Linux® 用户的FreeBSD 快速入门向导

John Ferrell

$FreeBSD: release/8.4.0/zh_CN.GB2312/articles/linux-users/article.xml 39632 2012-10-01 11:56:00Z gabor $

$FreeBSD: release/8.4.0/zh_CN.GB2312/articles/linux-users/article.xml 39632 2012-10-01 11:56:00Z gabor $

本文档旨在快速使那些高级Linux® 用户熟悉FreeBSD的一些基础知识。

目录

1 简介

本文档将突出介绍FreeBSD 与Linux 的差别，以使得那些Linux 高级用户能自己快速熟悉FreeBSD 的基础内容。这只是份技术上的快速入门，并非是试图描绘这两种操作系统之间的"哲学"上的差异。

此文档假定认为你已经安装好了FreeBSD。如果你还没有安装FreeBSD 或者对FreeBSD 的安装过程方面需要帮助，请参考FreeBSD 手册的 安装FreeBSD (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/install.html)一章。

2 Shell程序：没有Bash吗?

那些从Linux 转过来的用户经常会惊讶于**Bash** 不是FreeBSD 的默认Shell。事实上，**Bash** 甚至没有包括在FreeBSD 的默认安装中。代替的是，FreeBSD 使用tcsh(1) 作为自己的默认Shell，尽管如此，**Bash** 和其他你喜爱的Shell 程序在FreeBSD 的Packages 和Ports 套件 (article.html#SOFTWARE) 里都可以找到。

如果你安装了其他的Shell 你可以使用chsh(1) 来设置一个用户的默认Shell。通常情况下，强烈建议不要去更改`root` 用户的默认Shell。原因是这些Shell 没有包括在基本系统中，正常情况下它们会被安装到`/usr/local/bin` 和`/usr/bin` 目录下。万一某天`/usr/local/bin` 和`/usr/bin` 的文件系统不能被挂载，这样情况下`root` 将不能进入自己默认的Shell，从而`root` 将不能够登录进去。鉴于这个原因，第二个系统管理员帐户`toor` 创建时使用的是非默认的Shell。在安全FAQ 可以查阅到关于toor 帐户 (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/faq/security.html#TOOR-ACCOUNT) 的信息。

3 Packages和Ports：在FreeBSD 中添加软件

除了经典的UNIX® 安装软件的方法（下载源码包，解压，编辑源码，编译）外，FreeBSD 还提供了另外两种方法来安装应用程序：packages 和ports。你可以在这里 (http://www.freebsd.org/ports/master-index.html) 到一份完整可用的ports 和packages 的软件清单。

3.1 Packages

Packages 是预编译好的应用程序，在FreeBSD 中等价于基于Debian/Ubuntu 的系统中的`.deb` 软件包以及基于Red Hat/Fedora 的系统中的`.rpm` 软件包。Packages使用pkg_add(1) 来进行安装。例如，下面的命令将用来安装**Apache 2.2**:

```
# pkg_add /tmp/apache-2.2.6_2.tbz
```

使用`-r` 操作将告诉pkg_add(1) 来自动获取并安装一个软件包，以及解决所有的依赖关系:

```
# pkg_add -r apache22
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-release/Latest/apache22.tbz... Do
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-release/All/expat-2.0.0_1.tbz...
Fetching ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6.2-release/All/perl-5.8.8_1.tbz... D
[snip]

To run apache www server from startup, add apache22_enable="YES"
in your /etc/rc.conf. Extra options can be found in startup script.
```

注意: 如果你正运行着release 版本的FreeBSD （6.2，6.3，7.0等，通常从CD-ROM 被安装的）`pkg_add -r` 会为其下载专门为这些特定版本构建好的软件包。这些软件包**可能** 不是当前最新的程序。你可以使用`PACKAGESITE` 变量来覆盖默认的动作。例如，把`PACHAGESITE` 设置成ftp://ftp.freebsd.org/pub/FreeBSD/ports/i386/packages-6-stable/Latest/ 来下载6.X 系列最新的包。

想了解更多的packages 信息请查阅FreeBSD 手册的4.4 小节：使用Packages 系统 (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/packages-using.html)。

