



Общие вопросы

[printable page](#)

8.1 - Я забыл пароль root, что теперь делать?

процесс восстановления пароля включает следующие шаги - загрузку в однопользовательском режиме, монтирование необходимых разделов (/ и /usr), выполнение команды [passwd\(1\)](#) для смены пароля root. После этого можно загрузиться обычным образом и залогиниться.

Подробнее о процессе:

Загрузка в однопользовательском режиме. Эта часть процесса различается для разных [платформ](#). Для amd64 и i386, на втором этапе загрузки, при [boot\(8\)](#), есть пауза с приглашением для введения параметров ядра. Приглашение выглядит так:

```
probing: pc0 com0 com1 apm mem[636k 190M a20=on]
disk: fd0 hd0+
>> OpenBSD/i386 BOOT 2.10
boot>
```

В этот момент надо набрать "boot -s" для входа в однопользовательский режим:

```
boot> boot -s
```

Большинство других платформ передают параметры ядру используя boot ROM.

Конечно, существует проблема корректного выключения питания. Вероятно, придется нажать "Reset" или выключатель питания. Конечно, это нехорошо, но что же делать? Сильно волноваться не надо, файловая система OpenBSD весьма и весьма надежна.

Монтирование разделов. Нам потребуется доступ к "/" и /usr в режиме чтения и записи. Если предположить, что они находятся на разных разделах, надо набрать следующее:

```
# fsck -p / && mount -uw /
# fsck -p /usr && mount /usr
```

Запустите `passwd(1)` для смены пароля root. Вы уже имеете права root (поскольку загрузились в однопользовательском режиме), и не забудьте запомнить свой новый пароль.

Загрузка в многопользовательском режиме. Это можно сделать нажав "CTRL-D" для перехода в нормальный режим загрузки, или выполнив команду `reboot(8)`.

Если это не персональная (личная) машина, вы должны дать права на исполнение `sudo(8)` нескольким доверенным лицам для выполнения команд с привилегиями root.

"Постойте. Это же так просто! И не очень безопасно!". Если злоумышленник имеет физический доступ к вашему компьютеру, вас не спасет никакая ОС. Есть способы принудительной проверки пароля в однопользовательском режиме (смотрите `ttys(5)`), или отключение паузы на i386/amd64 (смотрите `boot.conf`), но, честно говоря, эти хитрости обойти очень легко (Один из вариантов: загрузка с floppy или CDROM и редактирование или замена файла `passwd`). Это тоже можно предотвратить, но, представьте, - кто-нибудь украдет ваш жесткий диск? Хотите безопасности - сделайте ваш компьютер наперво физически безопасным, иначе вся ваша безопасность не более чем фикция.

Замечание: многие "управляемые дистанционно" системы дают возможность использовать большую часть функциональности без физического доступа к компьютеру, и это тоже надо учесть. Не нужно думать, что атака будет заключать в себе лишь доступ к консоли, вставку дискеты и перезагрузку компьютера. Злоумышленники могут также иметь физический доступ к консоли системы управления, которая, скорее всего, далеко не так безопасна, как сама OpenBSD ...

8.2 - X не запускается, я получаю массу сообщений об ошибках

Распространенная причина проблем с X - параметр `sysctl(8)` `machdep.allowaperture`.

Вам нужно отредактировать `/etc/sysctl.conf` и установить `machdep.allowaperture=2` (или `1`, в зависимости от вашей платформы). Это позволит X получить доступ к апертурному драйверу, `xf86(4)`, при следующей перезагрузке. Включить доступ после загрузки невозможно. Этот параметр также можно настроить во время установки системы, если вы ответите "Y" на вопрос о том, предполагаете ли вы запуск Оконной Системы X.

OpenBSD требует активации апертурного драйвера для управления доступом к видеокарте на платформах alpha, amd64, cats, i386, macppc и sparc64. Другие платформы используют безопасный метод работы видео-подсистемы и, поэтому, не нуждаются в нем (он даже отсутствует в ядрах для них). Если вы не предполагаете использование X в вашей системе, рекомендуется не включать апертурный драйвер.

Более подробно конфигурация и использование X на вашей платформе описано в файле `/usr/X11R6/README` вашей установленной системы.

8.3 - Могу ли я использовать язык программирования "L" в OpenBSD?

Поддержку многих распространенных языков программирования можно найти в базовой системе (точнее говоря, в файловых наборах `baseXX.tgz` и `compXX.tgz`), или в системе пакетов и портов. Рекомендуется устанавливать необходимые вам компиляторы из файловый наборов или пакетов, вместо сборки из исходных кодов. Некоторым компиляторам требуется очень много ресурсов для сборки и это просто не имеет смысла, если только перед вами не стоит специфичная задача или необходимых пакетов просто нет.

В следующей таблице приведен обзор компиляторов для разных языков, способы их установки, а также возможные ограничения или проблемы. Некоторые из них ограничены определенными платформами. Это можно увидеть в результатах поиска в дереве портов, смотрите за упоминаниями в "Arch", или посмотреть непосредственно в Makefile порта. В последнем случае, поищите строки, содержащие `ONLY_FOR_ARCHS`, `NOT_FOR_ARCHS`, `BROKEN`, и т.п.

