

# Пособие для новичков во FreeBSD и UNIX®

Annelise Anderson <[andrsn@andrsn.stanford.edu](mailto:andrsn@andrsn.stanford.edu)>

Издание: 46680

FreeBSD это зарегистрированная торговая марка FreeBSD Foundation.

IBM, AIX, OS/2, PowerPC, PS/2, S/390 и ThinkPad это торговые марки International Business Machines Corporation в Соединенных Штатах, других странах, или по всему миру.

Microsoft, FrontPage, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media и Windows NT это или зарегистрированные торговые марки или торговые марки Microsoft Corporation в Соединенных Штатах и/или других странах.

Motif, OSF/1 и UNIX это зарегистрированные торговые марки, а IT DialTone и The Open Group это торговые марки Open Group в Соединенных Штатах и других странах.

Многие из обозначений, используемые производителями и продавцами для обозначения своих продуктов, заявляются в качестве торговых марок. Когда такие обозначения появляются в этом документе, и Проекту FreeBSD известно о торговой марке, к обозначению добавляется знак «™» или «®».

15 августа 1997 .

## Аннотация

Поздравляем вас с установкой FreeBSD! Это вводное пособие предназначено для тех, кто является новичком в мире FreeBSD и UNIX®—так что оно начнётся с основ.

## Содержание

1. Регистрация в системе и выход из неё .....	1
2. Добавление пользователя с привилегиями root .....	2
3. Просмотр окружения .....	3
4. Получение помощи и информации .....	3
5. Редактирование текста .....	4
6. Другие полезные команды .....	6
7. Следующие шаги .....	7
8. Ваше рабочее окружение .....	7
9. Остальное .....	8
10. Пожелания приветствуются .....	9

## 1. Регистрация в системе и выход из неё

Зарегистрируйтесь в системе (когда увидите приглашение `login:` ) как пользователь, которого вы создали во время установки, или войдите в систему как пользователь `root`. (В вашей установленной системе уже имеется учётная запись для пользователя `root`; который может переходить хоть куда и делать всё, что угодно, в том числе удаление необходимых для работы файлов, так что будьте внимательны!) Обозначения `%` и `#` в последующем тексте означают приглашения системы (ваше может отличаться от него), причём `%` обозначает обычного пользователя, а `#` пользователя `root`.

Чтобы выйти из системы (и получить новое приглашение login: ) наберите

```
# exit
```

столько раз, сколько нужно. Да, нажимайте enter после набора команд, и помните, что UNIX® чувствителен к регистру букв—набирайте exit, но не EXIT.

Для завершения работы машины наберите

```
# /sbin/shutdown -h now
```

Или, для перезагрузки нужно набрать

```
# /sbin/shutdown -r now
```

или

```
# /sbin/reboot
```

Перезагрузку можно также выполнить нажатием клавиш Ctrl+Alt+Delete. Подождите некоторое время, чтобы дать этой команде отработать. В последних релизах FreeBSD она эквивалента выдаче команды /sbin/reboot и гораздо, гораздо лучше, чем нажатие кнопки сброса. Вы ведь не хотите всё переустанавливать заново, не так ли?

## 2. Добавление пользователя с привилегиями root

Если при установке системы вы не создали ни одного пользователя, и поэтому вошли в систему как root, то теперь вы должны создать пользователя по команде

```
# adduser
```

При первом использовании утилиты adduser она может запрашивать сохранение некоторых параметров для использования их по умолчанию. вы можете сделать оболочкой, используемой по умолчанию, командный процессор csh(1), а не sh(1), если по умолчанию вам предлагается sh. В противном случае просто нажимайте enter для принятия всех предлагаемых по умолчанию вариантов. Эти значения по умолчанию сохраняются в файле /etc/adduser.conf, в форме, доступной для редактирования.

Предположим, что вы создали пользователя jack с полным именем Jack Benimble. Назначьте пользователю jack пароль, если информационная безопасность имеет значение (даже если это дети, которые могут стучать по клавиатуре). Когда вам будет задан вопрос по включению пользователя jack в другие группы, наберите wheel

```
Login group is `jack`. Invite jack into other groups: wheel
```

Это позволит входить в систему как пользователь jack и использовать команду su(1) для того, чтобы стать пользователем root. Тогда вас не будут больше обвинять в том, чтобы вы входите в систему как пользователь root.