3.2 Ports

FreeBSD 的第二种安装应用程序的方法就是使用Ports 套件了。Ports 套件是FreeBSD 上的一个利用`Makefile` 和一些补丁文件来特定从源码定制安装各种软件程序的框架。当安装一个port 时系统会获取程序源码，应用任何所需要的补丁，编译源码，并安装应用程序（并针对依赖关系以同样的方式安装解决）。

Ports 套件，常被称作ports 树，可以在`/usr/ports` 下找到。假设Ports套件已经在安装FreeBSD 时安装过了。如果Ports 套件还没有被安装可以通过sysinstall(8) 来进行安装，或者使用csup(1) 或portsnap(8) 来从FreeBSD 的服务器上面拉下来。在手册的4.5.1 小节 (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/ports-using.html) 可以找到安装Ports 套件的详细介绍。

安装一个port 就像进入port 的目录并开始构建过程一样简单（通常情况下），下面是从Ports 套件安装**Apache 2.2** 的例子：

```
# cd /usr/ports/www/apache22
# make install clean
```

使用ports 安装软件的最大好处就是能够自定义安装选项。例如，从ports 安装**Apache 2.2** 时你可以通过设置`WITH_LDAP` make(1) 变量来启用**mod_ldap**：

```
# cd /usr/ports/www/apache22
# make WITH_LDAP="YES" install clean
```

请查看FreeBSD 手册的4.5 小节， 使用Ports Collection (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/ports-using.html)，以获取更多关于Ports Collection 的信息。

3.3 Ports还是packages，我应该使用哪个?

Packages 就是预编译好的ports，所以从源码（ports）安装与从二进制packages 安装这两者间确实有很大关联。每种方法各有自己的优点：

Packages（二进制）

- 更快速的安装（比较大的应用程序编译起来会花很长时间）。
- 你不需要知道如何编译软件。
- 不需要在操作系统上安装编译器。

Ports（源码）

- 能够定制安装选项。（Packages通常都是使用标准选项构建的。使用ports 你能够定义各种各样的选项，比如类似构建附加的模块或是更改安装路径之类的。）
- 如果你喜欢的话还可以使用自己的补丁。

如果你没有一些特别的需求，packages 可能刚好最适合你的情况。如果你需要进一步定制，ports 是最适合的方法了。（请记得，如果你需要定制而自己又更倾向于使用packages，你可以使用`make package`从ports 构建一个定制的package，然后复制到其他的服务器。）

4 系统启动：运行级别在哪里?

Linux 使用Sysv init 初始化系统，而FreeBSD 使用的是传统的BSD 风格的init(8)。在BSD 风格的init(8)中没有运行级别和`/etc/inittab`，代替控制启动的是rc(8) 实用程序。`/etc/rc` 脚本读取`/etc/defaults/rc.conf` 和`/etc/rc.conf` 文件来决定哪个服务将被启动。特殊服务在此后由处于`/etc/rc.d/` 和`/usr/local/etc/rc.d/` 下的相应服务初始化脚本文件所启动。这些脚本类似于位于Linux 系统中的`/etc/init.d/` 目录下的脚本。

为何会有两个服务初始化脚本的目录呢? `/etc/rc.d/` 下的脚本是属于“基本” 系统一部分的应用程序所使用的。（cron(8)，sshd(8)，syslog(3)，以及其他。）`/usr/local/etc/rc.d/` 下的脚本是用户安装的应用程序如**Apache**，**Squid** 等使用的。

“基本” 系统和用户安装的应用程序之间的区别是什么? FreeBSD 是一套开发出来的完整的操作系统，也就是说，内核，系统类库，还有实用应用程序（如ls(1)，cat(1)，cp(1) 等）全部被做为一个整体一起开发并释出。这就是被认为归属于“基本”系统的程序。用户安装的程序并不是“基本”系统的一部分，如**Apache**，**X11**，**Moazilla Firefox**，等等。这些用户安装的应用程序通常是使用FreeBSD 的Packages 和Ports 套件安装上去的。为了将这些程序和“基本” 系统区分开来，用户安装的应用程序通常被安装到`/usr/local/`下。因此用户安装的二进制执行文件存在于`/usr/local/bin`下，配置文件在`/usr/local/etc`下，以此类推。