Примечание: Для удобства использования, список, представленный в этом разделе, упорядочен по алфавиту, без учета различий между разными категориями языков программирования. Это не исчерпывающий список всего, что доступно или может быть использовано в OpenBSD. Если вы считаете, что здесь допущены ошибки или что-то не упомянуто, не стесняйтесь [сообщать](#) об этом.

Язык	Где?	Примечания
Awk	base43.tgz, awk(1)	 GNU awk
C, C++	comp43.tgz, gcc(1)	Компиляторы C/C++ в базовой системе проверены и имеют несколько расширений безопасности (напр. ProPolice) включенных по умолчанию. Детали смотрите в gcc-local(1). Они также выдают предупреждения при использовании небезопасных функций, таких как sprintf(), strcpy(), strcat(), tmpnam(), и т.п. Обратите внимание, что большинство платформ используют gcc 3.3.5, но некоторые все еще используют gcc 2.95.3.
C, C++	lang/gcc	Эти компиляторы не прошли через аудит безопасности и не содержат расширений безопасности подобных тем, что присутствуют в базовой системе. Компиляторы переименованы в egcc, eg++, и т.п., чтобы избегать неразберихи с их аналогами из базовой системы
Caml	lang/ocaml	Objective Caml
COBOL	lang/ open-cobol	

```

|Fortran |comp43.tgz,|Только поддержка Fortran 77.
|         |g77(1)      |
|         |-----+-----|
|         |lang/gcc    | Fortran 95 также поддерживается egfortran в gcc 4.0
|         |            | и выше. Этот новый компилятор доступен как
|         |            | часть пакета (g95) gcc.
|-----+-----|
|Haskell |lang/ghc    |
|         |-----+-----|
|         |lang/nhc98  |
|-----+-----|
|Java    |devel/jdk   | Sun JDK - никаких пакетов недоступно; смотри
|         |            | инструкции по сборки ниже
|         |-----+-----|
|         |lang/jikes  |Быстрый компилятор, работающий хорошо. Он нуждается в
|         |            |"run-time jar", байт-код версии всего стандартного API
|         |-----+-----|
|         |devel/      | Большая IDE; работает с Sun JDK
|         |eclipse     |
|-----+-----|
|Lisp    |!lang/clisp |
|-----+-----|
|Lua     |lang/lua    |Дополнительные библиотеки Lua и вспомогательные
|         |            |утилиты доступны в дереве портов.
|-----+-----|
|Perl   |base43.tgz,|Много модулей Perl доступно в дереве портов, так что
|         |perl(1)    |ищите сначала там перед установкой модулей из
|         |            |CPAN.
|-----+-----|
|PHP    |www/php4    |Много подпакетов доступны для различных PHP
|         |-----+-----|модулей.
|         |www/php5    |
|-----+-----|
|Prolog  |lang/       | Компилятор GNU Prolog.
|         |gprolog    |
|-----+-----|
|Python  |lang/python| Другие порты используют по умолчанию Python 2.5.
|-----+-----|
|Ruby   |lang/ruby   |
|-----+-----|
|Scheme  |lang/chicken
!         !lang/scheme48
!         !lang/scm
!         !shells/scsh
|-----+-----|
|Smalltalk|lang/squeak|
|-----+-----|
|Tcl    |lang/tcl    |
|-----+-----|

```

< - - - *вниманию переводчика таблица не отформатирована!*

Сборка Sun JDK

Из-за ограничений Sun-овской лицензии SCSL, OpenBSD не может использовать двоичные пакеты для JDK. Это означает, что вы должны собрать их из портов. Обратите внимание, что для успешной сборки вам потребуется много памяти.

Порты JDK находятся в подкаталоге devel/jdk дерева портов. Вы можете выбирать из разных версий, каждая лежит в своем собственном подкаталоге. Если вы просто наберете make, вы увидите сообщение о том, что вам необходимо скачать файлы исходников вручную с Sun-овского вебсайта. Перед тем как сделать это, вам необходимо зарегистрироваться на этом веб-сайте и согласиться с лицензией. Именно из-за этого система портов не может запустить загрузку автоматически.

Как только вы загрузите необходимые дистрибутивные файлы и наборы патчей, скопируйте их в каталог /usr/ports/distfiles. Вам также необходимо иметь X, установленную на вашей системе. Запустите процесс сборки командой make в подкаталоге порта.

Для сборки JDK требуется рабочий компилятор Java 2 для раскрутки сборки. С этой целью, начиная с 4,0 OpenBSD, порт использует версию JDK kaffe, реализующую версию JDK 1,5, которая будет использоваться на i386 и amd64 платформах, значительно сокращая время сборки.

Предыдущие версии JDK-прежнему нуждаются в JDK для Linux. Эмуляция Linux на OpenBSD ограничивается системами i386, так что JDK можно собрать только на i386. Система портов должна позаботиться об установке необходимых файлов и настройке kern.emul.linux=1. Более подробно об эмуляции вы можете прочитать в странице руководства [compat_linux\(8\)](#), а также "FAQ 9 - Запуск бинарных программ для Linux в OpenBSD". Обратите внимание, что эмуляция Linux требуется только для сборки JDK, в результате чего получается версия JDK для OpenBSD. **Вам не нужна эмуляция Linux, чтобы работать с нативной версией JDK.**

После многих часов сборка завершится. Просто завершите дело make install, чтобы установить JDK.