Вы можете прекратить работы с adduser в любой момент, нажав Ctrl+C, а в завершении ввода у вас будет шанс подтвердить заведение нового пользователя или набрать n в качестве отрицательного ответа. Вам может захотеться создать второго нового пользователя, для того, чтобы при редактировании файлов для входа пользователя jack имелся горячий резерв на тот случай, если что-то пойдет не так.

После того, как вы это сделаете, воспользуйтесь командой exit для возврата к приглашению ко входу в систему и зарегистрируйтесь в ней как пользователь jack. Вообще говоря, лучше всего основную массу работы выполнять, работая как обычный пользователь, который не имеет мощь и опасность пользователя root.

Если вы уже создали пользователя и хотите, чтобы он мог выполнять команду su для получения привилегий root, вы можете войти в систему как root и отредактировать файл /etc/group, добавив пользователя jack в первую строчку (в группу wheel). Однако сначала вам нужно поупражняться с программой vi(1),

текстовым редактором,—или использовать более простой редактор, [ee\(1\)](#), имеющийся в последней версии FreeBSD.

Для удаления пользователя воспользуйтесь командой `rmuser`.

### 3. Просмотр окружения

Войдя в систему как обычный пользователь, оглянитесь вокруг и попробуйте выполнить некоторые команды, дающие доступ к источникам информации и помощи внутри FreeBSD.

Вот некоторые команды и то, что они делают:

`id`

Говорит вам, кто вы!

`pwd`

Показывает, где вы находитесь—текущий рабочий каталог.

`ls`

Выдаёт список файлов, находящихся в текущем каталоге.

`ls -F`

Выдаёт перечень файлов, находящихся в текущем каталоге, добавляя символы `*` после выполнимых файлов, `/` после каталогов и `@` после символических ссылок.

`ls -l`

Выдаёт перечень файлов в расширенном формате—размер, дата и права доступа.

`ls -a`

Вместе со всеми выдаёт и список скрытых «dot»-файлов (начинающихся с точки). Если вы являетесь пользователем `root`, то «dot»-файлы выдаются и без указания флага `-a`.

`cd`

Смена каталогов. `cd ..` перемещает на один уровень выше; обратите внимание на промежуток после `cd`. `cd /usr/local` перейдёт в указанное место. `cd ~` перейдёт в домашний каталог человека, который вошёл в систему—к примеру, `/usr/home/jack`. Попробуйте выполнить команду `cd /cdrom`, а затем `ls` для проверки того, что ваш CDROM смонтирован и работает.

`less filename`

Позволяет вам просмотреть файл (с именем *filename*) без внесения в него изменений. Попробуйте выполнить команду `less /etc/fstab`. Для выхода наберите `q`.

`cat filename`

Выдаёт содержимое *filename* на экран. Если он слишком длинный и вы можете увидеть только его конец, нажмите ScrollLock и используйте клавишу стрелка вверх для движения назад; вы можете также использовать ScrollLock и со страницами справки. Нажмите ScrollLock снова для прекращения прокрутки. Вам может захотеться попробовать команду `cat` с некоторыми из dot-файлов в вашем домашнем каталоге—`cat .cshrc`, `cat .login`, `cat .profile`.

В файле `.cshrc` вы заметите алиасы для некоторых из команд `ls` (они очень удобны). Вы можете создать другие алиасы, отредактировав файл `.cshrc`. Вы можете сделать эти алиасы доступными всем пользователям системы, поместив их в общесистемный конфигурационный файл для `cs`, `/etc/csh.cshrc`.

### 4. Получение помощи и информации

Вот несколько полезных источников получения помощи. Здесь *Text* обозначает что-то по вашему выбору, что вы вводите—обычно команду или имя файла.

**apropos text**

Всё, что содержит строку *text* в базе *whatis*.

**man text**

Страница справки по *text*. Это главный источник документации в UNIX®-системах. *man ls* покажет вам все способы использования команды *ls*. Нажимайте Enter для передвижения по тексту, Ctrl+B для возврата на страницу назад, Ctrl+F для продвижения вперёд, q или Ctrl+C для выхода.