您可以通过在`/etc/rc.conf`（rc.conf(5)）文件中增加与之对应的*`ServiceName`*`_enable="YES"` 配置来启用服务。看一下系统默认的`/etc/defaults/rc.conf` 文件，这些默认配置可以使用`/etc/rc.conf` 文件来改变。因此，当安装附加应用程序时最好回顾下文档来决定到底该如何启用任何相关的服务。

下面的一小段内容用来在`/etc/rc.conf` 中启用sshd(8) 和**Apache 2.2**。还指定了**Apache** 应该通过SSL 方式启动。

```
# enable SSHD
sshd_enable="YES"
# enable Apache with SSL
apache22_enable="YES"
apache22_flags="-DSSL"
```

一旦服务已经在`/etc/rc.conf` 中启用，服务将能够从命令行启动（不需要重新启动系统）：

```
# /etc/rc.d/sshd start
```

如果服务还没有被启用，可以使用`forcestart` 来从命令行启动：

```
# /etc/rc.d/sshd forcestart
```

5 网络配置

5.1 网络接口

代替Linux 中所使用的标识网络接口所常用的**ethX** 格式的是，FreeBSD 使用驱动名字后跟一个数字来标识。下面ifconfig(8) 的输出显示了两个Intel® Pro 1000 的网络接口（`em0` 和`em1`）：

```
% ifconfig
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet 10.10.10.100 netmask 0xffffff00 broadcast 10.10.10.255
        ether 00:50:56:a7:70:b2
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
em1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=b<RXCSUM,TXCSUM,VLAN_MTU>
        inet 192.168.10.222 netmask 0xffffff00 broadcast 192.168.10.255
        ether 00:50:56:a7:03:2b
        media: Ethernet autoselect (1000baseTX <full-duplex>)
        status: active
```

5.2 IP配置

一个IP 地址可以使用ifconfig(8) 来指定到一个网络接口。通常，要保持重启后依然能够使用的IP 配置信息需要包含在`/etc/rc.conf` 中。下列例子指定了主机名，IP 地址，以及默认网关：

```
hostname="server1.example.com"
ifconfig_em0="inet 10.10.10.100  netmask 255.255.255.0"
defaultrouter="10.10.10.1"
```

使用下面内容来为网络接口配置DHCP：

```
hostname="server1.example.com"
ifconfig_em0="DHCP"
```

6 防火墙

像Linux 中的**IPTABLES** 一样，FreeBSD 也提供了一个内核级的防火墙；实际上FreeBSD 提供了三个防火墙：

- IPFIREWALL (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipfw.html)
- IPFILTER (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/firewalls-ipf.html)
- PF (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/firewalls-pf.html)

IPFIREWALL 或是**IPFW**（管理**IPFW** 规则的ipfw(8) 命令）是FreeBSD 开发者开发并维持的。**IPFW** 能够与dummynet(4) 配合使用来提供流量图形功能以及模拟不同网络连接类型的功能。

允许**SSH** 进入的**IPFW** 规则样例如下：

```
ipfw add allow tcp from any to me 22 in via $ext_if
```

IPFILTER 是Darren Reed 所开发的防火墙程序。不是专门针对FreeBSD 的，它已经被移植到NetBSD，OpenBSD，SunOS，HP/UX，还有Solaris等一些操作系统之上。

允许**SSH** 进入的**IPFILTER** 命令样例如下：

```
pass in on $ext_if proto tcp from any to any port = 22
```

最后一种防火墙程序，**PF**，是OpenBSD 项目所开发的。**PF** 是被作为**IPFILTER** 的一个替代品而被创建出的。就这点而言，**PF** 的语法与**IPFILTER** 的非常相似。**PF** 可以与altq(4) 配合来提供QoS 的特性。

允许**SSH** 进入的**PF** 命令样例如下：

```
pass in on $ext_if inet proto tcp from any to ($ext_if) port 22
```