Если у вас возникли ошибки, такие как "Could not reserve enough space for object heap", попробуйте увеличить ограничение использования памяти процессом, встроенной командой оболочки ulimit, с параметром -d.

Другие средства разработки

Кроме того, есть много других инструментальных средств разработки, доступных в базовой системе или в виде пакетов или портов. Несколько примеров:

Оболочки Unix: ksh и csh в базовой системе, много других (напр zsh, tcsh) в подкаталоге shells дерева портов.

[lint\(1\)](#): верификатор C программ, который был значительно улучшен начиная с OpenBSD 3.9. Также

поставляются версии системных библиотек, прошедших lint.

"make" утилиты: традиционная программа BSD [make\(1\)](#) в базовой системе, а дерево портов содержит другие виды программы, которые требуются для компиляции некоторого программного обеспечения.

Графические наборы инструментов: многие популярные графические пакеты разработчика (напр. GTK+, Qt, wxWidgets,...) перенесены на OpenBSD. Их можно найти в подкаталоге x11 дерева портов.

Системы контроля версий: GNU CVS, как используемый проектом OpenBSD, содержится в базовой системе, а дерево портов содержит некоторые другие. Ожидайте новый [OpenCVS](#), который находится в разработке.

8.4 Что такое дерево портов?

Пожалуйста, смотрите "FAQ15 - Работа с портами".

8.5 - Что такое пакеты?

Пожалуйста, смотрите "FAQ15 - Управление пакетами".

8.6 - Использовать ли мне порты и пакеты?

Пожалуйста, смотрите FAQ15.

8.8 - Существует ли способ использовать привод дискет, если он не был подключен во время загрузки?

Вам необходимо настроить ядро так, чтобы оно всегда предполагало, что привод дискет подключен, даже если он не определяется во время обнаружения аппаратуры, путем установки флага 0x20 на [fdc\(4\)](#). Это можно сделать в [пользовательской конфигурации ядра](#) или с использованием [config\(8\)](#) для изменения вашего ядра,

```
# config -e -f /bsd
OpenBSD 4.3 (GENERIC) #698: Wed Mar 12 11:07:05 MDT 2008
  deraadt@i386.openbsd.org: /usr/src/sys/arch/i386/compile/GENERIC
Enter 'help' for information
ukc> change fd*
254 fd* at fdc0 drive -1 flags 0x0
change [n] y
drive [-1] ? ENTER
flags [0] ? 0x20
254 fd* changed
254 fd* at fdc0 drive -1 flags 0x20
ukc> q
```

```
Saving modified kernel.
```

```
#
```

8.9 - Загрузчик OpenBSD (*архитектуры i386, amd64*)

При загрузке вашей системы OpenBSD, вы, вероятно, обратили внимание на приглашение загрузчика.

```
boot>
```

Для большинства людей на данном этапе делать ничего не требуется. Если никакие команды не вводятся все само автоматически загрузится. Но иногда возникают проблемы или нужны специальные функции. Вот тогда эти опции окажутся полезными. Для начала, вы должны прочитать страницу руководства (map page) [boot\(8\)](#). А сейчас мы рассмотрим наиболее используемые команды для загрузчика.

Для начала, если никакие команды не введены, то загрузчик автоматически попытается загрузить **/bsd**. Если эта попытка терпит неудачу, то попытается **/obsd**, и если эта попытка терпит неудачу, это попробует **/bsd.old**. Вы можете определить ядро вручную напечатав:

```
boot> boot hd0a:/bsd
```

или

```
boot> b /bsd
```

Эта команда загрузит ядро названное bsd из раздела 'a' первого жесткого диска, определенного BIOS'ом.

Вот краткий список опций, который вы можете использовать с ядром OpenBSD.

- a** : Это позволит вам определить альтернативное корневое устройство после загрузки ядра.
- c** : Это позволяет вам вводить [пункт 5.8 - конфигурацию во время загрузки](#). FAQ.
- s** : Это – опция для загрузки в однопользовательском режиме.
- d** : Эта опция используется для дампа ядра в ddb. Имейте в виду, что вы должны иметь поддержку DDB в ядре.

Формат ввода: **boot [image [-acds]**

Для дальнейшего знакомства с данной темой можете прочитать страницу руководства [boot\(8\)](#).

8.10 - S/Key

S/Key - система идентификации "одноразовым паролем". Она может быть полезна для людей кто не имеет возможности использовать зашифрованный канал, который защищает их идентификационные удостоверения в пути, который, например, можно установить, используя [ssh\(1\)](#).

ПРЕДУПРЕЖДЕНИЕ: Системы одноразовых паролей защищают только идентификационную информацию. Они не мешают прослушиванию сети для получения доступа к личной информации. Кроме того, если вы подключаетесь к безопасной системе А, рекомендовано, чтобы вы делали это из другой поверенной системы В, чтобы гарантировать, что никто получает доступ к системе А, регистрируя ваши нажатия клавиши или захватывая и/или подделывая ввод и вывод на ваших терминальных устройствах.

