  4007 6C03 7263 ADFD 5C9A
uid                  Boris Samorodov (FreeBSD) <bsam@freebsd.org>
sub   2048g/7753A3F1 2006-06-21
```

D.3.302. Mark Santcroos <marks@FreeBSD.org>

```
pub   1024D/DBE7EB8E 2005-03-08
      Key fingerprint = C0F0 44F3 3F15 520F 6E32  186B BE0A BA42 DBE7 EB8E
uid                  Mark Santcroos <marks@ripe.net>
uid                  Mark Santcroos <mark@santcroos.net>
uid                  Mark Santcroos <marks@freebsd.org>
sub   2048g/FFF80F85 2005-03-08
```

D.3.303. Bernhard Schmidt <bschmidt@FreeBSD.org>

```
pub   1024D/5F754FBC 2009-06-15
      Key fingerprint = 6B87 C8A9 6BA5 6B18 11CF  8C38 A1B7 0731 5F75 4FBC
uid                  Bernhard Schmidt <bschmidt@FreeBSD.org>
uid                  Bernhard Schmidt <bschmidt@techwires.net>
sub   1024g/1945DC1D 2009-06-15
```

D.3.304. Wolfram Schneider <wosch@FreeBSD.org>

```
Type Bits/KeyID    Date       User ID
pub  1024/2B7181AD 1997/08/09 Wolfram Schneider <wosch@FreeBSD.org>
            Key fingerprint = CA 16 91 D9 75 33 F1 07  1B F0 B4 9F 3E 95 B6 09
```

D.3.305. Ed Schouten <ed@FreeBSD.org>

```
pub   4096R/3491A2BB 2011-03-12 [expires: 2016-03-10]
      Key fingerprint = A110 5982 A887 74A2 F4B1  D70A 6E5E D8FE 3491 A2BB
uid                  Ed Schouten (The FreeBSD Project) <ed@FreeBSD.org>
uid                  Ed Schouten <ed@80386.nl>
sub   4096R/81BB41E6 2011-03-12 [expires: 2016-03-10]
```

D.3.306. David Schultz <das@FreeBSD.org>

```
pub  1024D/BE848B57 2001-07-19 David Schultz <das@FreeBSD.ORG>
     Key fingerprint = 0C12 797B A9CB 19D9 FDAF  2A39 2D76 A2DB BE84 8B57
uid  David Schultz <dschultz@uclink.Berkeley.EDU>
uid  David Schultz <das@FreeBSD.ORG>
sub  2048g/69206E8E 2001-07-19
```

D.3.307. Michael Scheidell <scheidell@FreeBSD.org>

```
pub   2048R/34622C1D 2011-11-16
      Key fingerprint = 0A0C 9ECA 18EC 47AC C715  2187 91B9 F9FE 3462 2C1D
uid                  Michael Scheidell <scheidell@freebsd.org>
```

```
sub   2048R/8F241971 2011-11-16
```

D.3.308. Jens Schweikhardt <schweikh@FreeBSD.org>

```
pub  1024D/0FF231FD 2002-01-27 Jens Schweikhardt <schweikh@FreeBSD.org>
     Key fingerprint = 3F35 E705 F02F 35A1 A23E  330E 16FE EA33 0FF2 31FD
uid                            Jens Schweikhardt <schweikh@schweikhardt.net>
sub  1024g/6E93CACC 2002-01-27 [expires: 2005-01-26]
```

D.3.309. Matthew Seaman <matthew@FreeBSD.org>

```
pub   1024D/60AE908C 2005-12-17 [expires: 2012-03-21]
      Key fingerprint = B555 2A96 274E D248 5734  0EB4 F0C8 E4E7 60AE 908C
uid                  Matthew Seaman <m.seaman@infracaninophile.co.uk>
uid                  Matthew Seaman <m.seaman@black-earth.co.uk>
uid                  Matthew Seaman <matthew@freebsd.org>
sub   2048g/58BFDA29 2005-12-17 [expires: 2012-03-21]
sub   1024D/9B19F956 2006-12-18 [expires: 2012-03-21]
```

D.3.310. Thomas-Martin Seck <tmseck@FreeBSD.org>

```
pub   1024D/DF46EE05 2000-11-22
      Key fingerprint = A38F AE66 6B11 6EB9 5D1A  B67D 2444 2FE1 DF46 EE05
uid                  Thomas-Martin Seck (Privat 2) <tmseck@netcologne.de>
uid                  Thomas-Martin Seck (Privat) <tmseck@web.de>
uid                  Thomas-Martin Seck (FreeBSD) <tmseck@FreeBSD.org>
sub   2048g/3DC33B0F 2000-11-22
```

D.3.311. Stanislav Sedov <stas@FreeBSD.org>

```
pub   4096R/092FD9F0 2009-05-23
      Key fingerprint = B83A B15D 929A 364A D8BC  B3F9 BF25 A231 092F D9F0
uid                  Stanislav Sedov <stas@FreeBSD.org>
uid                  Stanislav Sedov <stas@SpringDaemons.com>
uid                  Stanislav Sedov (Corporate email) <stas@deglitch.com>
uid                  Stanislav Sedov (Corporate email) <stas@ht-systems.ru>
uid                  Stanislav Sedov (Corporate email) <ssedov@3playnet.com>
uid                  Stanislav Sedov <ssedov@mbsd.msk.ru>
uid                  Stanislav Sedov (Corporate email) <ssedov@swifttest.com>
sub   4096R/6FD2025F 2009-05-23
```

D.3.312. Johan van Selst <johans@FreeBSD.org>

```
pub   4096R/D3AE8D3A 2009-09-01
      Key fingerprint = 31C8 D089 DDB6 96C6 F3C1  29C0 A9C8 6C8D D3AE 8D3A
uid                  Johan van Selst
uid                  Johan van Selst <johans@gletsjer.net>
uid                  Johan van Selst <johans@stack.nl>
uid                  Johan van Selst <johans@FreeBSD.org>
uid                  Johan van Selst (GSWoT:NL50) <johans@gswot.org>
sub   2048R/B002E38C 2009-09-01
sub   2048R/1EBCAECB 2009-09-01
sub   2048R/639A1446 2009-09-01
sub   3072D/6F2708F4 2009-09-01
sub   4096g/D6F89E83 2009-09-01
```

D.3.313. Bakul Shah <bakul@FreeBSD.org>

```
pub   1024D/86AEE4CB 2006-04-20
      Key fingerprint = 0389 26E8 381C 6980 AEC0  10A5 E540 A157 86AE E4CB
uid                  Bakul Shah <bakul@freebsd.org>
sub   2048g/5C3DCC24 2006-04-20
```

D.3.314. Gregory Neil Shapiro <gshapiro@FreeBSD.org>

```
pub  1024R/4FBE2ADD 2000-10-13 Gregory Neil Shapiro <gshapiro@gshapiro.net>
     Key fingerprint = 56 D5 FF A7 A6 54 A6 B5  59 10 00 B9 5F 5F 20 09
uid                            Gregory Neil Shapiro <gshapiro@FreeBSD.org>

pub  1024D/F76A9BF5 2001-11-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
     Key fingerprint = 3B5E DAF1 4B04 97BA EE20  F841 21F9 C5BC F76A 9BF5
uid                            Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  2048g/935657DC 2001-11-14

pub  1024D/FCE56561 2000-10-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
     Key fingerprint = 42C4 A87A FD85 C34F E77F  5EA1 88E1 7B1D FCE5 6561
uid                            Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  1024g/285DC8A0 2000-10-14 [expires: 2001-10-14]
```

D.3.315. Arun Sharma <arun@FreeBSD.org>

```
pub  1024D/7D112181 2003-03-06 Arun Sharma <arun@sharma-home.net>
     Key fingerprint = A074 41D6 8537 C7D5 070E  0F78 0247 1AE2 7D11 2181
uid                            Arun Sharma <arun@freebsd.org>
uid                            Arun Sharma <arun.sharma@intel.com>
sub  1024g/ACAD98DA 2003-03-06 [expires: 2005-03-05]
```

D.3.316. Wesley Shields <wxs@FreeBSD.org>

```
pub   1024D/17F0AA37 2007-12-27
      Key fingerprint = 96D1 2E6B F61C 2F3D 83EF  8F0B BE54 310C 17F0 AA37
uid                  Wesley Shields <wxs@FreeBSD.org>
uid                  Wesley Shields <wxs@atarininja.org>
sub   2048g/2EDA1BB8 2007-12-27
```

D.3.317. Norikatsu Shigemura <nork@FreeBSD.org>

```
pub   1024D/7104EA4E 2005-02-14
      Key fingerprint = 9580 60A3 B58A 0864 79CB  779A 6FAE 229B 7104 EA4E
uid                  Norikatsu Shigemura <nork@cityfujisawa.ne.jp>
uid                  Norikatsu Shigemura <nork@ninth-nine.com>
uid                  Norikatsu Shigemura <nork@FreeBSD.org>
sub   4096g/EF56997E 2005-02-14
```

D.3.318. Shteryana Shopova <syrinx@FreeBSD.org>

```
pub   1024D/1C139BC5 2006-10-07
      Key fingerprint = B83D 2451 27AB B767 504F  CB85 4FB1 C88B 1C13 9BC5
uid                  Shteryana Shopova (syrinx) <shteryana@FreeBSD.org>
sub   2048g/6D2E9C98 2006-10-07
```

D.3.319. Vanilla I. Shu <vanilla@FreeBSD.org>

```
pub  1024D/ACE75853 2001-11-20 Vanilla I. Shu <vanilla@FreeBSD.org>
     Key fingerprint = 290F 9DB8 42A3 6257 5D9A  5585 B25A 909E ACE7 5853
sub  1024g/CE695D0E 2001-11-20
```

D.3.320. Ashish SHUKLA <ashish@FreeBSD.org>

