



## 15 - Система портов и пакетов OpenBSD

[printable page](#)

### 15.1 - Введение

Большая часть стороннего софта доступна для использования в OpenBSD. Для того, чтобы сделать эти программы простыми в установке и управлении, а также помочь им следовать политике и целям OpenBSD, этот сторонний софт *портируется* на OpenBSD. Такое портирование включает в себя множество различных вещей. К примеру: создание софта, использующего характерное для OpenBSD расположение файлов (конфигурационные файлы хранятся в /etc), приспособливание к разделяемым библиотекам, создание безопасного, насколько это возможно, софта и так далее.

Конечным результатом такого портирования является готовый к установке бинарный пакет. Основной целью системы пакетов является содержание установленного софта, таким способом, чтобы его можно было в любое время легко обновить или удалить. Такой подход, без потери файлов, помогает пользователям поддерживать систему в чистоте. Система пакетов также помогает гарантировать, что ничего не будет удалено случайным образом, останавливая работу софта должным образом. Другая особенность заключается в том, что **пользователям необязательно собирать софта из исходников**, так как пакеты содержат уже скомпилированный и готовый к работе на OpenBSD софт. За считанные минуты может быть скачано и правильно установлено большое количество пакетов.

**Коллекция пакетов и портов не проходит столь тщательного аудита как ОС OpenBSD. Но все же мы стремимся сохранять качество коллекции пакетов на высоте, мы просто не имеем достаточного ресурса людей, чтобы гарантировать такой же уровень работоспособности и безопасности.** Конечно, обновления безопасности для различных приложений коммитятся в дерево портов как можно скорее и corresponding package security updates are made available as snapshots for *-current*.

### 15.2 - Управление пакетами

#### 15.2.1 - Как это работает?

Пакеты - это бинарники наиболее используемого стороннего доступного софта. Пакеты могут легко управляться с помощью различных утилит, таких как pkg\* tools:

[pkg\\_add\(1\)](#) - утилита установки и обновления пакетов.

[pkg\\_delete\(1\)](#) - утилита удаления установленных ранее пакетов.

[pkg\\_info\(1\)](#) - утилита отображения информации об установленных пакетах.

[pkg\\_create\(1\)](#) - утилита создания пакетов.

Допустим для корректного запуска приложению X требуется наличие приложений Y и Z. Приложение X говорит о своей зависимости от этих приложений, именно по этому Y и Z называют *зависимостями* X. Y может требовать наличия P и Q, а Z в свою очередь наличие R для корректного функционирования. Таким образом формируется дерево *зависимостей*.

Пакеты выглядят подобно простым .tgz пакетам. Но есть одно очень важное отличие: пакет содержит важную дополнительную *информацию для распаковки*. Эта информация используется `pkg_add(1)` для нескольких целей:

Различных проверок: не установлен ли уже этот пакет или не конфликтует ли он с уже установленными пакетами?

Нахождения зависимостей. Необходимые в данный момент будут автоматически скачены и установлены прежде, чем продолжится установка текущего пакета.

Добавления информации о новом пакете в основную базу, по умолчанию /var/db/pkg/. Это гарантия того, что зависимости не будут удалены раньше, чем основной пакет. This helps ensure that an application cannot be accidentally broken by a careless user.

## 15.2.2 - Делаем вещи простыми: PKG\_PATH

Вы можете "упростить себе жизнь" используя переменную окружения PKG\_PATH. Просто укажите местоположение ваших пакетов, и `pkg_add(1)` автоматически просмотрит эту директорию в поиске необходимого пакета, а так же скачает и установит все необходимые зависимости для этого пакета.

Список возможных мест для скачивания пакетов приведен в следующем разделе.

Пример 1: Получение с [CDROM](#), смонтированного в /mnt/cdrom

```
$ export PKG_PATH=/mnt/cdrom/3.9/packages/`machine` -a`/
```

Пример 2: Получение с ближайшего [FTP зеркала](#)

```
$ export PKG_PATH=ftp://your.ftp.mirror/pub/OpenBSD/3.9/packages/`machine` -a`/
```

Хорошей практикой является определение переменной PKG\_PATH в ~/.profile. Она определяется подобно переменной окружения PATH. **ОДНАКО, каждый путь в PKG\_PATH должен содержать в конце косой слэш (/).** Такой путь `pkg_add(1)` раскроет верно, даже если URL содержит двоеточие. Если первая запись в PKG\_PATH ошибочна, то будет использована следующая, и так до тех пор, пока пакет не будет найден. Если все записи ошибочны, то будет выдана ошибка.

Использование `machine(1)` в командной строке позволяет автоматически определить архитектуру машины, а следовательно и устанавливаемого пакета Конечно, если вы используете snapshots, вы должны заменить "4.3" на "snapshots".

## 15.2.3 - Поиск пакетов

Большая коллекция заранее скомпилированных пакетов доступна для наиболее распространенных архитектур. Просто посмотрите необходимый пакет в этих местах:

В дереве [CD-ROM](#), содержащую вашу архитектуру. На CD-ROM-ах находятся только наиболее необходимые в использовании пакеты и только для наиболее распространенных архитектур.