Система S/Key генерирует последовательность одноразовых (единожды используемых) паролей из пользовательской секретной идентификационной фразы вместе с вызовом, полученным от сервера, посредством безопасной хэш функции. Система исключительно безопасна только, если секретная идентификационная фраза никогда не передается через сеть. Следовательно **инициализация или изменение вашей секретной идентификационной фразы ДОЛЖНО быть сделано через защищенный канал**, как, например, [ssh\(1\)](#) или с консоли.

Реализация S/Key в OpenBSD может использовать ряд алгоритмов в качестве односторонней хэш функции. Доступны следующие алгоритмы:

- [md4](#)
- [md5](#)
- [sha1](#)
- [rmd160](#).

Установка S/Key - Первые шаги

Для начала должен существовать каталог `/etc/skey`. Если этот каталог не существует, обратитесь к супер-пользователю для его создания. Это может быть сделано с помощью команды:

```
# skeyinit -E
```

Как только этот каталог будет существовать, вы сможете инициализировать ваш S/Key. Для этого вы должны использовать [skeyinit\(1\)](#). Так как `skeyinit(1)` спросит вас о вашей S/Key секретной идентификационной фразе, вы должны запустить эту команду **через безопасный канал**, как объяснено выше! Программа даже напомнит вам об этом. С помощью `skeyinit(1)`, вы сначала будете спрошены о вашем пароле для входа в систему. Это - тот же пароль, которым вы обычно логинитесь в систему. Как только вы идентифицируете себя с вашим системным паролем, вас спросят о вашей S/Key секретной идентификационной фразе. Это - **НЕ** ваш системный пароль. Ваша секретная идентификационная фраза, должна быть по крайней мере 10 символов. Мы предлагаем использовать не забывающуюся фразу, содержащую несколько слова в качестве секретной идентификационной фразы. Вот пример добавления пользователя.


```

$ skeyinit
Reminder - Only use this method if you are directly connected
           or have an encrypted channel.  If you are using telnet,
           exit with no password and use skeyinit -s.
(
Напоминание - Использование этого метода, только если вы соединены напрямую
              или имеете зашифрованный канал. Если Вы используете telnet,
              выходите, не вводя пароля, и используйте skeyinit -s.
)
Password:
[Adding ericj with md5]
Enter new secret passphrase:
Again secret passphrase:

ID ericj skey is otp-md5 100 oshi45820
Next login password: HAUL BUS JAKE DING HOT HOG

```

Одна строка особой важности здесь это *ID ericj skey - otp-md5 100 oshi45820*. Она дает много информации пользователю. Вот анализ секций и их значений.

otp-md5 - какой односторонний хэш был использован для создания вашего одноразового пароля One-Time Password(otp).

100 - Это - ваш номер последовательности. Это - число от 100 до 1. Как только оно достигнет 1, другая секретная идентификационная фраза должна быть создана командой [skeyinit\(1\)](#).

oshi45820 - Это - ключ.

Но более непосредственное значение - ваш одноразовый пароль. Ваш одноразовый пароль состоит из 6 небольших слов, сгруппированных вместе это - ваш одноразовый пароль, пробелы и все. Одноразовый пароль напечатанный skeyinit не может быть использован для входа в систему (об использовании для этого первого одноразового пароля, смотри [skeyinit\(1\)](#)). Для возможности залогиниться, одноразовый пароль, соответствующий приглашению, процессом входа, должен быть вычислен, используя [skey\(1\)](#). В следующей секции рассказывается как это делается.

Фактическое использование S/Key для входа в систему.

Теперь ваш skey был инициализирован. Вы готовы войти в систему. Вот Вот сеанс примера, использовавший S/Key для входа в систему. Чтобы выполнять S/Key вход, вы добавляете **":skey"** в ваш логин.

```

$ ftp localhost
Connected to localhost.
220 oshibana.shin.ms FTP server (Version 6.5/OpenBSD) ready.
Name (localhost:ericj): ericj:skey
331- otp-md5 96 oshi45820
331 S/Key Password:

```

```
230- OpenBSD 4.3 (GENERIC) #698: Wed Mar 12 11:07:05 MDT 2008
230-
230- Welcome to OpenBSD: The proactively secure Unix-like operating system.
230-
230- Please use the sendbug(1) utility to report bugs in the system.
230- Before reporting a bug, please try to reproduce it with the latest
230- version of the code. With bug reports, please try to ensure that
230- enough information to reproduce the problem is enclosed, and if a
230- known fix for it exists, include that as well.
230-
230 User ericj logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> quit
221 Goodbye.
```

Отметьте, что я добавил ":skey" в свое имя пользователя. Это сообщает ftpd, что я хочу аидентифицироваться, использование S/Key. Некоторые из вас могли обратить внимание, что мой номер последовательности изменился на `otp-md5 96 oshi45820`. Дело в том, что в данный момент, я использовал S/Key для входа в систему несколько раз. Но как вам получить ваш одноразовый пароль? Хорошо, чтобы вычислить одноразовый пароль, вам понадобится знать какой номер последовательности вами используется и ваш ключ. Как вы вероятно думаете, как можете вы запоминать с каким номером последовательности вы имеете дело?