```
pub   4096R/E74FA4B0 2010-04-13
      Key fingerprint = F682 CDCC 39DC 0FEA E116  20B6 C746 CFA9 E74F A4B0
uid                  Ashish SHUKLA <wahjava@gmail.com>
uid                  Ashish SHUKLA <wahjava@googlemail.com>
uid                  Ashish SHUKLA <wahjava.ml@gmail.com>
uid                  Ashish SHUKLA <wahjava@members.fsf.org>
uid                  Ashish SHUKLA <wahjava@perl.org.in>
uid                  Ashish SHUKLA <wahjava@users.sourceforge.net>
uid                  Ashish SHUKLA <wah.java@yahoo.com>
uid                  Ashish SHUKLA <wah_java@hotmail.com>
uid                  Ashish SHUKLA <ashish.shukla@airtelmail.in>
uid                  Ashish SHUKLA <wahjava@member.fsf.org>
uid                  [jpeg image of size 4655]
uid                  Ashish SHUKLA (FreeBSD Committer Address) <ashish@FreeBSD.ORG>
```

```
sub   4096R/F20D202D 2010-04-13
```

D.3.321. Bruce M. Simpson <bms@FreeBSD.org>

```
pub  1024D/860DB53B 2003-08-06 Bruce M Simpson <bms@freebsd.org>
     Key fingerprint = 0D5F 1571 44DF 51B7 8B12  041E B9E5 2901 860D B53B
sub  2048g/A2A32D8B 2003-08-06 [expires: 2006-08-05]
```

D.3.322. Dmitry Sivachenko <demon@FreeBSD.org>

```
pub  1024D/13D5DF80 2002-03-18 Dmitry Sivachenko <mitya@cavia.pp.ru>
     Key fingerprint = 72A9 12C9 BB02 46D4 4B13  E5FE 1194 9963 13D5 DF80
uid                            Dmitry S. Sivachenko <demon@FreeBSD.org>
sub  1024g/060F6DBD 2002-03-18
```

D.3.323. Jesper Skriver <jesper@FreeBSD.org>

```
pub  1024D/F9561C31 2001-03-09 Jesper Skriver <jesper@FreeBSD.org>
     Key fingerprint = 6B88 9CE8 66E9 E631 C9C5  5EB4 22AB F0EC F956 1C31
uid                            Jesper Skriver <jesper@skriver.dk>
uid                            Jesper Skriver <jesper@wheel.dk>
sub  1024g/777C378C 2001-03-09
```

D.3.324. Ville Skyttä <scop@FreeBSD.org>

```
pub  1024D/BCD241CB 2002-04-07 Ville Skyttä <ville.skytta@iki.fi>
     Key fingerprint = 4E0D EBAB 3106 F1FA 3FA9  B875 D98C D635 BCD2 41CB
uid                            Ville Skyttä <ville.skytta@xemacs.org>
uid                            Ville Skyttä <scop@FreeBSD.org>
sub  2048g/9426F4D1 2002-04-07
```

D.3.325. Andrey Slusar <anray@FreeBSD.org>

```
pub   1024D/AE7B5418 2005-12-12
      Key fingerprint = DE70 C24B 55A0 4A06 68A1  D425 3C59 9A9B AE7B 5418
uid                  Andrey Slusar <anray@ext.by>
uid                  Andrey Slusar <anrays@gmail.com>
uid                  Andrey Slusar <anray@FreeBSD.org>
sub   2048g/7D0EB77D 2005-12-12
```

D.3.326. Florian Smeets <flo@FreeBSD.org>

```
pub   1024D/C942BF09 2008-10-24
      Key fingerprint = 54BB 157B 8DB2 9E46 4A3C  69AB 6A9A 3C3F C942 BF09
uid                  Florian Smeets <flo@smeets.im>
uid                  Florian Smeets <flo@kasimir.com>
uid                  Florian Smeets <flo@FreeBSD.org>
sub   2048g/4AAF040E 2008-10-24
```

D.3.327. Gleb Smirnoff <glebius@FreeBSD.org>

```
pub   2048D/6C7E5E82 2013-01-30 [expires: 2023-08-25]
      Key fingerprint = 6E06 7260 B83D CF2C A93C  566F 5185 0968 6C7E 5E82
uid                  Gleb Smirnoff <glebius@FreeBSD.org>
sub   2048g/11E89DCE 2013-01-30 [expires: 2023-08-25]
```

D.3.328. Ken Smith <kensmith@FreeBSD.org>

```
pub  1024D/29AEA7F6 2003-12-02 Ken Smith <kensmith@cse.buffalo.edu>
     Key fingerprint = 4AB7 D302 0753 8215 31E7  F1AD FC6D 7855 29AE A7F6
uid                            Ken Smith <kensmith@freebsd.org>
sub  1024g/0D509C6C 2003-12-02
```

D.3.329. Ben Smithurst <ben@FreeBSD.org>

```
pub  1024D/2CEF442C 2001-07-11 Ben Smithurst <ben@LSRfm.com>
     Key fingerprint = 355D 0FFF B83A 90A9 D648  E409 6CFC C9FB 2CEF 442C
uid                            Ben Smithurst <ben@vinosystems.com>
uid                            Ben Smithurst <ben@smithurst.org>
uid                            Ben Smithurst <ben@FreeBSD.org>
uid                            Ben Smithurst <csxbcs@comp.leeds.ac.uk>
uid                            Ben Smithurst <ben@scientia.demon.co.uk>
sub  1024g/347071FF 2001-07-11
```

D.3.330. Dag-Erling C. Smørgrav <des@FreeBSD.org>

```
pub   4096R/F94E87B2 2013-02-15 [expires: 2015-01-01]
      Key fingerprint = 578A 3F4F 9E04 9FCF 3576  BF82 BB9B 471B F94E 87B2
uid                  Dag-Erling Smørgrav <des@usit.uio.no>
uid                  Dag-Erling Smørgrav <des@des.no>
uid                  Dag-Erling Smørgrav <des@freebsd.org>
uid                  [jpeg image of size 4779]
sub   4096R/F4DE87F5 2013-02-15 [expires: 2015-01-01]
```

D.3.331. Maxim Sobolev <sobomax@FreeBSD.org>

```
pub  1024D/888205AF 2001-11-21 Maxim Sobolev <sobomax@FreeBSD.org>
     Key fingerprint = 85C9 DCB0 6828 087C C977  3034 A0DB B9B7 8882 05AF
uid                            Maxim Sobolev <sobomax@mail.ru>
uid                            Maxim Sobolev <sobomax@altavista.net>
uid                            Maxim Sobolev <vegacap@i.com.ua>

pub  1024D/468EE6D8 2003-03-21 Maxim Sobolev <sobomax@portaone.com>
     Key fingerprint = 711B D315 3360 A58F 9A0E  89DB 6D40 2558 468E E6D8
uid                            Maxim Sobolev <sobomax@FreeBSD.org>
uid                            Maxim Sobolev <sobomax@mail.ru>
uid                            Maxim Sobolev <vegacap@i.com.ua>

pub  1024D/6BEC980A 2004-02-13 Maxim Sobolev <sobomax@portaone.com>
     Key fingerprint = 09D5 47B4 8D23 626F B643  76EB DFEE 3794 6BEC 980A
uid                            Maxim Sobolev <sobomax@FreeBSD.org>
uid                            Maksym Sobolyev (It’s how they call me in official documents. Pret
uid                            Maksym Sobolyev (It’s how they call me in official documents. Pret
sub  2048g/16D049AB 2004-02-13 [expires: 2005-02-12]
```

D.3.332. Alan Somers <asomers@FreeBSD.org>

```
pub   4096R/DA05FCE8 2013-04-25 [expires: 2018-04-24]
      Key fingerprint = 9CD4 C982 738F 8B90 25E8  E6B3 5F74 63BC DA05 FCE8
uid                  Alan Somers <asomers@freebsd.org>
uid                  Alan Somers <asomers@gmail.com>
sub   4096R/4E121B3E 2013-04-25 [expires: 2018-04-24]
```

D.3.333. Brian Somers <brian@FreeBSD.org>

```
pub  1024R/666A7421 1997-04-30 Brian Somers <brian@freebsd-services.com>
     Key fingerprint = 2D 91 BD C2 94 2C 46 8F  8F 09 C4 FC AD 12 3B 21
uid                            Brian Somers <brian@awfulhak.org>
uid                            Brian Somers <brian@FreeBSD.org>
uid                            Brian Somers <brian@OpenBSD.org>
uid                            Brian Somers <brian@uk.FreeBSD.org>
uid                            Brian Somers <brian@uk.OpenBSD.org>
```

D.3.334. Stacey Son <sson@FreeBSD.org>

```
pub   1024D/CE8319F3 2008-07-08
      Key fingerprint = 64C7 8D92 C1DF B940 1171  5ED3 186A 758A CE83 19F3
uid                   Stacey Son <sson@FreeBSD.org>
uid                   Stacey Son <stacey@son.org>
uid                   Stacey Son <sson@byu.net>
uid                   Stacey Son <sson@secure.net>
uid                   Stacey Son <sson@dev-random.com>
```

```
sub   2048g/0F724E52 2008-07-08
```

D.3.335. Nicolas Souchu <nsouch@FreeBSD.org>

```
pub  1024D/C744F18B 2002-02-13 Nicholas Souchu <nsouch@freebsd.org>
     Key fingerprint = 992A 144F AC0F 40BA 55AE  DE6D 752D 0A6C C744 F18B
sub  1024g/90BD3231 2002-02-13
```

D.3.336. Suleiman Souhlal <ssouhlal@FreeBSD.org>

```
pub  1024D/2EA50469 2004-07-24 Suleiman Souhlal <ssouhlal@FreeBSD.org>
     Key fingerprint = DACF 89DB 54C7 DA1D 37AF  9A94 EB55 E272 2EA5 0469
sub  2048g/0CDCC535 2004-07-24
```

D.3.337. Luiz Otavio O Souza <loos@FreeBSD.org>

```
pub   2048R/39165690 2013-07-03
      Key fingerprint = ABC9 71D9 016E 8D4A 936D  D748 6252 872F 3916 5690
uid                  Luiz Otavio O Souza <loos@freebsd.org>
sub   2048R/9D089395 2013-07-03
```

D.3.338. Ulrich Spörlein <uqs@FreeBSD.org>