На [FTP-зеркала](#) пакеты располагаются `/pub/OpenBSD/4.3/packages`. Оттуда пакеты разложены по архитектурам.

В списке пакетов на веб-сайте OpenBSD:

[Packages for OpenBSD 4.3](#)

[Packages for OpenBSD 4.2](#)

[Packages for OpenBSD 4.1](#)

Если у вас установлена коллекция портов, вы можете быстро найти необходимый пакет, используя поиск по дереву портов.

Вы заметите, что некоторые пакеты доступны в нескольких различных вариантах, называемых флаворами **flavors**. Другие части некоторых приложений могут быть установлены отдельно. Они называются субпакеты **subpackages**. Более детально они будут рассмотрены в разделе "Использование флавора и субпакетов". На самом деле флавор означает конфигурирование с различной установкой опций. Сейчас многие пакеты имеют флавор, к примеру: поддержка БД, поддержка системы без среды X, поддержка SSL или IPV6. Каждый флавор пакета будет иметь различный суффикс в названии этого пакета. Более детальную информацию об именах пакетов можно прочесть в [packages-specs\(7\)](#).

**Замечание:** Не все возможные пакеты обязательно доступны через FTP сервера. Некоторые приложения просто не работают на всех архитектурах, некоторые не могут распространяться через FTP (или CD-ROM) по лицензионным причинам. Также возможно, что порт имеет множество различных комбинаций флавора, и у проекта OpenBSD просто не хватает ресурсов собрать их все. Если вам необходима комбинация, которая не доступна, то вы можете собрать порт из исходников. Для более подробной информации как это сделать прочтите Использование флавора и субпакетов в разделе портов этого документа.

## 15.2.4 - Установка новых пакетов

Для установки пакетов используется утилита [pkg\\_add\(1\)](#). Если вы **облегчили себе жизнь** заданием переменной `PKG_PATH`, то можете просто вызвать `pkg_add(1)` с именем пакета, как это показано в примере.

```
$ sudo pkg_add -v screen-4.0.3p0
parsing screen-4.0.3p0
installed /etc/screenrc from /usr/local/share/examples/screen/screenrc | 71%
screen-4.0.3p0: complete
```

В этом примере используется флаг `-v` для отображения более подробной информации. Эта опция не нужна, но она помогает при отладке, и была установлена для демонстрации, что же все таки делает `pkg_add(1)`. Заметьте, в сообщении упоминается `/etc/screenrc`. Многократная установка флага `-v` выдаст еще более детальную информацию.

### Использование `pkg_add(1)` в интерактивном режиме

Начиная с OpenBSD 3.9, pkg\_add(1) поддерживает интерактивный режим, который может быть включен установкой флага **-i**, и который будет задавать вам вопросы, когда утилита не может принять решение сама. Например, если вы не знаете число версий пакета заранее, то можете попробовать сделать что-то наподобие этого:

```
$ sudo pkg_add -i screen
Ambiguous: screen could be screen-4.0.3p0 screen-4.0.3p0-shm screen-4.0.3p0-static
Choose one package
  0: <None>
  1: screen-4.0.3p0
  2: screen-4.0.3p0-shm
  3: screen-4.0.3p0-static
Your choice: 1
screen-4.0.3p0: complete
```

Для некоторых пакетов отображается важная информация о конфигурации или использовании приложения на OpenBSD. Если информация действительно важна, то будет показана, даже если флаг **-v** не указан. Вот один из таких случаев:

```
$ sudo pkg_add ghostscript-fonts-8.11
ghostscript-fonts-8.11: complete
You may wish to update your font path for /usr/local/share/ghostscript/fonts
--- ghostscript-fonts-8.11 -----
To install these fonts for X11, just make sure that the fontpath
lists the 75dpi or 100dpi bitmap fonts before the ghostscript fonts,
and make sure you have the string ":unscaled" appended to the bitmap
font's fontpath. This way, the bitmap fonts will be used if they
match, and the Type 1 versions will be used if the font needs to be
scaled. Below is the relevant section from a typical xorg.conf file.

FontPath  "/usr/X11R6/lib/X11/fonts/misc/"
FontPath  "/usr/X11R6/lib/X11/fonts/75dpi:unscaled"
FontPath  "/usr/X11R6/lib/X11/fonts/100dpi:unscaled"
FontPath  "/usr/local/lib/X11/fonts/ghostscript/"
FontPath  "/usr/X11R6/lib/X11/fonts/Type1/"
```