Когда вы логинитесь, процесс входа печатает строку, содержащую нужную информацию, которую вы можете использовать, чтобы генерировать одноразовый пароль на месте, используя другой доверенный компьютер, доступный по безопасному каналу, копи-паст (скопируйте) строку в командную оболочку:

```
otp-md5 96 oshi45820
```

После того, как наберете вашу идентификационную фразу, ваш одноразовый пароль будет напечатан, который вы можете затем копия-паст (скопируйте) в приглашение ввода пароля S/Key, чтобы войти в систему. `otp-md5` - не только описание использованного хэша, это также альтернативное название для [skey\(1\)](#).

Если Вы уже залогинены и хотите сгенерировать одноразовый пароль для следующего входа, используйте [skeyinfo\(1\)](#), оно сообщит вам, что надо использоваться для следующего входа. Например, мне нужно сгенерировать другой одноразовый пароль для входа, который я могу совершить в будущем. (запомните, что я делаю это через безопасный канал).

```
$ skeyinfo
95 oshi45820
```

Даже лучший путь - это использование команды **skeyinfo -v**, которая выводит команду пригодную для запуска в оболочке. Например:

```
$ skeyinfo -v
otp-md5 95 oshi45820
```

Так что, самый простой путь сгенерировать следующий S/Key пароль - просто:

```
$ `skeyinfo -v`
Напоминание - не использовать эту программу пока залогинены через telnet.
Enter secret passphrase:
NOOK CHUB HOYT SAC DOLE FUME
```

Обратите внимание на обратные кавычки в вышеуказанном примере.

Я уверен, что многие из вас не всегда смогут иметь безопасное соединение или надежный локальный компьютер, чтобы создавать эти пароли, а создание их через небезопасное соединение не осуществимо, так что можете ли вы создать много паролей за один раз? Вы можете снабдить skey(1) числом паролей, которое вы хотите создать. Они могут затем быть распечатаны и взяты с собой куда бы вы ни пошли.

```
$ otp-md5 -n 5 95 oshi45820
Напоминание - не использовать эту программу пока залогинены через telnet.
Enter secret passphrase:
91: SHIM SET LEST HANS SMUG BOOT
92: SUE ARTY YAW SEED KURD BAND
93: JOEY SOOT PHI KYLE CURT REEK
94: WIRE BOGY MESS JUDE RUNT ADD
95: NOOK CHUB HOYT SAC DOLE FUME
```

Заметим здесь все-же, что сначала должен быть использован нижний пароль , поскольку мы считаем в обратном порядке от 100.

Использование S/Key с ssh(1) и telnet(1)

Использование S/Key с ssh(1) или telnet(1) сделано в той же форме, что и с ftp -- вы просто добавляете ":skey" в конец вашего имени пользователя. Пример:

```
$ ssh -l ericj:skey localhost
otp-md5 98 oshi45821
S/Key Password: SCAN OLGA BING PUB REEL COCA
Last login: Thu Apr 7 12:21:48 on ttypl from 156.63.248.77
OpenBSD 4.3 (GENERIC) #698: Wed Mar 12 11:07:05 MDT 2008
```

Welcome to OpenBSD: The proactively secure Unix-like operating system.

Please use the `sendbug(1)` utility to report bugs in the system. Before reporting a bug, please try to reproduce it with the latest version of the code. With bug reports, please try to ensure that enough information to reproduce the problem is enclosed, and if a known fix for it exists, include that as well.

8.12 - Поддерживает ли OpenBSD SMP? (Symmetric Multi-Processor)

SMP поддерживается на [OpenBSD/i386](#), [OpenBSD/amd64](#), [OpenBSD/mvme88k](#), и [OpenBSD/sparc64](#) платформах

Отдельное ядро SMP, "bsd.mp", содержится в установочном комплекте, который может быть выбран во время установки. Рекомендуем вам протестировать загрузку этого ядра перед переименованием его в "bsd" для установки его ядром по умолчанию.

Надеемся, что другие SMP-совместимые платформы будут поддерживаться в будущем. На большинстве других платформ, OpenBSD будет работать на SMP системе, но только используя один процессор. Исключение из этого - платформа [SPARC](#) -- OpenBSD/sparc иногда требует, чтобы дополнительные модули MBus были удалены из системы, чтобы загрузиться.

Поддержка для процессоров UltraSPARC T1 имеется в 4.3-current.

8.13 - Я получаю ошибки ввода/вывода при попытке использования моих устройств tty

Вам нужно использовать `/dev/cuaXX` для соединения инициированного из системы OpenBSD, устройства `/dev/ttyXX` предназначены только для терминального или dial-in (входящего) использования. Ядро OpenBSD больше не совместимо с прежним "совместным" вариантом использования, как это было возможно для tty устройств в прошлом.