**which text**

Покажет, в каком месте из маршрута поиска пользователя находится команда *text*.

**locate text**

Все маршруты, где находится строчка *text*.

**whatis text**

Описывает, что делает команда *text* и её справочная страница. Команда *whatis \** расскажет вам обо всех двоичных файлах в текущем каталоге.

**whereis text**

Ищет файл *text* и выдаёт полный путь до него.

Вы можете захотеть попробовать использовать команду *whatis* с некоторыми полезными командами типа *cat*, *more*, *grep*, *mv*, *find*, *tar*, *chmod*, *chown*, *date*, и *script*. Команда *more* позволит вам читать постранично, как и в DOS, например, *ls -l | more* или *more filename*. Знак *\** работает как общий шаблон—например, *ls w\** выдаст перечень файлов, начинающихся с буквы *w*.

Некоторые из этих команд работают не очень хорошо? Обе команды [locate\(1\)](#) и [whatis\(1\)](#) зависят от базы данных, которая перестраивается еженедельно. Если ваша машина будет оставаться включенной на выходные (и она работает под FreeBSD), то вы можете пожелать запускать определённые команды раз в день, неделю, месяц. Запускайте их как *root* и дайте каждой отработать, прежде чем запускать следующую.

```
# periodic daily
выдача опущена
# periodic weekly
выдача опущена
# periodic monthly
выдача опущена
```

Если вам надоело ждать, нажмите Alt+F2 для перехода в другую виртуальную консоль, и войдите в систему снова. В конце концов, это многопользовательская и многозадачная система. Тем не менее эти команды, скорее всего, в процессе работы будут выдавать сообщения вам на экран; вы можете набрать *clear* в приглашении для очистки экрана. Пока они работают, вы можете смотреть в содержимое файлов */var/mail/root* и */var/log/messages*.

Выполнение таких команд является частью системного администрирования—и как единственный пользователь UNIX®-системы вы являетесь собственным системным администратором. Практически всё, для чего вам нужно быть пользователем *root*, это системное администрирование. Эти обязанности не описываются достаточно хорошо даже в тех больших толстых книгах по UNIX®, в которых слишком много места отдаётся описанию работы с меню в оконных менеджерах. Вам может понадобиться одна из двух лучших книг по системному администрированию, либо автора Эви Немет *UNIX System Administration Handbook* (Prentice-Hall, 1995, ISBN 0-13-15051-7)—второе издание с красной обложкой; или автора Аileen Frisch *Essential System Administration* (O'Reilly & Associates, 2002, ISBN 0-596-00343-9). Я использую книгу Немет.

## 5. Редактирование текста

Для конфигурации вашей системы вам нужно редактировать текстовые файлы. Большинство из них будут находиться в каталоге */etc*; и вам необходимо командой *su* получить полномочия пользователя *root*, чтобы их править. Вы можете использовать простой редактор *ee*, однако в смысле перспективности лучше

изучить текстовый редактор `vi`. В каталоге `/usr/src/contrib/nvi/docs/tutorial` есть прекрасный учебник по `vi`, если у вас есть исходники системы.

Перед тем, как редактировать файл, наверное, вы должны сохранить резервную копию. Предположим, что вы собираетесь отредактировать файл `/etc/rc.conf`. Вы можете воспользоваться командой `cd /etc` для перехода в каталог `/etc` и выполнить следующее:

```
# cp rc.conf rc.conf.orig
```

При этом файл `rc.conf` скопируется в `rc.conf.orig`, и в последующем вы сможете скопировать `rc.conf.orig` в файл `rc.conf` для восстановления оригинала. Но ещё лучше его переместить (переименовать), после чего скопировать обратно:

```
# mv rc.conf rc.conf.orig
# cp rc.conf.orig rc.conf
```

потому что команда `mv` сохраняет исходную информацию о дате и владельце файла. Теперь вы можете редактировать `rc.conf`. Если вы захотите восстановить исходное состояние, то выполните `mv rc.conf rc.conf.myedit` (полагаем, что вы хотите сохранить отредактированную версию), а затем

```
# mv rc.conf.orig rc.conf
```

для возврата всего на место.