А теперь продолжим примером пакета, имеющего зависимости:

```
$ sudo pkg_add -v tin-1.8.2p0
parsing tin-1.8.2p0
Dependencies for tin-1.6.2 resolve to: gettext-0.14.6, libutf8-0.8, pcre-6.4p1, libiconv-1.9.2p3 \
      (todo: libiconv-1.9.2p3,gettext-0.14.6,pcre-6.4p1,libutf8-0.8)
tin-1.8.2p0:parsing libiconv-1.9.2p3
tin-1.8.2p0:libiconv-1.9.2p3: complete
tin-1.8.2p0:parsing gettext-0.14.6
Dependencies for gettext-0.14.6 resolve to: expat-2.0.0, libiconv-1.9.2p3 (todo: expat-2.0.0)
tin-1.8.2p0:parsing expat-2.0.0
```

```
tin-1.8.2p0:expat-2.0.0: complete
tin-1.8.2p0:gettext-0.14.6: complete
tin-1.8.2p0:parsing pcre-6.4p1
tin-1.8.2p0:pcre-6.4p1: complete
tin-1.8.2p0:parsing libutf8-0.8
tin-1.8.2p0:libutf8-0.8: complete
tin-1.8.2p0: complete
```

Опять мы используем флаг **-v** для того, чтобы увидеть что происходит. После исследования распаковочной информации, были сначала найдены и установлены необходимые зависимости. Вы также можете увидеть, что был установлен пакет `gettext`, который имеет зависимость `libiconv`. Прежде чем установить пакет `gettext` извлечена распаковочная информация и проверено наличие установленного `libiconv`.

Также возможно указывать несколько имен пакета в строке, они будут установлены однократно со всеми зависимостями.

Если в силу каких-то причин, вы не используете переменную `PKG_PATH`, то можно указать абсолютный путь к пакету в командной строке. Этот путь может быть как локальным путем, так и ссылкой на FTP, HTTP, или SCP. Посмотрим пример установки через FTP в следующем примере:

```
$ sudo pkg_add ftp://ftp.openbsd.org/pub/OpenBSD/4.3/packages/`machine` -a`/screen-4.0.3p0.tgz
screen-4.0.3p0: complete
```

Флаг **-v** не был использован, поэтому отобразились только необходимые сообщения. Обратим внимание - имя файла завершается добавлением суффикса **.tgz**. Суффикс можно опустить, `pkg_add(1)` добавит его автоматически.

**Замечание:** Не все архитектуры имеют одинаковые пакеты. Некоторые порты не работают на определенных архитектурах, и производительность ограничена числом пакетов, которые могут быть использованы на других.

Внимание, `pkg_add` **не** перезаписывает модифицированные конфигурационные файлы повторно устанавливаемых пакетов. Информацию о таких файлах можно получить, используя флаг **-v**:

```
$ sudo pkg_add -v screen-4.0.3p0
parsing screen-4.0.3p0
The existing file /etc/screenrc has NOT been changed** | 71%
It does NOT match the sample file /usr/local/share/examples/screen/screenrc
You may wish to update it manually
screen-4.0.3p0: complete
```

Временами вы можете сталкиваться с ошибкой, приведенной в следующем примере:

```
$ sudo pkg_add xv-3.10ap4
xv-3.10ap4:jpeg-6bp3: complete
xv-3.10ap4:png-1.2.14p0: complete
xv-3.10ap4:tiff-3.8.2p0: complete
Can't install xv-3.10ap4: lib not found X11.9.0
Even by looking in the dependency tree:
    tiff-3.8.2p0, jpeg-6bp3, png-1.2.14p0
```

Maybe it's in a dependent package, but not tagged with @lib ?

(check with `pkg_info -K -L`)

If you are still running 3.6 packages, update them.

Установка xv была прервана `pkg_add(1)` во время установки зависимостей. Сработал еще один из механизмов системы безопасности, доступный с версии OpenBSD 3.7. Распаковочная информация содержит так же информацию о разделяемых библиотеках, если одна из таких библиотек не будет найдена, пакет не будет установлен.

Для разрешения конфликтов такого типа необходимо найти и установить требуемые библиотеки в вашу систему. Что можно проверить:

У вас может быть установлена старая версия пакета: присутствует старая версия необходимой библиотеки. Если это так, то обновите пакет.

Ваша система не полна: вы не установили файл, содержащий данную библиотеку. Просто добавьте необходимый файл.

Возможно ваша система устарела: у вас старая версия необходимой библиотеки. Загрузитесь с загрузочного диска (детально об этом в FAQ 4) и выберите пункт (U)upgrade

## 15.2.5 - Просмотр установленных пакетов

Вы можете увидеть список установленных пакетов, используя утилиту `pkg_info(1)`.

```
$ pkg_info
aterm-0.4.2p1      color vt102 terminal emulator with transparency support
bzip2-1.0.4        block-sorting file compressor, unencumbered
expat-2.0.0        XML 1.0 parser written in C
fluxbox-0.9.15.1p0 window manager based on the original Blackbox code
gettext-0.14.6     GNU gettext
imlib2-1.3.0       image manipulation library
jpeg-6bp3          IJG's JPEG compression utilities
libiconv-1.9.2p3   character set conversion library
libltdl-1.5.22p1   GNU libtool system independent dlopen wrapper
libungif-4.1.4p0   tools and library routines for working with GIF images
libutf8-0.8        provides UTF-8 locale support
mutt-1.4.2.2i      tty-based e-mail client
pcre-6.4p1         perl-compatible regular expression library
png-1.2.14p0       library for manipulating PNG images
screen-4.0.3p0     multi-screen window manager
tcsh-6.14.00p1     extended C-shell with many useful features
tiff-3.8.2p0       tools and library routines for working with TIFF images
tin-1.8.2p0        threaded NNTP and spool based UseNet newsreader
```

Когда задано имя пакета (или его расположение), `pkg_info(1)` отобразит более подробную информацию, характерную для данного пакета.