Из [cua\(4\)](#):

Для аппаратных терминальных портов, dial-out (исходящие вызовы) поддерживается через согласующие устройства узлов, называемых вызываемыми устройствами (matching device nodes called calling units). Например, терминал называемый `/dev/tty03` должен иметь согласующее вызываемое устройство называемое `/dev/cua03`. Эти два устройства обычно различаются путем созданием узла вызываемого устройства с номером устройства большим 128, чем dial-in (входящего) узла устройства. *Поскольку dial-in (входящему) устройству (tty) обычно требуется аппаратный сигнал, чтобы указать системе, что оно активно, dial-out (исходящему) устройству (cua) нет, и следовательно можно передавать беспрепятственно с устройства такого как модем.* Это означает, что процесс подобный [getty\(8\)](#) ожидает на dial-in (входящем) устройстве пока соединение не будет установлено. Тем временем, dial-out (исходящее) соединение может быть установлено на dial-out (исходящем) устройстве (для того же аппаратного терминального порта), не мешая чему-либо еще в системе. Процесс [getty\(8\)](#) даже не уведомляется, что что-либо происходит в терминальном порту. Если вызов

соединения приходит после того, dial-out (исходящее) соединение завершилось, процесс [getty\(8\)](#) будет иметь дело с ним правильно, без уведомленного промежуточного dial-out (исходящего) действия.

8.14 - Какие веб браузеры доступны для OpenBSD?

[Lynx](#), текстовый браузер, - в базовой системе, имеет поддержку SSL. Другие браузеры доступные в дереве портов, включают (список без конкретного порядка):

Графические (X) браузеры

[Konqueror](#) устанавливается как часть [окружения KDE](#).

[Konqueror-embedded](#) (konq-e) Konqueror, использующий только библиотеки KDE, а не весь KDE.

[Links+](#) Другой быстрый и компактный графический браузер. (Также есть и в текстовом режиме)

[Firefox](#) and [SeaMonkey](#) Богатые возможностями браузеры. SeaMonkey много дополнительных перимуществ, не свойственных браузерам (почтовый клиент, IRC клиент и другие), Firefox только браузер на движке Mozilla. Работают на многих архитектурах.

[Opera](#) Коммерческий браузер, только для i386. (Требуется эмуляция Linux).

[Amaya](#) Браузер W3C и редактор.

Консольные браузеры (текстовый режим)

[elinks](#) Богатый возможностями, может отображать как фреймы так и таблицы, высокая настраиваемость.

[w3m](#) Имеет поддержку таблиц и фреймов (также имеет графический режим).

[links](#) Имеют поддержку таблиц.

Вы можете найти все это в коллекции пакетов. Все вышеупомянутые браузеры после установки дерева портов расположены в `/usr/ports/www/`. Большинство из них также доступны в качестве предварительно скомпилированных пакетов, доступных на серверах [FTP](#) и на [CD-ROM](#). Поскольку большинство графических браузеров очень большие требуют некоторого времени, чтобы их скачать и скомпилировать, следует *предпочсть* вариант использование пакетов, когда они доступны.

8.15 - Как мне использовать редактор mg?

Mg - включенный в OpenBSD текстовый микроредактор в стиле Emacs. Микро означает, что он небольшой (Emacs очень большой!) Для начала прочитайте [mg\(1\)](#) страницу руководства и [учебное пособие](#), которое поставляется вместе с исходным кодом. Для более интересных вопросов (как например, "У меня нет Мета ключа!" ("I don't have a Meta key!")) ознакомьтесь с [Emacs FAQ](#).

Заметьте, что так как `mg` - маленькая реализация Emacs, которая по большей части подобна текстовым характеристикам редактора Emacs 17, она не обеспечивает многих других функций, выполняемых Emacs. (Включая почту и новостная функциональность, а также режимы для Lisp, C++, Lex, Awk, Java, и т.п...)

8.16 - ksh(1), по-видимому, не читает мой .profile!

Есть две возможные причины для `ksh(1)` игнорировать файл `.profile` с настройками пользователя.

`.profile` не принадлежит пользователю. Устраните ошибку для пользователя **usernsme**,

```
# chown username ~username/.profile
```

Вы используете `ksh(1)` из Оконной Системы X

Из-под `xterm(1)`, `argv[0]` для `ksh(1)` не добавлено спереди тире ("-"). Добавленное спереди тире перед `argv[0]` послужит причиной, чтобы `csh(1)` и `ksh(1)` узнали, они должны интерпретировать их файлы входа. (Для `csh(1)` - `.login`, с отдельным `.cshrc`, который всегда запускается, когда `csh(1)` стартует. С `ksh(1)`, это более достойно внимания, поскольку есть только один сценарий запуска, `.profile`. Этот файл игнорируется если оболочка не является оболочкой входа (а `login shell`)).

Для устранения проблемы добавьте строку `"XTerm*loginShell: true"` в файл `.Xdefaults` в вашем домашнем каталоге. Примечание: этот файл не существует по умолчанию, вы можете создать его.

```
$ echo "XTerm*loginShell: true" >> ~/.Xdefaults
```

Вы возможно не должны делать это в других системах, так как некоторые установки Оконной Системы X поставляются с этой установкой по умолчанию. OpenBSD решил следовать за образом действий X.org.

8.17 - Почему мой файл /etc/motd перезаписывается, хотя я модифицировал его?

Файл `/etc/motd` редактируется при каждой загрузке системы, заменяя все, но не включая, первую пустую строку с информацией версии системного ядра. При редактировании этого файла, убедитесь, что вы начинаете после этой пустой строки, чтобы уберечь `/etc/rc` от удаления этих строк, когда скрипт редактирует `/etc/motd` в процессе загрузки.

8.18 - Почему сайт www.openbsd.org работает на Solaris?