```
pub   2048R/4AAF82CE 2010-01-27 [expires: 2015-01-26]
      Key fingerprint = 08DF A6A0 B1EB 98A5 EDDA  9005 A3A6 9864 4AAF 82CE
uid                  Ulrich Spörlein <uqs@spoerlein.net>
uid                  Ulrich Spoerlein <uspoerlein@gmail.com>
uid                  Ulrich Spörlein (The FreeBSD Project) <uqs@FreeBSD.org>
uid                  Ulrich Spörlein <ulrich.spoerlein@web.de>
sub   2048R/162E8BD2 2010-01-27 [expires: 2015-01-26]
```

D.3.339. Rink Springer <rink@FreeBSD.org>

```
pub   1024D/ECEDBFFF 2003-09-19
      Key fingerprint = A8BE 9C82 9B81 4289 A905  418D 6F73 BAD2 ECED BFFF
uid                  Rink Springer <rink@il.fontys.nl>
uid                  Rink Springer (FreeBSD Project) <rink@FreeBSD.org>
uid                  Rink Springer <rink@stack.nl>
sub   2048g/3BC3E67E 2003-09-19
```

D.3.340. Vsevolod Stakhov <vsevolod@FreeBSD.org>

```
pub   4096R/90081437 2012-05-16 [expires: 2017-05-15]
      Key fingerprint = DD9A 126C E675 1EA5 2A97  04A3 0764 7B67 9008 1437
uid                  Vsevolod Stakhov <vsevolod@FreeBSD.org>
sub   4096R/4A5A0B54 2012-05-16 [expires: 2017-05-15]
```

D.3.341. Ryan Steinmetz <zi@FreeBSD.org>

```
pub   1024D/7AD7FAF2 2004-01-21
      Key fingerprint = EF36 D45A 5CA9 28B1 A550  18CD A43C D111 7AD7 FAF2
uid                  Ryan Steinmetz <zi@FreeBSD.org>
uid                  Ryan Steinmetz <rpsfa@rit.edu>
uid                  Ryan Steinmetz <zi@zi0r.com>
sub   1024g/058BC057 2004-01-21
sub   4096g/0EB108D2 2006-02-27
sub   1024D/FEF36DD7 2006-02-27
```

D.3.342. Randall R. Stewart <rrs@FreeBSD.org>

```
pub   1024D/0373B8B2 2006-09-01
      Key fingerprint = 74A6 810E 6DEA D69B 6496  5FA9 8AEF 4166 0373 B8B2
uid                  Randall R Stewart <randall@lakerest.net>
uid                  Randall R Stewart <rrs@cisco.com>
uid                  Randall R Stewart <rrs@FreeBSD.org>
sub   2048g/88027C0B 2006-09-01
```

D.3.343. Murray Stokely <murray@FreeBSD.org>

```
pub  1024D/0E451F7D 2001-02-12 Murray Stokely <murray@freebsd.org>
     Key fingerprint = E2CA 411D DD44 53FD BB4B  3CB5 B4D7 10A2 0E45 1F7D
sub  1024g/965A770C 2001-02-12
```

D.3.344. Volker Stolz <vs@FreeBSD.org>

```
pub  1024R/3FD1B6B5 1998-06-16 Volker Stolz <vs@freebsd.org>
     Key fingerprint = 69 6F BD A0 2E FE 19 66  CF B9 68 6E 41 7D F9 B9
uid                           Volker Stolz <stolz@i2.informatik.rwth-aachen.de> (LSK)
uid                           Volker Stolz <vs@foldr.org>
```

D.3.345. Ryan Stone <rstone@FreeBSD.org>

```
pub   1024D/3141B73A 2010-04-13
      Key fingerprint = 4A6D DC04 DDC5 0822 2687  A086 FD3F 16CB 3141 B73A
uid                  Ryan Stone (FreeBSD) <rstone@freebsd.org>
sub   2048g/A8500B5F 2010-04-13
```

D.3.346. Søren Straarup <xride@FreeBSD.org>

```
pub   1024D/E683AD40 2006-09-28
      Key fingerprint = 8A0E 7E57 144B BC25 24A9  EC1A 0DBC 3408 E683 AD40
uid                  Soeren Straarup <xride@xride.dk>
uid                  Soeren Straarup <xride@FreeBSD.org>
uid                  Soeren Straarup <xride@x12.dk>
sub   2048g/2B18B3B8 2006-09-28
```

D.3.347. Marius Strobl <marius@FreeBSD.org>

```
pub   1024D/E0AC6F8D 2004-04-16
      Key fingerprint = 3A6C 4FB1 8BB9 4F2E BDDC  4AB6 D035 799C E0AC 6F8D
uid                  Marius Strobl <marius@FreeBSD.org>
uid                  Marius Strobl <marius@alchemy.franken.de>
sub   1024g/08BBD875 2004-04-16
```

D.3.348. Carlo Strub <cs@FreeBSD.org>

```
pub   3072R/D06F0BD7 2012-11-25 [expires: 2017-11-24]
      Key fingerprint = 61A4 F2B8 2A6C B81E 5557  0798 78E7 DE70 D06F 0BD7
uid                  Carlo Strub <cs@carlostrub.ch>
uid                  Carlo Strub <cs@FreeBSD.org>
sub   3072R/71C75997 2012-11-25 [expires: 2017-11-24]
sub   3072R/318AEB16 2012-11-25 [expires: 2017-11-24]
```

D.3.349. Cheng-Lung Sung <clsung@FreeBSD.org>

```
pub  1024D/956E8BC1 2003-09-12 Cheng-Lung Sung <clsung@FreeBSD.org>
     Key fingerprint = E0BC 57F9 F44B 46C6 DB53  8462 F807 89F3 956E 8BC1
uid                           Cheng-Lung Sung (Software Engineer) <clsung@dragon2.net>
uid                           Cheng-Lung Sung (Alumnus of CSIE, NCTU, Taiwan) <clsung@sungsung.c
uid                           Cheng-Lung Sung (AlanSung) <clsung@tiger2.net>
uid                           Cheng-Lung Sung (FreeBSD@Taiwan) <clsung@freebsd.csie.nctu.edu.tw>
uid                           Cheng-Lung Sung (Ph.D. Student of NTU.EECS) <d92921016@ntu.edu.tw>
uid                           Cheng-Lung Sung (FreeBSD Freshman) <clsung@tw.freebsd.org>
uid                           Cheng-Lung Sung (ports committer) <clsung@FreeBSD.org>
sub  1024g/1FB800C2 2003-09-12
```

D.3.350. Gregory Sutter <gsutter@FreeBSD.org>

```
pub  1024D/845DFEDD 2000-10-10 Gregory S. Sutter <gsutter@zer0.org>
     Key fingerprint = D161 E4EA 4BFA 2427 F3F9  5B1F 2015 31D5 845D FEDD
uid                            Gregory S. Sutter <gsutter@freebsd.org>
uid                            Gregory S. Sutter <gsutter@daemonnews.org>
uid                            Gregory S. Sutter <gsutter@pobox.com>
sub  2048g/0A37BBCE 2000-10-10
```

D.3.351. Koichi Suzuki <metal@FreeBSD.org>

```
pub  1024D/AE562682 2004-05-23 SUZUKI Koichi <metal@FreeBSD.org>
     Key fingerprint = 92B9 A202 B5AB 8CB6 89FC  6DD1 5737 C702 AE56 2682
sub  4096g/730E604B 2004-05-23
```

D.3.352. Ryusuke SUZUKI <ryusuke@FreeBSD.org>

```
pub   1024D/63D29724 2009-12-18
      Key fingerprint = B108 7109 2E62 BECB 0F78  FE65 1B9A D1BE 63D2 9724
uid                  Ryusuke SUZUKI <ryusuke@FreeBSD.org>
uid                  Ryusuke SUZUKI <ryusuke@jp.FreeBSD.org>
sub   1024g/5E4DD044 2009-12-18
```

D.3.353. Gary W. Swearingen <garys@FreeBSD.org>

```
pub   1024D/FAA48AD5 2005-08-22 [expires: 2007-08-22]
      Key fingerprint = 8292 CC3E 81B5 E54F E3DD  F987 FA52 E643 FAA4 8AD5
uid                  Gary W. Swearingen <garys@freebsd.org>
sub   2048g/E34C3CA0 2005-08-22 [expires: 2007-08-22]
```

D.3.354. Yoshihiro Takahashi <nyan@FreeBSD.org>

```
pub   4096R/6624859E 2012-11-18
      Key fingerprint = 1CA5 445E 7ABD BC21 AEC0  7B89 47D7 4EFF 6624 859E
uid                  Yoshihiro TAKAHASHI <nyan@furiru.org>
uid                  Yoshihiro TAKAHASHI <nyan@FreeBSD.org>
uid                  Yoshihiro TAKAHASHI <nyan@jp.FreeBSD.org>
sub   4096R/362726EA 2012-11-18
```

D.3.355. Sahil Tandon <sahil@FreeBSD.org>

```
pub   2048R/C016D977 2010-04-08
      Key fingerprint = 6AD2 BA99 8E3A 8DA6 DFC1  53CF DBD0 6001 C016 D977
uid                  Sahil Tandon <sahil@tandon.net>
```

```
uid                  Sahil Tandon <sahil@FreeBSD.org>
sub   2048R/F7776FBC 2010-04-08
```

D.3.356. TAKATSU Tomonari <tota@FreeBSD.org>

```
pub   1024D/67F58F29 2009-05-17
      Key fingerprint = 6940 B575 FC4A FA26 C094  279A 4B9B 6326 67F5 8F29
uid                  TAKATSU Tomonari <tota@FreeBSD.org>
sub   2048g/18B112CD 2009-05-17
```

D.3.357. Romain Tartière <romain@FreeBSD.org>