## 15.2.6 - Обновление установленных пакетов

Начиная с версии OpenBSD 3.7, стало возможно обновление уже установленных пакетов, используя флаг **-r** (= replace) утилиты `pkg_add(1)`. OpenBSD 3.8 содержит флаг **-u** для `pkg_add(1)`, который был добавлен в более истинный механизм обновления в 3.9.

Допустим вы имели старую версию пакета `unzip` перед обновлением OpenBSD 4.2 до 4.3. Теперь вы можете легко обновить его более новым пакетом из 4.3 следующим образом:

```
$ sudo pkg_add -u unzip
unzip-5.52 (extracting): complete
unzip-5.51 (deleting): complete
unzip-5.52 (installing): complete
Clean shared items: complete
```

Использование `pkg_add(1)` с флагом **-u** и без указания имени пакета приводит к проверке всех установленных пакетов на предмет обновления. Если пакет имеет зависимости, то они так же будут проверены для обновления.

**Замечание:** Флаг **-u** использует переменную окружения `PKG_PATH`. Если она не установлена, то `pkg_add(1)` не сможет найти обновления.

Начиная с OpenBSD 4.2 имеется несколько записей в `PKG_PATH`. Это не означает, что для обновления будут использованы они все. Напротив, `pkg_add(1)` остановится на первом найденном значении.

Если в системе имеются модифицированные конфигурационные файлы старой версии пакета, то они не будут затронуты при обновлении. Однако, вы можете заменить их файлами по умолчанию, передав `pkg_add(1)` флаг **-c**.

## 15.2.7 - Удаление установленных пакетов

Для удаления пакета просто возьмите имя пакета, показываемое `pkg_info(1)` (смотрите Просмотр установленных пакетов выше) и используйте `pkg_delete(1)`. В примере ниже был удален пакет `screen`. Запомните, что в некоторых случаях есть файлы, которые нужно удалить, но которые не удалил `pkg_delete(1)`. Так же как и в утилите `pkg_add(1)`, вы можете использовать флаг **-v** для получения более подробной информации.

```
$ sudo pkg_delete screen
screen-4.0.3p0: complete
Clean shared items: complete
```

**Замечание:** Чаще всего необязательно указывать версию или флavor в имени пакета, так как `pkg_delete(1)` обычно в состоянии найти полное имя пакета. Необходимо указывать полное имя пакета (к примеру `screen-4.0.23p0`) только если возможна двусмысленность из-за множества установленных пакетов с таким же названием. В этом случае `pkg_delete(1)` не сможет узнать какой пакет удалить.

К сведению, `pkg_delete(1)` не удалит конфигурационные файлы, если они были модифицированы. Об этом вам будет сообщено как указано ниже:

```
$ sudo pkg_delete screen
screen-4.0.3p0: complete
Clean shared items: complete
--- screen-4.0.3p0 -----
You should also remove /etc/screenrc (which was modified)
```

Если вы желаете удалить такие конфигурационные файлы автоматически, передайте `pkg_add(1)` флаг `-c`.

## 15.2.8 - Некорректная установка или удаление пакета

В очень редких случаях вы можете обнаружить, что пакет не был полностью добавлен или удален из-за конфликта с другими файлами. Неполная установка обычно помечается добавлением "partial-" в название пакета. Это может произойти, например, если вы нажмете случайно CTRL+C в процессе установки:

```
$ sudo pkg_add screen-4.0.3p0
screen-4.0.3p0: complete
Adjusting md5 for /usr/local/info/screen.info-3 from \
49fb3fe1cc3a3b0057518459811b6dac to 3b9c7811244fb9f8d83bb27d3a0f60d8
/usr/sbin/pkg_add: Installation of screen-4.0.3p0 failed , \
partial installation recorded as partial-screen-4.0.3p0
```

Хорошей идеей является удаление "частичных" пакетов из системы и устранение потенциальных проблем, приводящих к ошибке. Это признак того, что у вас нет чистой системы с софтом, установленным из пакетов, возможно пакеты смешались с другим софтом, установленным из исходников напрямую.

## 15.3 - Работа с портами

Как говорилось в введении, пакеты собираются из дерева портов. В этой секции мы поговорим как работает дерево портов, когда вы должны его использовать и как вы должны его использовать.

**ВАЖНОЕ ЗАМЕЧАНИЕ:** Использование дерева портов педназначается для продвинутых пользователей. **Для всех рекомендуется использование бинарных пакетов.** НЕ задавайте детских вопросов в рассылке, подобных "Как я могу сделать дерево портов готовым к работе?". Если у вас есть вопрос по дереву портов, то это предполагает, что вы прочитали страницы map и этот FAQ, и вы готовы к работе с ними.