Хотя ни один из разработчиков не думает, что это особо важно, этот вопрос достаточно часто возникает в списках почтовой рассылки, так что ответ находится здесь. www.openbsd.org и основное ftp хранилище OpenBSD hosted (хостится) в SunSITE в Университете Альберта (Альберта - провинция Канады), в Канаде. Эти сайты hosted (хостятся) в большой системе Sun, которая имеет доступ к большому пространству для хранения информации и Internet большой пропускной способностью. Присутствие на SunSITE дает групповой доступ OpenBSD к этой пропускной способности. Это причина почему основной сайт работает здесь. Много зеркал сайта OpenBSD работают на OpenBSD, но так как они не гарантируют доступ к этому каналу с большой пропускной способностью, группа решила запустить основной сайт на SunSITE Университета Альберта.

8.19 - Шрифты Антиалиас и TrueType в X

Смотри [этот документ](#).

8.20 - Поддерживает ли OpenBSD какие-либо журналируемые файловые системы?

Нет. Мы используем другой механизм, чтобы достигать аналогичных результатов, который называется [Soft Updates](#). Пожалуйста прочитайте в FAQ 14, чтобы получить больше подробностей.

Обратное преобразование DNS

- или -

Почему мне приходится так долго ждать, чтобы войти в систему?

Много новых пользователей OpenBSD испытывают примерно двухминутные задержки при входе (login), при использовании таких сервисов, как например, [ssh](#), [ftp](#), или [telnet](#). Это может также проявляться при использовании прокси, таких, как например, [ftp-proxy](#), или при отправке письма с рабочей станции через [sendmail](#).

Это почти всегда происходит из-за проблемы обратного преобразования DNS. DNS - Domain Name Services (службы имен доменов), используемой системой Internet для преобразования имен, как например, "www.openbsd.org" в числовой адрес IP. Другая задача DNS - способность преобразовывать числовой адрес назад в "имя", это - "Обратное преобразование DNS".

Для того, чтобы обеспечивать лучший вход в систему, OpenBSD выполняет обратный DNS поиск в любой машине, к которой подключаются многими различными путями, включая [ssh](#), [ftp](#), [telnet](#), [sendmail](#) или

[ftp-proxy](#). К несчастью, в некоторых случаях, машина, которая делает соединение не имеет соответствующей обратной DNS записи.

Пример такой ситуации:

Пользователь устанавливает OpenBSD в качестве файрвола и шлюза в его внутреннюю домашнюю , отображающую все их внутренние компьютеры на один внешний IP, используя NAT. Они могут также использовать его в качестве исходящего почтового реле. Пользователи следуют руководству по установке, и очень счастливы результатам, за исключением одной вещи - каждый раз, когда они пытаются подключиться к блоку любым путем, их попытки заканчиваются двух минутной задержкой прежде, чем происходит подключение.

Что происходит

:

С рабочей станции за NAT шлюзом с [незарегистрированного IP](#) адреса 192.168.1.35, пользователь использует [ssh](#) для доступа к межсетевой системе. Клиент ssh считывает имя пользователя и пароль, и посылает им межсетевому блоку. Шлюз затем пытается вычислить, кто пытается войти, выполняя запрос на обратное преобразование DNS 192.168.1.35. Проблема в том, что 192.168.0.0 адреса для частного использования, так что правильно сконфигурированный сервер DNS за пределами вашей сети знает, что не может иметь информации о таких адресах. Некоторые быстро возвращают сообщение об ошибке, в этих случаях, OpenBSD примет, что нельзя добыть больше информации, и он быстро махнет на это рукой и просто допустит пользователя. Другие серверы DNS не возвращает КАКОЙ-ЛИБО ответ. В этом случае вы начнете ждать распознавателя имени OpenBSD до таймаута, который длится около двух минут прежде, чем вход будет разрешен для продолжения работы. В случае [ftp-proxy](#), некоторые ftp клиенты получают таймаут прежде запроса обратного преобразования DNS, создавая впечатление, что ftp-proxy не работает.

Это может довольно рассердить. К счастью, это легко подправить.

Исправление, использующее

[/etc/hosts](#):

Самое простое исправление это заполнить ваш файл [/etc/hosts](#) всеми рабочими станциями, имеющимися в вашей внутренней сети, и проверить, что ваш файл [/etc/resolv.conf](#) содержит, строку lookup file bind, которая гарантирует, что распознаватель знает, что начинать нужно с файла [/etc/hosts](#), и в противном случае, чтобы использовать серверы DNS, определенные строками "nameserver" в вашем файле [/etc/resolv.conf](#).

Ваш файл [/etc/hosts](#) будет выглядеть приблизительно так:

```
:::1 localhost.in.example.org localhost
127.0.0.1 localhost.in.example.org localhost
192.168.1.1 gw.in.example.org gw
```



```
192.168.1.20 scrappy.in.example.org scrappy
192.168.1.35 shadow.in.example.org shadow
```

Ваш resolv.conf файл будет выглядеть приблизительно так:

```
search in.example.org
nameserver 24.2.68.33
nameserver 24.2.68.34
lookup file bind
```

Распространенное возражение этому "Да это понятно, но я использую DHCP для своей внутренней сети! Как могу я сконфигурировать свои /etc/hosts?" Предпочтительно попроще, и актуальней. Просто введите строки для всех адресов вашего сервера DHCP, перебрав их все, плюс любые статические устройства:

```
:::1 localhost.in.example.org localhost
127.0.0.1 localhost.in.example.org localhost
192.168.1.1 gw.in.example.org gw
192.168.1.20 scrappy.in.example.org scrappy
192.168.1.35 shadow.in.example.org shadow
192.168.1.100 d100.in.example.org d100
192.168.1.101 d101.in.example.org d101
192.168.1.102 d102.in.example.org d102
    [... вырезано ...]
192.168.1.198 d198.in.example.org d198
192.168.1.199 d199.in.example.org d199
```

В этом случае, я предполагаю, что вы имеете DHCP диапазон установленным от 192.168.1.100 до 192.168.1.199, плюс три статических определения как указано в начале файла.