```
pub   3072R/5112336F 2010-04-09
      Key fingerprint = 8234 9A78 E7C0 B807 0B59  80FF BA4D 1D95 5112 336F
uid                  Romain Tartière <romain@blogreen.org>
uid                  Romain Tartière (FreeBSD) <romain@FreeBSD.org>
sub   3072R/C1B2B656 2010-04-09
sub   3072R/8F8125F4 2010-04-09
```

D.3.358. Sylvio Cesar Teixeira <sylvio@FreeBSD.org>

```
pub   2048R/AA7395A1 2009-10-28
      Key fingerprint = B319 6AAF 0016 4308 6D93  E652 3C5F 21A2 AA73 95A1
uid                  Sylvio Cesar Teixeira (My key) <sylvio@FreeBSD.org>
sub   2048R/F758F556 2009-10-28
```

D.3.359. Ion-Mihai Tetcu <itetcu@FreeBSD.org>

```
pub   4096R/29597D20 2013-05-02
      Key fingerprint = AB6F 39B6 605D E6B7 0D54  ED3D BCA2 129A 2959 7D20
uid                  Ion-Mihai Tetcu (FreeBSD Committer key) <itetcu@FreeBSD.org>
sub   4096R/EC9E17E3 2013-05-02
```

D.3.360. Mikhail Teterin <mi@FreeBSD.org>

```
pub  1024R/3FC71479 1995-09-08 Mikhail Teterin <mi@aldan.star89.galstar.com>
     Key fingerprint = 5F 15 EA 78 A5 40 6A 0F  14 D7 D9 EA 6E 2B DA A4
```

D.3.361. Gordon Tetlow <gordon@FreeBSD.org>

```
pub  1024D/357D65FB 2002-05-14 Gordon Tetlow <gordont@gnf.org>
     Key fingerprint = 34EF AD12 10AF 560E C3AE  CE55 46ED ADF4 357D 65FB
uid                            Gordon Tetlow <gordon@FreeBSD.org>
sub  1024g/243694AB 2002-05-14
```

D.3.362. Lars Thegler <lth@FreeBSD.org>

```
pub  1024D/56B0CA08 2004-05-31 Lars Thegler <lth@FreeBSD.org>
     Key fingerprint = ABAE F98C EA78 1C8D 6FDD  CB27 1CA9 5A63 56B0 CA08
uid                            Lars Thegler <lars@thegler.dk>
sub  1024g/E8C58EF3 2004-05-31
```

D.3.363. Jase Thew <jase@FreeBSD.org>

```
pub   3072R/3EEAF1EB 2012-05-30
      Key fingerprint = F5FB 959F CF1B 6550 054E  2819 A484 BCDB 3EEA F1EB
uid                  Jase Thew (FreeBSD) <jase@FreeBSD.org>
uid                  Jase Thew <freebsd@beardz.net>
```

D.3.364. David Thiel <lx@FreeBSD.org>

```
pub   1024D/A887A9B4 2006-11-30 [expires: 2011-11-29]
      Key fingerprint = F08F 6A12 738F C9DF 51AC  8C62 1E30 7CBE A887 A9B4
uid                  David Thiel <lx@FreeBSD.org>
sub   2048g/B9BD92C5 2006-11-30 [expires: 2011-11-29]
```

D.3.365. Fabien Thomas <fabient@FreeBSD.org>

```
pub   1024D/07745930 2009-03-16
      Key fingerprint = D8AC EFA2 2FBD 7788 9628  4E8D 3F35 3B88 0774 5930
uid                  Fabien Thomas <fabient@FreeBSD.org>
sub   2048g/BC173395 2009-03-16
```

D.3.366. Thierry Thomas <thierry@FreeBSD.org>

```
pub   1024D/C71405A2 1997-10-11
      Key fingerprint = 3BB8 F358 C2F1 776C 65C9  AE51 73DE 698C C714 05A2
uid                  Thierry Thomas <thierry@pompo.net>
uid                  Thierry Thomas <tthomas@mail.dotcom.fr>
uid                  Thierry Thomas (FreeBSD committer) <thierry@FreeBSD.org>
sub   1024R/C5529925 2003-11-26
sub   2048g/05CF3992 2008-02-05
```

D.3.367. Andrew Thompson <thompsa@FreeBSD.org>

```
pub   1024D/BC6B839B 2005-05-05
      Key fingerprint = DE74 3F49 B97C A170 C8F1  8423 CAB6 9D57 BC6B 839B
uid                  Andrew Thompson <thompsa@freebsd.org>
uid                  Andrew Thompson <andy@fud.org.nz>
sub   2048g/92E370FB 2005-05-05
```

D.3.368. Florent Thoumie <flz@FreeBSD.org>

```
pub   1024D/5147DCF4 2004-12-04
      Key fingerprint = D203 AF5F F31A 63E2 BFD5  742B 3311 246D 5147 DCF4
uid                  Florent Thoumie (FreeBSD committer address) <flz@FreeBSD.org>
uid                  Florent Thoumie (flz) <florent@thoumie.net>
uid                  Florent Thoumie (flz) <flz@xbsd.org>
uid                  [jpeg image of size 1796]
sub   2048g/15D930B9 2004-12-04
```

D.3.369. Jilles Tjoelker <jilles@FreeBSD.org>

```
pub   4096R/D5AE6220 2011-07-02
      Key fingerprint = 4AF5 F1CC BDD7 700B F005  79A4 A2C4 C4D4 D5AE 6220
uid                  Jilles Tjoelker <jilles@stack.nl>
uid                  Jilles Tjoelker <tjoelker@zonnet.nl>
uid                  Jilles Tjoelker (FreeBSD) <jilles@FreeBSD.org>
sub   4096R/14CB5775 2011-07-02
```

D.3.370. Ganbold Tsagaankhuu <ganbold@FreeBSD.org>

```
pub   1024D/78F6425E 2008-02-26 [expires: 2013-02-24]
      Key fingerprint = 9B8E DC41 D3F4 F7FC D8EA  417C D4F7 2AEF 78F6 425E
uid                  Ganbold <ganbold@freebsd.org>
sub   2048g/716FCBF9 2008-02-26 [expires: 2013-02-24]
```

D.3.371. Michael Tuexen <tuexen@FreeBSD.org>

```
pub   1024D/04EEDABE 2009-06-08
      Key fingerprint = 493A CCB8 60E6 5510 A01D  360E 8497 B854 04EE DABE
uid                  Michael Tuexen <tuexen@FreeBSD.org>
sub   2048g/F653AA03 2009-06-08
```

D.3.372. Andrew Turner <andrew@FreeBSD.org>

```
pub   2048R/31B31614 2010-07-01
      Key fingerprint = 08AC 2C57 F14F FDD1 2232  B5CD AA16 EFB8 31B3 1614
uid                  Andrew Turner <andrew@freebsd.org>
uid                  Andrew Turner <andrew@fubar.geek.nz>
sub   2048R/9ACBF138 2010-07-01
```

D.3.373. Hajimu UMEMOTO <ume@FreeBSD.org>

```
pub   1024D/BF9071FE 2005-03-17
      Key fingerprint = 1F00 0B9E 2164 70FC 6DC5  BF5F 04E9 F086 BF90 71FE
uid                  Hajimu UMEMOTO <ume@mahoroba.org>
uid                  Hajimu UMEMOTO <ume@FreeBSD.org>
uid                  Hajimu UMEMOTO <ume@jp.FreeBSD.org>
sub   2048g/748DB3B0 2005-03-17
```

D.3.374. Stephan Uphoff <ups@FreeBSD.org>

```
pub  2048R/D684B04A 2004-10-06 Stephan Uphoff <ups@freebsd.org>
     Key fingerprint = B5D2 04AE CA8F 7055 7474  3C85 F908 7F55 D684 B04A
uid                           Stephan Uphoff <ups@tree.com>
sub  2048R/A15F921B 2004-10-06
```

D.3.375. Bryan Venteicher <bryanv@FreeBSD.org>

```
pub   4096R/E97DB7DB 2012-11-05
      Key fingerprint = 0F8F 11EF F4D2 EDCA ECEA  CB16 744C BF25 E97D B7DB
uid                  Bryan Venteicher (DITC) <bryanv@daemonithecloset.org>
uid                  Bryan Venteicher (FreeBSD) <bryanv@freebsd.org>
sub   4096R/2EBC1A46 2012-11-05
```

D.3.376. Jacques Vidrine <nectar@FreeBSD.org>

```
pub  2048R/33C1627B 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
     Key fingerprint = CB CE 7D A0 6E 01 DC 61  E5 91 0A BE 79 17 D3 82
uid                           Jacques A. Vidrine <jvidrine@verio.net>
uid                           Jacques A. Vidrine <n@nectar.com>
uid                           Jacques A. Vidrine <jacques@vidrine.cc>
uid                           Jacques A. Vidrine <nectar@FreeBSD.org>
uid                           Jacques A. Vidrine <n@nectar.cc>

pub  1024D/1606DB95 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
     Key fingerprint = 46BC EA5B F70A CC81 5332  0832 8C32 8CFF 1606 DB95
uid                           Jacques A. Vidrine <jvidrine@verio.net>
uid                           Jacques A. Vidrine <n@nectar.com>
```

```
uid                          Jacques A. Vidrine <jacques@vidrine.cc>
uid                          Jacques A. Vidrine <nectar@FreeBSD.org>
uid                          Jacques A. Vidrine <n@nectar.cc>
sub  2048g/57EDEA6F 2001-07-05
```

D.3.377. Alberto Villa <avilla@FreeBSD.org>

```
pub   1024R/44350A8B 2010-01-24
      Key fingerprint = F740 CE4E EDDD DA9B 4A1B  1445 DF18 82EA 4435 0A8B
uid                  Alberto Villa <avilla@FreeBSD.org>
sub   1024R/F7C8254C 2010-01-24
```

D.3.378. Nicola Vitale <nivit@FreeBSD.org>