Для редактирования файла наберите

```
# vi filename
```

Передвигайтесь по тексту при помощи клавиш со стрелками. `Esc` (клавиша отмены) переводит редактор `vi` в командный режим. Вот некоторые из них:

`x`

удалить символ, на котором находится курсор

`dd`

удалить целую строку (даже если на экране она не помещается в целую строку)

`i`

вставка текста в позиции курсора

`a`

вставка текста после курсора

Сразу после набора `i` или `a` вы можете вводить текст. `Esc` возвратит вас обратно в командный режим, где вы можете набрать

`:w`

для записи ваших изменений на диск и продолжения редактирования

`:wq`

для записи и выхода

`:q!`

для выхода без сохранения изменений

`/text`

для перемещения курсора на `text`; `/Enter` (клавиша ввода) для поиска следующего экземпляра `text`.

`G`

для перехода в конец файла

**nG**Для перехода к строке *n* в файле, где *n* является числом**Ctrl+L**

для перерисовки экрана

**Ctrl+b** и **Ctrl+f**для перемотки на экран назад и вперёд, как при работе с *more* и *view*.

Поупражняйтесь с редактором *vi* в своём домашнем каталоге, создав новый файл по команде *vi filename*, добавляя и удаляя текст, сохраняя файл и вызывая его снова. Редактор *vi* преподносит некоторые сюрпризы, потому что он на самом деле достаточно сложный, и иногда вы можете неправильно вызвать команду, которая сделает нечто, чего вы не ожидали. (Некоторым людям действительно нравится *vi*—он более мощный, чем *EDIT* из *DOS*—посмотрите команду *:r*.) Для того, чтобы удостовериться, что вы находитесь в режиме команд, нажимайте *Esc* один или несколько раз, и начинайте снова с этого места, если возникли какие-то проблемы, часто сохраняйте текст командой *:w* и используйте *:q!* для того, чтобы прекратить работу и начать всё сначала (с вашей последней команды *:w*), если это нужно.

Теперь вы можете выполнить *cd* для перехода в каталог */etc*, *su* в пользователя *root*, использовать *vi* для редактирования файла */etc/group* и добавлять пользователя в группу *wheel*, чтобы он имел полномочия пользователя *root*. Просто добавьте запятую и имя входа пользователя в конце первой строки этого файла, нажмите *Esc* и воспользуйтесь *:wq* для записи файла на диск и выхода. Работает всегда. (Вы не поставили пробел после запятой, ведь так?)

## 6. Другие полезные команды

**df**

выдаёт данные о занятом файлами пространстве и смонтированных файловых системах.

**ps aux**показывает работающие процессы. *ps ax* является частоупотребительной формой.**rm filename**удаляет *filename*.**rm -R dir**удаляет каталог *dir* и все его подкаталоги—осторожно!**ls -R**выдаёт список файлов в текущем каталоге и всех его подкаталогах; я использовал вариант, *ls -AFR > where.txt*, для получения перечня всех файлов в */* и (отдельно) */usr* до того, как узнал о более эффективном способе поиска файлов.**passwd**для изменения пароля пользователя (или пароля *root*)**man hier**

справочная страница по файловой структуре UNIX®

Используйте *find* для поиска *filename* в */usr* или в любом из её подкаталогов при помощи команды

```
% find /usr -name " filename "
```

Вы можете использовать *\** в качестве шаблона внутри *"filename"* (это выражение должно быть в кавычках). Если вы укажете команде *find* на поиск в */*, а не в */usr*, то она будет искать файл(ы) во всех смонтированных файловых системах, включая *CDROM* и раздел *DOS*.

Прекрасным пособием, описывающим команды и утилиты UNIX®, является книга Abrahams & Larson, *Unix for the Impatient* (2nd ed., Addison-Wesley, 1996). Масса информации по UNIX® есть и в Internet.