### 15.3.1 - Как это работает?

Дерево портов, идея, первоначально заимствованная из [FreeBSD](#), содержит Makefiles для управления каждым сторонним приложением, в том числе следующую информацию

где и как скачать исходники программного обеспечения,  
какие зависимости он имеет,



какие есть альтернативные источники (если необходимо),  
как его сконфигурировать и собрать,  
как его проверить (опционально),  
как его установить.

Кроме Makefile, каждый порт также содержит по крайней мере следующее:

PLIST или упаковывающий список, который содержит инструкции для создания пакета собранного приложения,  
DESCR или описание приложения,  
distfile, содержит контрольные суммы и размер.

Вся эта информация хранится иерархически в директории /usr/ports. Эта иерархия включает три основных поддиректории:

distfiles/ - где система портов хранит ПО после скачивания.

infrastructure/ - главная директория портовой инфраструктуры, содержащая все необходимые скрипты и мейкфайлы.

packages/ - содержит все бинарные пакеты, собранные системой портов.

Другие поддиректории различных категорий приложений содержат поддиректории с непосредственно самими портами. Комплекс портов может быть организован на более глубоком уровне, например, если они имеют основную часть и ряд расширений или stable и snapshot версии приложения. Каждая директория порта должна содержать поддиректорию pkg/, хранящую упаковывающий список и описание приложения. Также могут быть поддиректории patches/ и files/ для патчей и дополнительных файлов соответственно.

Когда пользователь запускает [make\(1\)](#) в поддиректории какого-то порта, система рекурсивно проходит по дереву зависимостей, проверяет установленные зависимости, собирает и устанавливает недостающие зависимости и затем продолжает построение желаемого порта. Все построения производятся в *рабочей директории*, которую создает порт. This is either a subdirectory of the port's main directory, in which case it is recognized by its prefix "w-", или директории \$WRKOBJDIR, если установлена переменная WRKOBJDIR (смотри Конфигурирование системы портов).

**Замечание:** порты никогда не устанавливаются непосредственно в вашу систему! Для их установки используется *fake installation directory*. Всё что установится в него, собирается вместе в пакет (который сохраняется в подкаталоге packages/ дерева портов, как упоминалось ранее). Установка порта в действительности означает: создание пакета и его установку!

**Подробная информация** о системе портов может быть найдена на страницах [map](#):

[ports\(7\)](#) - описывает различные стадии (цели make) установки порта, использование флаворс, субпакетов и других опций.

[bsd.port.mk\(5\)](#) - подробная информация о всех целях, переменных make, fake installation directory framework, и т.д.

## 15.3.2 - Получение дерева портов

Перед тем как продолжить, вы должны ознакомиться с секцией о НЕ смешивании версий вашей системы OpenBSD и дерева портов. После принятия вами решения об использовании того или иного дерева портов, вы можете получить дерево портов из различных источников. В ниже приведенной таблице обзор, где вы можете получить flavors, и в какой форме. Знак 'x' обозначает доступность, а '-' невозможность получения данным способом.

Источник	Форма	Flavor			
		-release	-stable	snapshots	-current
CD-ROM	.tar.gz	x	-	-	-
FTP зеркала	.tar.gz	x	-	x	-
AnonCVS	cvs checkout	x	x	-	x

На [CD-ROM](#) или [зеркале FTP](#), найдите файл `ports.tar.gz`. Вы должны его распаковать `untar` в `/usr`, что приведет к созданию `/usr/ports` и вложенной структуры. К примеру:

```
$ cd /tmp
$ ftp ftp://ftp.openbsd.org/pub/OpenBSD/4.3/ports.tar.gz
$ cd /usr
$ sudo tar xzf /tmp/ports.tar.gz
```

Snapshot-ы доступны на зеркалах FTP и создаются ежедневно `-current` деревья портов. Также вы можете найти `snapshots` в `/pub/OpenBSD/snapshots/`. Если вы используете `snapshot` дерева портов, вы должны использовать `snapshot` OpenBSD. Сохраняйте версии OpenBSD и портов синхронизированы!

Для дополнительной информации о получении дерева портов с использованием `AnonCVS` читайте страницу о [AnonCVS](#), где содержатся также информация о доступных серверах и примеры использования.

### 15.3.3 - Конфигурирование системы портов

**ВНИМАНИЕ:** В этом разделе рассказывается о некоторых глобальных установках для сборки приложений из портов. Вы можете пропустить этот раздел, но тогда вам придется выполнять многие из вызовов `make(1)`, приведенные в примерах инструкции, от суперпользователя `root`.

Поскольку проект OpenBSD не имеет ресурсов для полного аудита кодов всех программ, содержащихся в дереве портов, вы можете сконфигурировать систему портов для принятия дополнительных мер безопасности. Инфраструктура портов может быть построена от обычного пользователя, и только несколько шагов потребуют привелегий суперпользователя `root`. Например `fake` и `install` `make` `targets`. Однако, поскольку права `root` все равно потребуются на определенном этапе, система портов не сможет защитить вас при сборке вредоносного приложения.