```
pub   1024D/F11699E5 2006-12-05
      Key fingerprint = 2C17 C591 2C6D 82BD F3DB  F1BF 8FC9 6763 F116 99E5
uid                  Nicola Vitale (Public key for nivit@FreeBSD.org) <nivit@FreeBSD.org>
sub   2048g/4C90805D 2006-12-05
```

D.3.379. Ivan Voras <ivoras@FreeBSD.org>

```
pub   1024D/569C05C8 2000-05-24
      Key fingerprint = AB9A A555 C47C B61D BF83  154C 95D9 C041 569C 05C8
uid                  Ivan Voras <ivoras@fer.hr>
uid                  Ivan Voras <ivan.voras@fer.hr>
uid                  Ivan Voras <ivoras@geri.cc.fer.hr>
uid                  [jpeg image of size 4567]
uid                  Ivan Voras <ivoras@sharanet.org>
uid                  Ivan Voras <ivoras@gmail.com>
uid                  Ivan Voras <ivoras@yahoo.com>
uid                  Ivan Voras <ivoras@freebsd.org>
uid                  Ivan Voras <ivan.voras@zg.t-com.hr>
sub   1536g/149FDD60 2000-05-24
```

D.3.380. Stefan Walter <stefan@FreeBSD.org>

```
pub   3072R/12B9E0B3 2003-03-06
      Key fingerprint = 85D8 6A49 22C7 6CD9 B011  5D6A 5691 111B 12B9 E0B3
uid                  Stefan Walter <stefan@freebsd.org>
uid                  Stefan Walter <sw@gegenunendlich.de>
sub   3072R/6D35457A 2003-03-06
```

D.3.381. Kai Wang <kaiw@FreeBSD.org>

```
pub   1024D/AEB910EB 2006-09-27
      Key fingerprint = 3534 10A3 F143 B760 EF3E  BEDF 8509 6A06 AEB9 10EB
uid                  Kai Wang <kaiw@FreeBSD.org>
uid                  Kai Wang <kaiw@student.chalmers.se>
uid                  Kai Wang <kaiwang27@gmail.com>
uid                  Kai Wang <kaiw27@gmail.com>
sub   2048g/1D5AA4DD 2006-09-27
```

D.3.382. Adam Weinberger <adamw@FreeBSD.org>

```
pub   2048D/C57CF3A8 2012-11-15
      Key fingerprint = CCD9 F28A BD1D 50A1 8D08  18A7 F48B B195 C57C F3A8
uid                  Adam Weinberger (FreeBSD) <adamw@FreeBSD.org>
uid                  Adam Weinberger (adamw.org) <adamw@adamw.org>
sub   2048g/9C6D0E30 2012-11-15
```

D.3.383. Peter Wemm <peter@FreeBSD.org>

```
pub  1024D/7277717F 2003-12-14 Peter Wemm <peter@wemm.org>
     Key fingerprint = 622B 2282 E92B 3BAB 57D1  A417 1512 AE52 7277 717F
uid                            Peter Wemm <peter@FreeBSD.ORG>
sub  1024g/8B40D9D1 2003-12-14
pub  1024R/D89CE319 1995-04-02 Peter Wemm <peter@netplex.com.au>
     Key fingerprint = 47 05 04 CA 4C EE F8 93  F6 DB 02 92 6D F5 58 8A
uid                            Peter Wemm <peter@perth.dialix.oz.au>
uid                            Peter Wemm <peter@haywire.dialix.com>
```

D.3.384. Nathan Whitehorn <nwhitehorn@FreeBSD.org>

```
pub   1024D/FC118258 2008-07-03
      Key fingerprint = A399 BEA0 8D2B 63B3 47B5  056D 8513 5B96 FC11 8258
uid                  Nathan Whitehorn <nwhitehorn@freebsd.org>
uid                  Nathan Whitehorn <nwhitehorn@icecube.wisc.edu>
uid                  Nathan Whitehorn <nwhitehorn@physics.wisc.edu>
uid                  Nathan Whitehorn <whitehorn@wisc.edu>
sub   2048g/EDB55363 2008-07-03
```

D.3.385. Martin Wilke <miwi@FreeBSD.org>

```
pub   1024D/B1E6FCE9 2009-01-31
      Key fingerprint = C022 7D60 F598 8188 2635  0F6E 74B2 4884 B1E6 FCE9
uid                  Martin Wilke <miwi@FreeBSD.org>
sub   4096g/096DA69D 2009-01-31
```

D.3.386. Nate Williams <nate@FreeBSD.org>

```
pub  1024D/C2AC6BA4 2002-01-28 Nate Williams (FreeBSD) <nate@FreeBSD.org>
     Key fingerprint = 8EE8 5E72 8A94 51FA EA68  E001 FFF9 8AA9 C2AC 6BA4
sub  1024g/03EE46D2 2002-01-28
```

D.3.387. Steve Wills <swills@FreeBSD.org>

```
pub   2048R/207B1BA1 2010-09-02 [expires: 2011-09-02]
      Key fingerprint = 98FA 414A 5C2A 0EF9 CFD0  AD0D F5CF 62B3 207B 1BA1
uid                  Steve Wills <swills@freebsd.org>
uid                  Steve Wills <steve@mouf.net>
sub   2048R/E9B254FD 2010-09-02 [expires: 2011-09-02]
```

D.3.388. Thomas Wintergerst <twinterg@FreeBSD.org>

```
pub   1024D/C45CB978 2006-01-08
      Key fingerprint = 04EE 8114 7C6D 22CE CDC8  D7F8 112D 01DB C45C B978
uid                  Thomas Wintergerst <twinterg@gmx.de>
uid                  Thomas Wintergerst <twinterg@freebsd.org>
uid                  Thomas Wintergerst
uid                  Thomas Wintergerst <thomas.wintergerst@nord-com.net>
uid                  Thomas Wintergerst <thomas.wintergerst@materna.de>
sub   2048g/3BEBEF8A 2006-01-08
sub   1024D/8F631374 2006-01-08
sub   2048g/34F631DC 2006-01-08
```

D.3.389. Garrett Wollman <wollman@FreeBSD.org>

```
pub  1024D/0B92FAEA 2000-01-20 Garrett Wollman <wollman@FreeBSD.org>
     Key fingerprint = 4627 19AF 4649 31BF DE2E  3C66 3ECF 741B 0B92 FAEA
sub  1024g/90D5EBC2 2000-01-20
```

D.3.390. Jörg Wunsch <joerg@FreeBSD.org>

```
pub  1024D/69A85873 2001-12-11 Joerg Wunsch <j@uriah.heep.sax.de>
     Key fingerprint = 5E84 F980 C3CA FD4B B584  1070 F48C A81B 69A8 5873
pub  1024D/69A85873 2001-12-11 Joerg Wunsch <j@uriah.heep.sax.de>
uid                            Joerg Wunsch <joerg_wunsch@interface-systems.de>
uid                            Joerg Wunsch <joerg@FreeBSD.org>
uid                            Joerg Wunsch <j@ida.interface-business.de>
sub  1024g/21DC9924 2001-12-11
```

D.3.391. David Xu <davidxu@FreeBSD.org>

```
pub   1024D/48F2BDAB 2006-07-13 [expires: 2009-07-12]
      Key fingerprint = 7182 434F 8809 A4AF 9AE8  F1B5 12F6 3390 48F2 BDAB
uid                  David Xu <davidxu@freebsd.org>
sub   4096g/ED7DB38A 2006-07-13 [expires: 2009-07-12]
```

D.3.392. Maksim Yevmenkin <emax@FreeBSD.org>

```
pub  1024D/F050D2DD 2003-10-01 Maksim Yevmenkin <m_evmenkin@yahoo.com>
     Key fingerprint = 8F3F D359 E318 5641 8C81  34AD 791D 53F5 F050 D2DD
```

D.3.393. Bjoern A. Zeeb <bz@FreeBSD.org>

```
pub   1024D/3CCF1842 2007-02-20
      Key fingerprint = 1400 3F19 8FEF A3E7 7207  EE8D 2B58 B8F8 3CCF 1842
uid                  Bjoern A. Zeeb <bz@zabbadoz.net>
uid                  Bjoern A. Zeeb <bzeeb@zabbadoz.net>
uid                  Bjoern A. Zeeb <bz@FreeBSD.org>
uid                  Bjoern A. Zeeb <bzeeb-lists@lists.zabbadoz.net>
sub   4096g/F36BDC5D 2007-02-20
```

D.3.394. Niclas Zeising <zeising@FreeBSD.org>

```
pub   4096R/EA4BF1EC 2012-11-28 [expires: 2013-12-31]
      Key fingerprint = A8DE D126 D346 E9CB 6176  AECB 0401 4392 EA4B F1EC
uid                  Niclas Zeising <zeising@daemonic.se>
uid                  Niclas Zeising (FreeBSD Project) <zeising@freebsd.org>
uid                  Niclas Zeising (Lysator ACS) <zeising@lysator.liu.se>
sub   4096R/BB8B5551 2012-11-29 [expires: 2013-12-31]
sub   4096R/B8D43CD2 2012-11-29 [expires: 2013-12-31]
```

D.3.395. Alexey Zelkin <phantom@FreeBSD.org>

```
pub  1024D/9196B7D9 2002-01-28 Alexey Zelkin <phantom@FreeBSD.org>
     Key fingerprint = 4465 F2A4 28C1 C2E4 BB95  1EA0 C70D 4964 9196 B7D9
sub  1024g/E590ABA4 2002-01-28
```

D.3.396. Sepherosa Ziehau <sephe@FreeBSD.org>

```
pub   2048R/3E51FB42 2005-10-21
      Key fingerprint = 5F47 3861 7ABA 8773 9E32  0474 5C33 841C 3E51 FB42
uid                  Sepherosa Ziehau (freebsd) <sephe@freebsd.org>
uid                  Sepherosa Ziehau (sephe) <sepherosa@gmail.com>
```

```
sub   2048R/7AA31321 2005-10-21
```

D.3.397. Andrey Zonov <zont@FreeBSD.org>