## 7. Следующие шаги

Теперь вы должны иметь инструменты, которые необходимо держать под рукой и умеете редактировать файлы, так что вы должны суметь запустить всё, что угодно. Много полезной информации содержится в Руководстве по FreeBSD (которое, скорее всего, есть на вашем жёстком диске) и [Web-сайте FreeBSD](#). На CDROM, а также Web-сайте находятся различные пакеты и порты. В Руководстве рассказывается более подробно о том, как их использовать (получить пакет, если он существует, командой `pkg_add /cdrom/packages/All/packagename`, где *packagename* является именем файла пакета). На CDROM находится перечни пакетов и портов с их краткими описаниями в файлах `cdrom/packages/index`, `cdrom/packages/index.txt` и `cdrom/ports/index`, а более полные описания можно найти в `/cdrom/ports/*/*/pkg/DESCR`, где знаки \* обозначают тематические подкаталоги с программами и названиями программ, соответственно.

Если вы считаете, что Руководство является слишком сложной книгой (что с `lndir` и всё) по установке портов с CDROM, вот рецепт, который обычно срабатывает:

Найдите нужный вам порт, скажем, `kermi`t. На CDROM для него должен существовать каталог. Скопируйте этот подкаталог в каталог `/usr/local` (хорошее место для программного обеспечения, которое вы добавляете, и которое должно быть доступно всем пользователям) такой командой:

```
# cp -R /cdrom/ports/comm/kermi /usr/local
```

В результате должен образоваться подкаталог `/usr/local/kermi`t, содержащий все файлы, что есть в подкаталоге `kermi`t на CDROM.

Затем создайте каталог `/usr/ports/distfiles`, если он ещё не существует, при помощи команды `mkdir`. Теперь проверьте содержимое `/cdrom/ports/distfiles` на предмет наличия файла с именем, говорящем о том, что это тот порт, который вы хотите иметь. Скопируйте этот файл в каталог `/usr/ports/distfiles`; в последних версиях вы можете пропустить этот шаг, и FreeBSD выполнит его за вас. В случае с `kermi`t, дистрибутивного файла не существует.

После этого по команде `cd` перейдите в подкаталог `/usr/local/kermi`t, в котором есть файл `Makefile`. Наберите

```
# make all install
```

Во время выполнения порт обратится к FTP для получения всех архивных файлов, нужных ему и которых не найдено на CDROM или в каталоге `/usr/ports/distfiles`. Если сеть у вас ещё не работает, и файла для порта в каталоге `/cdrom/ports/distfiles` нет, вам потребуется получить дистрибутивный файл на другой машине и скопировать его в каталог `/usr/ports/distfiles`. Прочтите `Makefile` (при помощи команд `cat`, `more` или `view`), чтобы понять, как называется файл и куда нужно обратиться (основной сайт распространения), чтобы его получить. (Используйте двоичный тип передачи файлов!) Затем перейдите обратно в каталог `/usr/local/kermi`t, найдите каталог с `Makefile` и наберите `make all install`.

## 8. Ваше рабочее окружение

Ваш командный процессор является самой важной частью вашего рабочего окружения. Оболочка занимается интерпретацией команд, которые вы вводите в командной строке, и таким образом взаимодействует с остальной частью операционной системы. Вы можете также писать скрипты командного процессора, то есть последовательности команд, которые должны выполняться без вашего участия.

Вместе с FreeBSD устанавливаются два командных процессора: `csh` и `sh`. `csh` хорош для работы в командной строке, однако скрипты должны писаться на языке оболочек `sh` (или `bash`). Вы можете выяснить, какой командный процессор у вас используется, набрав `echo $SHELL`.

Оболочка `csh` подходящая, однако `tcsh` может всё, что умеет `csh` и ещё больше. Она позволяет вам восстанавливать прошлые команды клавишами со стрелками и редактировать их. В нём есть автозавершение имён файлов по нажатию клавиши табуляции (в `csh` используется клавиша `Esc`) и он позволяет вам пере-

ключаться в каталог, в котором вы были ранее, по команде `cd -`. Также в `tcsh` гораздо легче изменять системное приглашение. Это гораздо упрощает жизнь.