Если ваш шлюз должен использовать DHCP для конфигурации, у вас могут возникнуть проблемы -- [dhclient](#) перезаписывает ваш /etc/resolv.conf каждый раз, когда выделяемый адрес обновляется, который также удалит строку "lookup file bind". Это может быть решено размещением "lookup file bind" в файле /etc/resolv.conf.tail.

Исправление, использующее локальный сервер DNS

Детали этого лежат отчасти за рамками этого документа, но основная хитрость - это установить ваш любимый сервер DNS, и убедиться, что он знает, что он надежен как для прямого так и обратного преобразования DNS для всех узлов в вашей сети, и убедиться, что ваши компьютеры (включая ваш шлюз) знают, что надо использовать его в качестве сервера DNS.

8.23 - Почему веб страницы OpenBSD не соответствуют HTML4/XHTML?

Настоящие веб страницы были тщательно и искусно написаны, чтобы работать на широком спектре действующих браузеров, восходящих к браузерам версий 4.0 и позже. Мы не хотим делать эти более старые страницы соответствующими HTML4 или XHTML до тех пор пока мы не будем уверены, что они также будут работать с более старыми браузерами; это - просто не приоритет. Мы приветствуем новых участников, но советуем вам работать над написанием кода, или над документированием новых возможностей системы, а не над тонкой настройкой (tweaking) существующих веб страниц для соответствия новым стандартам.

Почему мои часы отличаются на двадцать с лишним секунд?

При использовании `rdate(8)`, чтобы синхронизировать ваши часы с сервером NTP, вы можете найти, что ваши часы отличаются на двадцать с лишним секунды от вашего локального определения времени.

Это вызвано различием между UTC временем (Согласованное Универсальное Время (Coordinated Universal Time), основанном на астрономических наблюдениях) и TAI временем (Международное Атомное Время (International Atomic Time), основанном на атомных часах). Для того, чтобы компенсировать изменения земного вращения, "високосные секунды" (leap seconds) вставляются в UTC, но TAI это не регулируется. Эти високосные секунды (leap seconds) являются причиной этого несоответствия. Для более подробного описания, поищите в сети "leap seconds UTC TAI".

Обхождение проблемы довольно просто. В большинстве странах вы получите правильное время, если вы используете параметр "-c" для `rdate(8)` и используете часовой пояс из каталога `/usr/share/zoneinfo/right/`. Например, если вы находитесь в Германии, вы могли бы использовать эти команды:

```
# cd /etc && ln -sf /usr/share/zoneinfo/right/CET localtime
# rdate -ncv ptbtime1.ptb.de
```

В других странах, правила могут отличаться.

8.25 - Почему мои часы отличаются на несколько часов?

OpenBSD предполагает, что ваши аппаратные часы настроены по UTC (Universal Coordinated Time), а не по местному времени, которое установила другая операционная система, установленная на вашем компьютере - читайте Установка OpenBSD - Multibooting.

Большинство других операционных систем может быть настроено по тому же принципу, что и OpenBSD, чтобы вообще избегать подобного рода проблем.

Если это - проблема (например, если ваша система загружается с несколькими другими операционными системами), вы можете изменить поведение по умолчанию, используя `config(8)`. Например, чтобы сконфигурировать OpenBSD для использования аппаратные часы установленных на US/Eastern (5 часов от UTC, 300 минут):

```
# config -ef /bsd
OpenBSD 4.3 (GENERIC) #698: Wed Mar 12 11:07:05 MDT 2008
  deraadt@i386.openbsd.org: /usr/src/sys/arch/i386/compile/GENERIC
Enter 'help' for information
ukc> timezone 300
timezone = 300, dst = 0
ukc> quit
Saving modified kernel.
```

Посмотрите [options\(4\)](#) и поищите опции "TIMEZONE=value" для подробной информации.

Обычно, часовой пояс настраивается в течение установки. Если вам нужно изменить часовой пояс, вы можете создать новую символическую ссылку на подходящий файл часового пояса в `/usr/share/zoneinfo`. Например, чтобы настроить машину для использования EST5EDT в качестве нового локального часового пояса:

```
# ln -fs /usr/share/zoneinfo/EST5EDT /etc/localtime
```

Смотрите также:

[date\(1\)](#)

["Почему мои часы отличаются на двадцать с лишним секунд?"](#)

[OpenBSD's NTPD](#)

Перевод соответствует \$OpenBSD: [faq8.html](#),v 1.203 2008/05/02 10:58:59 steven Exp \$