Вы можете настроить [sudo\(8\)](#) и использовать в системе портов ее при необходимости. Добавьте в `/etc/mk.conf` строку следующего содержания

```
SUDO=/usr/bin/sudo
```

Вы можете сменить владельца дерева портов на обычного пользователя. В этом случае владелец дерева портов должен быть добавлен в группу `wsgc` и необходимые каталоги должны быть доступны для записи.

```
# chgrp -R wsrc /usr/ports
# find /usr/ports -type d -exec chmod g+w {} \;
```

вы можете использовать [systrace\(1\)](#) в системе портов, добавив в `/etc/mk.conf`

```
USE_SYSTRACE=Yes
```

Это предписывает процедуре сборки находиться в разрешенных каталогах и предотвращает запись в недопустимых местах, таким образом риск повреждения системы значительно снижается. Обратите внимание, использование `systrace(1)` добавит ко времени на сборку примерно 20%.

Также возможно использование доступного только для чтения дерева портов, разрешив запись только в указанные вами необходимые для сборки каталоги:

Рабочий каталог порта. Задается переменной `WRKOBJDIR`, где указаны необходимые для работы каталоги.  
Содержащий дистрибутив каталог. Задается переменной `DISTDIR`.  
Каталог для размещения собранных бинарных пакетов. Задается переменной `PKGREPOSITORYBASE`.

Для примера, добавим следующие строки в `/etc/mk.conf`

```
WRKOBJDIR=/usr/obj/ports
DISTDIR=/usr/distfiles
PKGREPOSITORYBASE=/usr/packages
```

Если необходимо, вы можете также сменить владельца этих каталогов на локального пользователя и группу так, чтобы система портов могла создать директории от обычного пользователя.

### 15.3.4 - Поиск в дереве портов

Как только вы установили дерево портов на вашу систему, поиск программ становится простым. Используйте `make search key="searchkey"`, как показано в приведенном примере.

```
$ cd /usr/ports
$ make search key=rsnapshot
Port:    rsnapshot-1.2.9
Path:    net/rsnapshot
Info:    remote filesystem snapshot utility
Maint:   Sigfred Haversen
Index:   net
L-deps:
B-deps:  :net/rsync
R-deps:  :net/rsync
Archs:   any
```

Результат поиска будет содержать краткое описание каждого найденного приложения: имя порта, путь к порту,

краткому описанию, ментейнера порта, ключи, зависимости library/build/runtime, архитектуры, на которых порт работает.

Этот механизм, однако, чрезвычайно прост, он использует для работы с индексным файлом портов awk(1). Начиная с OpenBSD 4.0 создан новый механизм, порт "sqlports", позволяющий осуществлять точный поиск с помощью SQL. Это база данных SQLite, но изначально база данных о инфраструктуре портов должна быть создана. Порт sqlports включает в свой состав скрипт для генерации базы данных, который может быть взят за основу при создании баз данных в разных форматах.

Просто используйте pkg\_add(1) для пакета sqlports, и, в этом случае, запустится sqlite3. Пример сессии выглядит примерно так:

```
$ sqlite3 /usr/local/share/sqlports
SQLite version 3.3.12
Enter ".help" for instructions
sqlite> SELECT FULLPKGNAME,COMMENT FROM Ports WHERE COMMENT LIKE '%statistics%';
Guppi-0.40.3p1|GNOME-based plot program with statistics capabilities
mailgraph-1.12|a RRDtool frontend for Postfix statistics
R-2.4.1|clone of S, a powerful math/statistics/graphics language
py-probstat-0.912p0|probability and statistics utilities for Python
darkstat-3.0.540p1|network statistics gatherer with graphs
pfstat-2.2p0|packet filter statistics visualization
tcpstat-1.4|report network interface statistics
wmwave-0.4p2|Window Maker dockapp to display wavelan statistics
diffstat-1.43p0|accumulates and displays statistics from a diff file
sqlite>
```

Приведен пример, конечно, очень простого поиска. SQL позволяет гораздо большее, включая зависимости, конфигурационные опции, используемые библиотеки и так далее.