Вот три шага по установке нового командного процессора:

1. Установите командный процессор как порт или пакет, как вы обычно это делаете с другим портом или пакетом.
2. Работая как пользователь `root`, отредактируйте файл `/etc/shells`, добавив в него строку с новой оболочкой, в нашем случае это `/usr/local/bin/tcsh`, и сохраните файл. (Некоторые порты могут делать это за вас.)
3. Воспользуйтесь командой `chsh` для смены постоянно используемой вами оболочки на `tcsh`, либо наберите `tcsh` в командной строке для смены вашей оболочки без повторного входа в систему.



### Примечание

Менять командный процессор для пользователя `root` на что-то, отличающееся от `sh` или `csh`, в ранних версиях FreeBSD и во многих других версиях UNIX® может быть опасно; вы можете лишиться работающей оболочки при переходе системы в однопользовательский режим. Решением является использование `su -m` для того, чтобы стать пользователем `root`, что даст в качестве оболочки `tcsh`, но вы будете являться пользователем `root`, потому что оболочка является частью окружения. Вы можете сделать это постоянным, добавив в ваш файл `.tcshrc` в качестве алиаса по такой команде:

```
alias su su -m
```

При запуске `tcsh` он будет считывать файлы `/etc/csh.cshrc` и `/etc/csh.login`, как и `csh`. Эта оболочка также читает файл `.login` из вашего домашнего каталога, а также файл `.cshrc`, если только вы не создали файл `.tcshrc`. Это вы можете сделать простым копированием файла `.cshrc` в `.tcshrc`.

Теперь, когда у вас установлен командный процессор `tcsh`, вы можете настроить приглашение командной строки. Все подробности можно найти на странице справки по `tcsh`, но всё же вот строка, которая помещается в ваш файл `.tcshrc`, которая может показать, сколько команд вы уже набрали, сколько сейчас времени и в каком каталоге вы находитесь. Она также выдаёт `>`, если вы являетесь обычным пользователем, и `#`, если вы являетесь пользователем `root`, однако `tsch` будет делать это в любом случае:

```
set prompt = "%h %t %~ %# "
```

Эта строка должна быть поставлена на то же самое место, что и существующая строка установки приглашения, если она есть, либо после строки `"if($?prompt) then"`, если её нет. Закомментируйте старую строку; вы всегда сможете вернуться к ней обратно, если предпочтёте её. Не забудьте о пробелах и кавычках. Вы можете заставить перечитать `.tcshrc`, набрав `source .tcshrc`.

Перечень других установленных переменных окружения вы можете получить, набрав `env` в приглашении командной строки. В результате, кроме всего прочего, будут показаны редактор, используемый по умолчанию, программа постраничной выдачи и тип терминала. Командой, полезной при входе в систему с удалённого места и невозможности запуска программы, потому что терминал не обладает некоторыми возможностями, является команда `setenv TERM vt100`.

## 9. Остальное

Работая как пользователь `root`, вы можете отмонтировать CDROM по команде `/sbin/umount /cdrom`, вытащить его из привода, вставить другой диск и смонтировать его командой `/sbin/mount_cd9660 /dev/`



`cd0a /cdrom`, при этом предполагается, что `cd0a` является именем устройства для вашего привода CDROM. Самые последние версии FreeBSD позволяют вам монтировать CDROM просто по команде `/sbin/mount /cdrom`.

Использование живой файловой системы—она находится на втором диске FreeBSD из набора CDROM—полезно при нехватке пространства. То, что находится в этой файловой системе, меняется от релиза к релизу. Вы можете попытаться поиграть в игры с CDROM. При этом применяется команда `lndir`, которая устанавливается с X Window System, и служит для указания программам, где искать необходимые файлы, потому что они находятся в файловой системе `/cdrom`, а не в `/usr` и её подкаталогах, где должны находиться. Прочтите справку по команде `man lndir`.

## 10. Пожелания приветствуются

Если вы используете это руководство, мне будет интересно знать, в каком месте оно написано непонятно и что упущено из того, что, по вашему мнению, должно быть включено ценного. Мои благодарности Eugene W. Stark, профессору информатики в SUNY-Stony Brook, и John Fieber за ценные советы.

Annelise Anderson, <[andrsn@andrsn.stanford.edu](mailto:andrsn@andrsn.stanford.edu)>