7 升级FreeBSD

共有三种方法来升级FreeBSD 系统：源码，二进制更新，还有安装光盘。

从源码升级是最复杂的升级方法，但是提供了最棒的总体灵活性。这个过程包含了使用FreeBSD **CVS**（并行版本系统）来同步一个本地的FreeBSD 源代码。一旦本地源码已经更新到当前最新你便可以构建新版本的内核以及应用程序。关于源码更新的更多信息可见于FreeBSD 手册 关于如何更新操作系统的章节 (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/cutting-edge.html)。

二进制更新类似于使用`yum` 或`apt-get` 更新Linux 系统。freebsd-update(8) 命令会获取新的更新并安装它们。这些更新可以通过cron(8) 使用程序来调度。

> **注意:** 如果你使用cron(8) 来预定更新，请确信在你的crontab(1) 中使用了`freebsd-update cron` 来控制大数目的机器同时获取更新。
>
> ```
> 0 3 * * * root /usr/sbin/freebsd-update cron
> ```

最后一种更新的方法，从安装光盘来升级，是个直接的过程。从安装光盘启动并选择该选项来更新。

8 procfs：已是过去式但仍未被遗忘

Linux 中，你可能会通过看一看`/proc/sys/net/ipv4/ip_forward` 来确定IP 转发是否被启用。在FreeBSD 中你应该使用sysctl(8) 来查看这和其他方面的系统设置，在当前的FreeBSD 版本中procfs(5) 已经不赞成使用了。（虽然`sysctl`在FreeBSD 也同样可用。）

在IP 转发样例中，你应该使用下列内容来确定FreeBSD 系统中是否已经开启了IP 转发：

```
% sysctl net.inet.ip.forwarding
net.inet.ip.forwarding: 0
```

`-a` 标志用来列出所有的系统设置：

```
% sysctl -a
kern.ostype: FreeBSD
kern.osrelease: 6.2-RELEASE-p9
kern.osrevision: 199506
kern.version: FreeBSD 6.2-RELEASE-p9 #0: Thu Nov 29 04:07:33 UTC 2007
    root@i386-builder.daemonology.net:/usr/obj/usr/src/sys/GENERIC

kern.maxvnodes: 17517
kern.maxproc: 1988
kern.maxfiles: 3976
kern.argmax: 262144
kern.securelevel: -1
kern.hostname: server1
kern.hostid: 0
kern.clockrate: { hz = 1000, tick = 1000, profhz = 666, stathz = 133 }
kern.posix1version: 200112
...
```

注意: 某些`sysctl` 的参数是只读的。

需要procfs 的情况是，运行一些较老的软件，使用truss(1) 来跟踪系统信号，以及Linux 二进制兼容(http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/linuxemu.html).（尽管，Linux 二进制兼容性使用其本身的procfs，linprocfs(5)。）如果你需要挂载procfs 你可以在`/etc/fstab` 中加入如下内容:

```
proc                /proc           procfs  rw,noauto       0       0
```

注意: `noauto` 会防止`/proc` 在启动时被自动挂载。

然后使用如下命令挂载procfs:

```
# mount /proc
```

9 常用命令

9.1 软件包管理

Linux 命令(Red Hat/Debian)	FreeBSD 等价命令	目的
`yum install` *package* / `apt-get install` *package*	`pkg_add -r` *package*	从远程仓库安装*package*
`rpm -ivh` *package* / `dpkg -i` *package*	`pkg_add -v` *package*	安装package
`rpm -qa` / `dpkg -l`	`pkg_info`	列出已安装的软件包

9.2 系统管理

Linux 命令	FreeBSD 等价命令	目的
`lspci`	`pciconf`	列出PCI 设备
`lsmod`	`kldstat`	列出已载入的内核模块
`modprobe`	`kldload` / `kldunload`	载入/卸载内核模块
`strace`	`truss`	跟踪系统调用

10 总结

非常希望这篇文档能够给予你足够的帮助来开始你的FreeBSD 之路。务必要再去看一下FreeBSD 手册 (http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/index.html) 所提到的但并没有被包含在本文档中的那些更深入广泛的主题。