```
pub   2048R/E8A68B1C 2012-08-17 [expires: 2016-08-17]
      Key fingerprint = 3DFF AA2F C10A A979 2FB9  A764 F145 4BB6 E8A6 8B1C
uid                  Andrey Zonov <zont@FreeBSD.org>
uid                  Andrey Zonov <andrey@zonov.org>
sub   2048R/57FC2BD3 2012-08-17 [expires: 2016-08-17]
```

FreeBSD Glossar

Dieser Abschnitt enthält Begriffe und Abkürzungen, die innerhalb des FreeBSD-Projekts sowie der zugehörigen Dokumentation verwendet werden.

A

ACL

Siehe: Access Control List

ACPI

Siehe: Advanced Configuration and Power Interface

AMD

Siehe: Automatic Mount Daemon

AML

Siehe: ACPI Machine Language

API

Siehe: Application Programming Interface

APIC

Siehe: Advanced Programmable Interrupt Controller

APM

Siehe: Advanced Power Management

APOP

Siehe: Authenticated Post Office Protocol

ASL

Siehe: ACPI Source Language

ATA

Siehe: Advanced Technology Attachment

ATM

Siehe: Asynchronous Transfer Mode

ACPI Machine Language

Pseudocode, der von einer virtuellen Maschine innerhalb eines ACPI-konformen Betriebssystems ausgeführt wird. Bietet eine Verbindungsschicht (*Layer*) zwischen der verwendeten Hardware und der dokumentierten Schnittstelle, auf die das Betriebssystem zugreift.

ACPI Source Language

Die Programmiersprache, in der die AML geschrieben ist.

Access Control List

Eine Liste von Zugriffsrechten, die einem Objekt, normalerweise eine Datei oder ein Gerät im Netzwerk, angehängt ist.

Advanced Configuration and Power Interface

Eine Spezifikation, die eine Abstrahierung der Schnittstelle darstellt, die Hardware und Betriebssystem verbindet. Dadurch benötigt das Betriebssystem keine Informationen über die vorhandene Hardware, um diese einsetzen zu können. ACPI ist eine Weiterentwicklung von APM, PNPBIOS und anderen Technologien und bietet Funktionen zur Kontrolle des Energieverbrauchs, zur Versetzung von Rechnern in den Ruhezustand, zur Aktivierung und Deaktivierung von Geräten und andere mehr.

Application Programming Interface

Eine Sammlung von Prozeduren, Protokollen und Werkzeugen, die das Zusammenspiel von verschiedenen Programmteilen festlegt. Wie, wann und warum arbeiten sie zusammen, welche Daten werden zwischen ihnen ausgetauscht und anderes mehr.

Advanced Power Management

Eine API, die es dem Betriebssystem ermöglicht, zusammen mit dem BIOS die Stromversorgung zu verwalten. APM wurde für die meisten Anwendungen durch die allgemeinere und leistungsfähigere ACPI Spezifikation abgelöst.

Advanced Programmable Interrupt Controller

Advanced Technology Attachment

Asynchronous Transfer Mode

Authenticated Post Office Protocol

Automatic Mount Daemon

Ein Daemon, der ein Dateisystem automatisch einhängt, wenn auf eine Datei oder ein Verzeichnis dieses Dateisystems zugegriffen wird.

B

BAR

Siehe: Base Address Register

BIND

Siehe: Berkeley Internet Name Domain

BIOS

Siehe: Basic Input/Output System

BSD

Siehe: Berkeley Software Distribution

Base Address Register

Register, die den zu einem PCI-Gerät gehörenden Adressbereich festlegen.

Basic Input/Output System

Die Bedeutung des Begriffs BIOS hängt vom Kontext ab, in dem es verwendet wird. Einmal wird damit der ROM-Chip bezeichnet, der über einen Basisbefehlssatz eine Schnittstelle zwischen Hard- und Software schafft. Aber auch die Routinen, die in diesen Chip implementiert wurden, und die dabei helfen, Ihr System zu starten, werden als BIOS bezeichnet. Und nicht zuletzt wird manchmal die Bildschirmmaske, über die der Bootprozess konfiguriert werden kann, ebenfalls als BIOS bezeichnet. Der Begriff BIOS ist zwar PC-spezifisch, andere Systeme verfügen aber über ähnliche Systeme.

Berkeley Internet Name Domain

Eine Implementierung des DNS-Protokolls.

Berkeley Software Distribution

Diesen Namen gab die Computer Systems Research Group (CSRG) der The University of California at Berkeley (http://www.berkeley.edu) den Verbesserungen und Änderungen an AT&Ts 32V UNIX. FreeBSD beruht auf der Arbeit der CSRG.

Bikeshed Building

Die Beobachtung, dass viele Leute Meinungen zu unkomplizierten Themen äußern, während gleichzeitig über ein kompliziertes Thema gar nicht oder nur wenig diskutiert wird. Die Herkunft des Ausdrucks wird in den FAQ (http://www.FreeBSD.org/doc/de_DE.ISO8859-1/books/faq/misc.html#BIKESHED-PAINTING) erläutert.

C

CD

Siehe: Carrier Detect

CHAP

Siehe: Challenge Handshake Authentication Protocol

CLIP

Siehe: Classical IP over ATM

COFF

Siehe: Common Object File Format

CPU

Siehe: Central Processing Unit

CTS

Siehe: Clear To Send

CVS

Siehe: Concurrent Versions System

Carrier Detect

Ein RS232C-Signal. Notwendig, um eine serielle Verbindung aufbauen zu können.

Central Processing Unit

Auch als Prozessor bekannt. Dieser stellt das Gehirn eines Computers dar, in dem alle Berechnungen erfolgen. Es gibt verschiedene Prozessor-Architekturen, die über verschiedene Befehlssätze verfügen, beispielsweise Intel-x86-, Sun SPARC-, PowerPC- und Alpha-Systeme.

Challenge Handshake Authentication Protocol

Eine Vorgehensweise, einen Benutzer anhand eines Geheimnisses zu authentisieren, dass zwischen Client und Server ausgetauscht wird.

Classical IP over ATM

Clear To Send

Ein RS232C-Signal. Das entfernte System erhält durch dieses Signal die Erlaubnis, Daten zu senden.

Siehe auch: Request To Send.

Common Object File Format

Concurrent Versions System

Ein Versionskontrollsystem, das es erlaubt, mit vielen verschiedenen Versionen einer Datei zu arbeiten und die an diesen Dateien durchgeführten Änderungen zu verfolgen. CVS ermöglicht es, individuelle Änderungen durchzuführen, in ein Repository einzubringen und auch wieder rückgängig zu machen. Außerdem ist es möglich, nachzuvollziehen, welche Änderungen wann, von wem und warum erfolgten.

D

DAC

Siehe: Discretionary Access Control

DDB

Siehe: Debugger

DES

Siehe: Data Encryption Standard

DHCP

Siehe: Dynamic Host Configuration Protocol

DNS

Siehe: Domain Name System

DSDT

Siehe: Differentiated System Description Table

DSR

Siehe: Data Set Ready

DTR

Siehe: Data Terminal Ready

DVMRP

Siehe: Distance-Vector Multicast Routing Protocol

Discretionary Access Control

Data Encryption Standard

Eine Methode zur Verschlüsselung von Informationen. Wird traditionellerweise zur Verschlüsselung von UNIX-Passwörtern und von crypt(3) verwendet.

Data Set Ready

Ein RS232C-Signal, das von einem Modem an einen Computer oder ein Terminal geschickt wird und die Sende- und Empfangsbereitschaft des Modems meldet.

Siehe auch: Data Terminal Ready.

Data Terminal Ready

Ein RS232C-Signal, das von einem Computer oder einem Terminal an das Modem geschickt wird und die Sende- und Empfangsbereitschaft des Computers oder des Terminals meldet.

Debugger

Eine interaktive, in den Kernel eingebaute Funktion, um den Status eines Systems zu untersuchen. Wird in der Regel nach einem Systemabsturz eingesetzt, um die Ursache für den Absturz zu finden.

Differentiated System Description Table

Eine ACPI-Tabelle, die Informationen über die Konfiguration des Basissystems enthält.

Distance-Vector Multicast Routing Protocol

Domain Name System

Das System, dass Klartext-Rechnernamen (wie mail.example.net) in Internet-IP-Adressen (oder umgekehrt) konvertiert.

Dynamic Host Configuration Protocol

Ein Protokoll, das auf Anforderung dynamisch eine IP-Adresse an einen Rechner vergibt. Diese Adresszuweisung wird als “Lease” bezeichnet.

E

ECOFF

Siehe: Extended COFF

ELF

Siehe: Executable and Linking Format

ESP

Siehe: Encapsulated Security Payload

Encapsulated Security Payload

Executable and Linking Format

Extended COFF

F

FADT

Siehe: Fixed ACPI Description Table

FAT

Siehe: File Allocation Table

FAT16

Siehe: File Allocation Table (16-bit)

FTP

Siehe: File Transfer Protocol

File Allocation Table

File Allocation Table (16-bit)

File Transfer Protocol

Ein auf TCP aufsetzendes Protokoll, das zum Transfer von Daten über ein TCP/IP-Netzwerk verwendet wird.

Fixed ACPI Description Table

G

GUI

Siehe: Graphical User Interface

Giant

Der Name für einen wechselseitigen Ausschluss (*mutual exclusion*), der einen großen Teil der Kernel-Ressourcen schützt. Zu Zeiten, als auf einer Maschine nur ein paar Prozesse liefen und die Maschine nur eine Netzwerkkarte und insbesondere nur einen Prozessor besaß, war dieser einfache Mechanismus zum Verriegeln (*lock*) einer Ressource völlig ausreichend. Heutzutage entstehen durch den wechselseitigen Ausschluss Geschwindigkeitsengpässe. Die FreeBSD-Entwickler arbeiten daran, Giant durch Locks zu ersetzten, die einzelne Ressourcen schützen. Auf Einprozessor- und Mehrprozessor-Maschinen können dadurch mehr Prozesse parallel ausgeführt werden.