### 15.3.5 - Инсталляция с удовлетворением зависимостей: простой пример

Для простоты понимания пример: rsnapshot. Приложение имеет зависимость: rsync.

```
$ cd /usr/ports/net/rsnapshot
$ make install
==> Checking files for rsnapshot-1.2.9
>> rsnapshot-1.2.9.tar.gz doesn't seem to exist on this system.
>> Fetch http://www.rsnapshot.org/downloads/rsnapshot-1.2.9.tar.gz.
100% |*****| 173 KB 00:02
>> Size matches for /usr/ports/distfiles/rsnapshot-1.2.9.tar.gz
>> Checksum OK for rsnapshot-1.2.9.tar.gz. (sha1)
==> rsnapshot-1.2.9 depends on: rsync-2.6.9 - not found
==> Verifying install for rsync-2.6.9 in net/rsync
==> Checking files for rsync-2.6.9
>> rsync-2.6.9.tar.gz doesn't seem to exist on this system.
>> Fetch ftp://ftp.samba.org/pub/rsync/old-versions/rsync-2.6.9.tar.gz.
```

```

100% |*****| 792 KB 00:31
>> Size matches for /usr/ports/distfiles/rsync-2.6.9.tar.gz
>> Checksum OK for rsync-2.6.9.tar.gz. (sha1)
==> Verifying specs: c
==> found c.40.3
==> Extracting for rsync-2.6.9
==> Patching for rsync-2.6.9
==> Configuring for rsync-2.6.9
    [...snip...]
==> Building for rsync-2.6.9
    [...snip...]
==> Faking installation for rsync-2.6.9
    [...snip...]
==> Building package for rsync-2.6.9
Link to /usr/ports/packages/i386/ftp/rsync-2.6.9.tgz
Link to /usr/ports/packages/i386/cdrom/rsync-2.6.9.tgz
==> Installing rsync-2.6.9 from /usr/ports/packages/i386/all/rsync-2.6.9.tgz
rsync-2.6.9: complete
==> Returning to build of rsnapshot-1.2.9
==> rsnapshot-1.2.9 depends on: rsync-2.6.9 - found
==> Extracting for rsnapshot-1.2.9
==> Patching for rsnapshot-1.2.9
==> Configuring for rsnapshot-1.2.9
    [...snip...]
==> Building for rsnapshot-1.2.9
    [...snip...]
==> Faking installation for rsnapshot-1.2.9
    [...snip...]
==> Building package for rsnapshot-1.2.9
Link to /usr/ports/packages/i386/ftp/rsnapshot-1.2.9.tgz
Link to /usr/ports/packages/i386/cdrom/rsnapshot-1.2.9.tgz
==> rsnapshot-1.2.9 depends on: rsync-2.6.9 - found
==> Installing rsnapshot-1.2.9 from /usr/ports/packages/i386/all/rsnapshot-1.2.9.tgz
rsnapshot-1.2.9: complete

```

Как мы видим, система портов многие вещи делает автоматически. Это получение, распаковка, наложение патчей на исходный код, конфигурирование и сборка (компиляция) из исходников, установка в заданное место, сборка пакета (регистрация в списке пакетов) и установка готового пакета в систему (обычно в /usr/local/). Все это делается рекурсивно **для всех зависимостей** порта. Обратите внимание на "==> Verifying install for ..." и "==> Returning to build of ..." в выводимых сообщениях, указывающих на прохождение по дереву портов.

Если предыдущая версия устанавливаемого приложения уже установлена в систему вы должны использовать make update вместо make install. Это вызовет исполнение pkg\_add(1) с ключом **-r**.

**Внимание:** Большие приложения могут потребовать значительных системных ресурсов. Хороший пример - компиляторы вроде GCC 4.0 или Java 2 Software Development Kit. Если вы получили сообщение об ошибке "out of memory" при сборке порта, это скорее всего вызвано одной из двух причин:

Ресурсы вашей системы сильно ограничены. Настройте их с помощью ulimit (в ksh) или limit (в csh). Если это не помогает, войдите в систему как root перед сборкой или используйте [sudo\(8\)](#) с ключом -c для сборки с заданными ограничениями ресурсов для class (см. [login.conf\(5\)](#) для получения информации о login classes).

вашей системе просто не хватает ОЗУ.

### 15.3.6 - Очистка после сборки

Вероятно, Вы захотите очистить рабочий каталог порта после формирования пакета и его установки.

```
$ make clean
===> Cleaning for rsnapshot-1.2.9
```

Дополнительно, вы также можете очистить рабочие каталоги всех зависимостей:

```
$ make clean=depends
===> Cleaning for rsync-2.6.9
===> Cleaning for rsnapshot-1.2.9
```

если вы желаете очистить исходные коды дистрибутивов использованного порта

```
$ make clean=dist
===> Cleaning for rsnapshot-1.2.9
===> Dist cleaning for rsnapshot-1.2.9
```

Если вы собирали различные flavor порта, вы можете очистить рабочие каталоги всех flavor

```
$ make clean=flavors
```

You can also clean things up as they get built, by setting a special variable. Work directories will automatically be cleaned after packages have been created:

```
$ make package BULK=Yes
```

### 15.3.7 - Деинсталляция пакетов, созданных портами

Очень просто удалить порт:

```
$ make uninstall
===> Deinstalling for rsnapshot-1.2.9
rsnapshot-1.2.9: complete
Clean shared items: complete
```

Это вызовет исполнение `pkg_delete(1)` для удаления соответствующего приложения из системы. При необходимости вы можете деинсталлировать порт и повторно его установить используя

```
$ make reinstall
==> Cleaning for rsnapshot-1.2.9
/usr/sbin/pkg_delete rsnapshot-1.2.9
rsnapshot-1.2.9: complete
Clean shared items: complete
==> Installing rsnapshot-1.2.9 from /usr/ports/packages/i386/all/rsnapshot-1.2.9.tgz
rsnapshot-1.2.9: complete
```

If you would like to get rid of the packages you just built, you can do so as follows:

```
$ make clean=packages
==> Cleaning for rsnapshot-1.2.9
rm -f /usr/ports/packages/i386/all/rsnapshot-1.2.9.tgz
```

### 15.3.8 - Использование flavors и subpackages

Пожалуйста, ознакомьтесь со страницей руководства (`man ports(7)`), которое дает хороший обзор данной темы. Имеется два различных механизма управления пакетами приложений в зависимости от потребностей.