Graphical User Interface

Eine grafische Oberfläche, über die der Anwender mit dem System interagiert.

H

HTML

Siehe: HyperText Markup Language

HUP

Siehe: HangUp

HangUp

HyperText Markup Language

Die Auszeichnungssprache, mit der Internetseite erstellt werden können.

I

I/O

Siehe: Input/Output

IASL

Siehe: Intel’s ASL-Compiler

IMAP

Siehe: Internet Message Access Protocol

IP

Siehe: Internet Protocol

IPFW

Siehe: IP Firewall

IPP

Siehe: Internet Printing Protocol

IPv4

Siehe: IP Version 4

IPv6

Siehe: IP Version 6

ISP

Siehe: Internet Service Provider

IP Firewall

IP Version 4

Die IP-Protokollversion 4, die 32-Bit-Adressen einsetzt. Diese Version stellt derzeit noch den in der Praxis am meisten verwendeten Standard dar, sollt aber sukzessive durch IPv6 ersetzt werden.

Siehe auch: IP Version 6.

IP Version 6

Das neue IP-Protokoll. Es wurde entwickelt, weil der Adressraum von IPv4 nicht mehr ausreichend ist. IPv6 verwendet 128-Bit-Adressen.

Input/Output

Intel's ASL-Compiler

Intel's Compiler zur Konvertierung von ASL nach AML.

Internet Message Access Protocol

Ein Protokoll für den Zugriff auf einen E-Mail-Server. Charakteristisch für dieses Protokoll ist, dass die Nachrichten in der Regel auf dem Server verbleiben und nicht vom E-Mail-Client heruntergeladen werden.

Siehe auch: Post Office Protocol Version 3.

Internet Printing Protocol

Internet Protocol

Das Standardprotokoll zur Paketübertragung im Internet. Wurde ursprünglich vom U.S. Department of Defense entwickelt, und ist ein zentraler Bestandteile des TCP/IP-Stacks. Ohne das Internet Protocol wäre das Internet in der heutigen Form nicht möglich. Das Internet Protocol ist im RFC 791 (ftp://ftp.rfc-editor.org/in-notes/rfc791.txt) definiert.

Internet Service Provider

Ein Unternehmen, das anderen den Zugang zum Internet ermöglicht.

K

KAME

Japanisch für "Schildkröte". Der Begriff KAME wird in Computerkreisen für das KAME Project (http://www.kame.net/) verwendet, das an einer IPv6-Implementierung arbeitet.

KDC

Siehe: Key Distribution Center

KLD

Siehe: Kernel ld(1)

KSE

Siehe: Kernel Scheduler Entities

KVA

Siehe: Kernel Virtual Address

Kbps

Siehe: Kilo Bits Per Second

Kernel ld(1)

Eine Methode, um den Kernel dynamisch um zusätzliche Funktionen zu erweitern, ohne das System neu zu starten.

Kernel Scheduler Entities

Threads, die im Kernel laufen. Näheres entnehmen Sie der Home-Page des Projekts (http://www.FreeBSD.org/kse).

Kernel Virtual Address

Key Distribution Center

Kilo Bits Per Second

Maßeinheit, in der die Bandbreite (also die Menge der Daten, die in einer bestimmten Zeit übertragen werden kann) angegeben wird. Statt Kilo können auch Mega, Giga, Tera und weitere Präfixe verwendet werden.

L

LAN

Siehe: Local Area Network

LOR

Siehe: Lock Order Reversal

LPD

Siehe: Line Printer Daemon

Line Printer Daemon

Local Area Network

Ein Netzwerk, das nur in einem lokalen Bereich, wie einem Büro, einen Unternehmen oder einem Haus, eingesetzt wird.

Lock Order Reversal

Der FreeBSD-Kernel benutzt eine Reihe von Ressource-Locks, um den Zugriff auf Ressourcen zu regeln. In FreeBSD-CURRENT-Kerneln (nicht in Release-Kerneln) befindet sich das Diagnose-System witness(4), das Verklemmungen (*deadlock*) zur Laufzeit erkennt. witness(4) ist vorsichtig: daher gibt es schon mal Falschmeldungen aus. Eine richtig erkannte Verklemmung bedeutet soviel wie “Wenn Sie Pech gehabt hätten, wäre es jetzt zu einer Verklemmung gekommen”.

Richtig erkannte Verklemmungen (LOR) werden schnell behoben. Prüfen Sie daher http://lists.FreeBSD.org/mailman/listinfo/freebsd-current und die Seite LORs Seen (http://sources.zabbadoz.net/freebsd/lor.html) bevor Sie die Mailinglisten kontaktieren.

M

MAC

Siehe: Mandatory Access Control

MADT

Siehe: Multiple APIC Description Table

MFC

Siehe: Merge From Current

MFP4

Siehe: Merge From Perforce

MFS

Siehe: Merge From Stable

MIT

Siehe: Massachusetts Institute of Technology

MLS

Siehe: Multi-Level Security

MOTD

Siehe: Message Of The Day

MTA

Siehe: Mail Transfer Agent

MUA

Siehe: Mail User Agent

Mail Transfer Agent

Eine Anwendung zum Transfer von E-Mails. Ein MTA war von jeher im BSD-Basissystem enthalten. Aktuell handelt es sich dabei um Sendmail. Es exisitieren aber auch zahlreiche andere MTAs, darunter postfix, qmail und Exim.

Mail User Agent

Ein Programm zur Anzeige und zum Verfassen von E-Mails.

Mandatory Access Control

Massachusetts Institute of Technology

Merge From Current

Das Einbringen von Funktionen oder Fehlerbehebungen aus dem -CURRENT-Zweig in einen anderen Zweig, meist -STABLE.

Merge From Perforce

Das Einbringen von Funktionen oder Fehlerbehebungen aus dem Perforce-Repository des -CURRENT-Zweigs.

Siehe auch: Perforce.

Merge From Stable

Normalerweise werden Änderungen an FreeBSD zuerst im -CURRENT-Zweig getestet und dann in den -STABLE-Zweig übernommen. Selten kommt es vor, dass eine Änderung zuerst im -STABLE-Zweig vorgenommen wird und anschließend im -CURRENT-Zweig übernommen wird.

Dieser Ausdruck wird auch benutzt, wenn eine Fehlerbehebung von -STABLE in einem der Sicherheitszweige übernommen wird.

Siehe auch: Merge From Current.

Message Of The Day

Eine Nachricht, die in der Regel beim Anmelden an einem System angezeigt wird. Enthält häufig Informationen für die Benutzer des Systems.

Multi-Level Security

Multiple APIC Description Table

N

NAT

Siehe: Network Address Translation

NDISulator

Siehe: Project Evil

NFS

Siehe: Network File System

NTFS

Siehe: New Technology File System

NTP

Siehe: Network Time Protocol

Network Address Translation

Eine Technik, bei der IP-Pakete auf dem Weg durch ein Gateway umgeschrieben werden. Dadurch wird es möglich, dass sich mehrere Rechner hinter dem Gateway eine einzige IP-Addresse teilen.

Network File System

New Technology File System

Ein von Microsoft entwickeltes Dateisystem, das in dessen “New Technology”-Betriebssystemen, wie Windows 2000, Windows NT und Windows XP, eingesetzt wird.

Network Time Protocol

Ein Protokoll, um die Systemzeit über ein Netzwerk zu synchronisieren.

O

OBE

Siehe: Overtaken By Events

ODMR

Siehe: On-Demand Mail Relay

OS

Siehe: Operating System

On-Demand Mail Relay

Operating System

Eine Sammlung von Programmen, Bibliotheken und Werkzeugen, die den Zugriff auf die Hardware eines Computers erlauben. Die Bandbreite aktueller Betriebssysteme reicht von einfachen Designs, die lediglich die Ausführung eines einzigen Programms und die Nutzung eines einzigen Geräts zur gleichen Zeit erlauben bis hin zu Multitasking- und Multiprozess-Systemen, die gleichzeitig Tausende Benutzer bedienen können, von denen jeder wiederum Dutzende Programme laufen lassen kann.

Overtaken By Events

Zeigt an, dass eine gewünschte Änderung (aus einem Fehlerbericht oder einer Anforderung) überholt ist. Die Ursache können beispielsweise spätere Änderungen in FreeBSD, geänderte Netzwerk-Standards oder jetzt veraltete Hardware sein.

P

p4

Siehe: Perforce

PAE

Siehe: Physical Address Extensions

PAM

Siehe: Pluggable Authentication Modules

PAP

Siehe: Password Authentication Protocol

PC

Siehe: Personal Computer

PCNSFD

Siehe: Personal Computer Network File System Daemon

PDF

Siehe: Portable Document Format

PID

Siehe: Process ID

POLA

Siehe: Principle Of Least Astonishment

POP

Siehe: Post Office Protocol

POP3

Siehe: Post Office Protocol Version 3

PPD

Siehe: PostScript Printer Description

PPP

Siehe: Point-to-Point Protocol

PPPoA

Siehe: PPP over ATM

PPPoE

Siehe: PPP over Ethernet

PPP over ATM

PPP over Ethernet

PR

Siehe: Problem Report

PXE

Siehe: Preboot eXecution Environment

Password Authentication Protocol

Perforce

Ein von Perforce Software (http://www.perforce.com/) entwickeltes Versionskontrollsystem, das mehr Funktionen als CVS aufweist. Obwohl es sich dabei nicht um Open-Source handelt, dürfen Open-Source-Projekte wie FreeBSD die Software kostenlos einsetzen.

Einige FreeBSD-Entwickler verwenden ein Perforce-Repository, um Quellcode zu verwalten, der selbst für den -CURRENT-Zweig zu experimentell ist.

Personal Computer

Personal Computer Network File System Daemon

Physical Address Extensions

Eine Möglichkeit, um auf Systemen, die physikalisch nur über einen 32-Bit-Adressraum verfügen, bis zu 64 GB RAM ansprechen zu können. Ohne PAE wären diese Systeme auf maximal 4 GB Hauptspeicher beschränkt.

Pluggable Authentication Modules

Point-to-Point Protocol

Pointy Hat

Ein Kopfschmuck, ähnlich den Eselsohren, der FreeBSD-Committern gereicht wird, wenn sie den Bau kaputtmachen, Revisionsnummern verkleinern oder sonstigen Schaden im Quellbaum anrichten. Jeder Committer, der etwas taugt, besitzt schnell eine stattliche Sammlung. Der Begriff wird (meist?) scherzhaft verwendet.

Portable Document Format

Post Office Protocol

Siehe auch: Post Office Protocol Version 3.

Post Office Protocol Version 3

Ein Protokoll für den Zugriff auf einen E-Mail-Server. Dadurch gekennzeichnet, dass neue Nachrichten vom E-Mail-Client heruntergeladen und nicht auf dem Server verbleiben.

Siehe auch: Internet Message Access Protocol.

PostScript Printer Description

Preboot eXecution Environment

Principle Of Least Astonishment

Prinzip der kleinsten Überraschung

Änderungen an FreeBSD sollten nach Möglichkeit für den Benutzer nachvollziehbar sein. Das willkürliche Umordnen der Variablen in `/etc/defaults/rc.conf` verletzt zum Beispiel dieses Prinzip. Entwickler beachten das Prinzip, wenn Sie über für Benutzer sichtbare Änderungen nachdenken.

Problem Report

Die Beschreibung eines Problems, das im FreeBSD-Quellcode oder in der Dokumentation gefunden wurde. Lesen Sie dazu auch den Artikel Writing FreeBSD Problem Reports (http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/problem-reports/index.html).

Process ID

Eine eindeutige Zahl, die einem Prozess zugewiesen ist. Identifiziert den Prozess und erlaubt es, diesen Prozess zu bearbeiten.

Project Evil

Der Arbeitstitel des von Bill Paul geschriebenen NDISulator. Der Name bezieht sich darauf, dass es (philosophisch gesehen) schlimm ist, einen solchen Treiber überhaupt schreiben zu müssen. Der NDISulator ist ein Kompatibilitätsmodul, das es erlaubt, Microsoft Windows™ NDIS-Miniport-Netzwerktreiber mit FreeBSD/i386 zu benutzen. Für gewöhnlich ist dies die einzige Möglichkeit, Karten mit einem Treiber, dessen Quellen verschlossen sind, zu benutzen. Siehe `src/sys/compat/ndis/subr_ndis.c`.

R

RA

Siehe: Router Advertisement

RAID

Siehe: Redundant Array of Inexpensive Disks

RAM

Siehe: Random Access Memory

RD

Siehe: Received Data

RFC

Siehe: Request For Comments

RISC

Siehe: Reduced Instruction Set Computer

RPC

Siehe: Remote Procedure Call

RS232C

Siehe: Recommended Standard 232C

RTS

Siehe: Request To Send

Random Access Memory

Revision Control System

Das *Revision Control System* (RCS) ist eines der ältesten “Versionsverwaltungssysteme” für reine Textdateien. Es erlaubt das Speichern, Laden, Archivieren, Protokollieren, Identifizieren sowie das Zusammenführen von verschiedenen Revisionen einer Datei. Bei RCS handelt es sich um eine Sammlung von vielen kleinen zusammenarbeitenden Werkzeugen. Zwar fehlen im Vergleich zu CVS oder Subversion einige Funktionen, allerdings ist RCS sehr einfach zu installieren, zu konfigurieren und zu benutzen, solange die Anzahl der zu verwaltenden Dateien überschaubar bleibt. RCS ist dabei für praktisch alle wichtigen UNIX-artigen Betriebssysteme verfügbar.

Siehe auch: Concurrent Versions System, Subversion.

Received Data

Ein RS232C-Pin oder -Draht, über den neue Daten ankommen.

Siehe auch: Transmitted Data.

Recommended Standard 232C

Ein Standard für die Kommunikation zwischen seriellen Geräten.

Reduced Instruction Set Computer

Ein Ansatz im Prozessordesign, bei dem die von der Hardware durchzuführenden Operationen so weit als möglichst vereinfacht und verallgemeinert werden. Vorteile dieses Design sind ein geringerer Energieverbrauch, eine geringere Transistoranzahl und übersichtlicherer Code. Zu den RISC-Plattformen gehören Alpha, SPARC, ARM sowie PowerPC.

Redundant Array of Inexpensive Disks

Remote Procedure Call

repocopy

Siehe: Repository Copy

Repository Copy

Eine direkte Kopie von Dateien innerhalb eines Repositories.

Ohne eine Repocopy müsste ein Committer eine Datei mit `cvs add` an der neuen Position einfügen und mit `cvs rm` an der alten Position löschen.

Der Nachteil dieser Methode wäre allerdings, dass dabei die Datei-Historie (also die CVS-Logs) nicht an die neue Position kopiert werden würde. Da das FreeBSD-Project diese Informationen als äßerst nützlich ansieht, wird stattdessen häfig eine Repocopy durchgeführt. Bei diesem Prozess kopiert ein Repositoy Meister die Datei direkt innerhalb des Repository an die neue Position, statt cvs(1) einzusetzen.

Request For Comments

Eine Sammlung von Dokumenten, die wichtige Internetstandards, Protokolle und so weiter definieren und die unter www.rfc-editor.org (http://www.rfc-editor.org/) zu finden sind.

Kann aber auch allgemein verwendet werden, wenn jemand eine Änderung vorschlägt und dazu Feedback möchte.

Request To Send

Ein RS232C-Signal, das der Gegenstelle signalisiert, dass sie mit dem Senden der Daten beginnen kann.

Siehe auch: Clear To Send.

Router Advertisement

S

SCI

Siehe: System Control Interrupt

SCSI

Siehe: Small Computer System Interface

SG

Siehe: Signal Ground

SMB

Siehe: Server Message Block

SMP

Siehe: Symmetric MultiProcessor

SMTP

Siehe: Simple Mail Transfer Protocol

SMTP AUTH

Siehe: SMTP Authentication

SSH

Siehe: Secure Shell

STR

Siehe: Suspend To RAM

SVN

Siehe: Subversion

SMTP Authentication

Server Message Block

Signal Ground

Ein RS232-Pin oder -Draht, der als Untergrundreferenz für das Signal verwendet wird.

Simple Mail Transfer Protocol

Secure Shell

Small Computer System Interface

Subversion

Subversion ist ein Versionskontrollsystem, ähnlich wie CVS, aber mit einer grösseren Liste von Eigenschaften.

Siehe auch: Concurrent Versions System.

Suspend To RAM

Symmetric MultiProcessor

System Control Interrupt

T

TCP

Siehe: Transmission Control Protocol

TCP/IP

Siehe: Transmission Control Protocol/Internet Protocol

TD

Siehe: Transmitted Data

TFTP

Siehe: Trivial FTP

TGT

Siehe: Ticket-Granting Ticket

TSC

Siehe: Time Stamp Counter

Ticket-Granting Ticket

Time Stamp Counter

Ein interner Zähler bei modernen Pentium-Prozessoren, der die Ticks der *core frequency clock* bestimmt.

Transmission Control Protocol

Ein Protokoll, das auf dem IP-Protokoll aufsetzt. Es garantiert, dass Datenpakete zuverlässig und geordnet transportiert werden.

Transmission Control Protocol/Internet Protocol

Die Kombination aus TCP- und IP-Protokoll. Ein Großteil des Internets basiert auf TCP/IP.

Transmitted Data

Ein RS232C-Pin oder -Draht, über den Daten verschickt werden.

Siehe auch: Received Data.

Trivial FTP

U

UDP

Siehe: User Datagram Protocol

UFS1

Siehe: Unix File System Version 1

UFS2

Siehe: Unix File System Version 2

UID

Siehe: User ID

URL

Siehe: Uniform Resource Locator

USB

Siehe: Universal Serial Bus

Uniform Resource Locator

Eine Methode um eine Ressource, z.B. ein Dokument im Internet, zu lokalisieren und eine Art, diese Ressource zu identifizieren.

Unix File System Version 1

Das Original UNIX Dateisystem, manchmal auch das Berkeley Fast File System genannt.

Unix File System Version 2

Eine Erweiterung für UFS1, eingeführt in FreeBSD5-CURRENT. UFS2 enthält 64-bit Blockzeiger (durchbricht dadurch die 1T Grenze), Unterstützung für extended file storage und andere Merkmale.

Universal Serial Bus

Ein Hardware-Standard, der verwendet wird um eine grosse Vielfalt von Computerperipherie an eine einheitliche Schnittstelle anzuschliessen.

User ID

Eine eindeutige Nummer, die einem Benutzer eines Computers zugewiesen wird. Kann zur Identifizierung von zugewiesenen Ressourcen und Berechtigungen verwendet werden.

User Datagram Protocol

Ein einfaches, nicht-zuverlässiges Protokoll für Datagramme, das beim Datenaustausch in einem TCP/IP Netzwerk benutzt wird. UDP enthält keine Fehlerüberprüfung und -korrektur wie TCP.

V

VPN

Siehe: Virtual Private Network

Virtual Private Network

Eine Methode ein öffentliches Netzwerk wie das Internet zu nutzen, um einen entfernten Zugriff auf ein lokales Netz, wie etwa ein Unternehmens-LAN, zu ermöglichen.

Kolophon

Dieses Buch ist aus den Beiträgen vieler Freiwilliger zum “FreeBSD Documentation Project” entstanden. Der Text ist in SGML entsprechend der Docbook DTD verfasst. Mit Hilfe von **Jade**, einem Open Source DSSSL-Prozessor, wird er in verschiedene Formate umgewandelt. Die Umwandlung wird von Norm Walsh’s DSSSL Stylesheets und eigens entwickelten Stylesheets gesteuert. Die gedruckte Ausgabe des Buchs wäre ohne die Satzbeschreibungssprache TEX von Donald Knuth, LATEX von Leslie Lamport oder den **JadeTeX**-Makros von Sebastian Rahtz nicht möglich.