Первый механизм - **flavors**. Flavor обычно указывает на опции при компиляции приложения. К примеру, некоторые приложения имеют "no\_x11" flavor, указывающий, что приложение будет использоваться без X. Некоторые оболочки имеют "static" flavor, что указывает на сборку статической слинкованной версии. Некоторые порты могут быть собраны с использованием различных средств разработки графических приложений. Другие примеры: работа с разными базами данных, различными сетевыми опциями (SSL, IPv6, ...), различными размерами страниц и т.д.

**Резюме:** наиболее вероятно, что вы будете использовать flavors когда нет пакета с необходимым для вас flavor. В этом случае вы зададите желаемый flavor и соберете порт самостоятельно.

Каждый flavor порта будет иметь свой рабочий каталог для сборки и каждый собранный пакет будет иметь свое соответствующее название для избежания бардака. Чтобы просмотреть различные flavors порта, смените каталог и выполните команду

```
$ make show=FLAVORS
```

Также вы должны не забывать про файл `DESCR`, где описываются имеющиеся flavors.

Второй механизм называется подпакеты (**subpackages**). Собирающий порт может решить создать subpackages для логически разделенных различных составляющих того же самого приложения. Часто можно наблюдать подпакеты (subpackages) для серверной и клиентской частей приложения. Иногда значительная по размеру документация выделяется в отдельный подпакет для экономии дискового пространства. Дополнительная функциональность,

повлекшая большое количество зависимостей часто делается отдельным пакетом. Разработчик порта также должен принять решение, какие subpackages должны входить в главный, устанавливаемый по умолчанию subpackage. Другие примеры: наборы тестовых программ, различные модули для поддержки разных опций и т.д.

**Резюме:** Некоторые порты разделены на несколько различных packages. make install повлечет только установку основного subpackage.

Вывести список packages port

```
$ make show=PACKAGES
```

make install установит только основной subpackage. Для установки всех

```
$ make install-all
```

Показать список различных subpackages, доступных для порта

```
$ make show=MULTI_PACKAGES
```

Возможно выбрать, какие subpackage(s) установить для порта. После некоторых проверок, будет вызвана pkg\_add(1) для установки желаемых subpackage(s).

```
$ env SUBPACKAGE="-server" make install
```

Примечание: механизм subpackages работает с пакетами, не изменяя конфигурационные параметры перед сборкой порта. Для их изменения используйте flavors.

## 15.4 - Часто задаваемые вопросы

### 15.4.1 - Я получаю целую кучу разнообразных и непонятных ошибок при сборке порта. Я не могу получить набор портов, работающих везде.

Скорее всего вы используете не синхронизованную систему и дерево портов.

*То есть?*

Прочитайте BCE о Flavor`ax OpenBSD: -release, -stable, and -current. Пожалуйста, прочтите документ указанный выше, чтобы получить основную идею того, что вы хотите использовать. Вкратце:

**Release:** То, что распространяется на CD.

**Stable:** Release с патчами безопасности.

**Current:** Разрабатываемая версия OpenBSD.



НЕ создавайте -current дерево портов, полагая, что оно будет работать на -release или -stable системе. Это одна из наиболее распространенных ошибок и вы разозлите людей, когда спросите почему "ничего не работает!" Если **вы следуете -current, то у вас должна быть и -current система, и -current дерево портов.** Да, это значит, что чудесный новый порт не заработает на вашей "старой" системе -- даже если она была -current всего несколько недель назад. Помните, что если вы используете X11 как часть системы, то также должны следовать соответствующей ветке!

Поскольку никаких важных изменений в -stable не делается, то можно использовать -stable дерево портов на -release системе и наоборот. Не обязательно обновлять все установленные пакеты после наложения нескольких патчей на вашу систему.

Еще одна частая проблема - отсутствие установленной X11. Даже если собираемый вами порт не имеет прямых зависимостей с X11, его subpackage или их зависимости могут требовать заголовки и библиотеки X11. Собранные порты не поддерживаются на системах без X11, поэтому вы должны принимать это во внимание. Тем не менее, есть много портов с flavor "no\_x11", которые не требуют установки X11 на вашей системы.

В OpenBSD 4.2 многие пакеты, требовавшие libxpat, влекли за собой необходимость установки xbase42.tgz, хотя они не имели графической функциональности. Это исправлено в OpenBSD 4.3, теперь libxpat включена в состав base43.tgz. Теперь libxpat больше не требует установки наборов установочных файлов X file sets. Подробности об этом в [Upgrade Guide](#).

## Последняя версия моего любимого программного обеспечения недоступна!

Если вы используете release или stable версию OpenBSD, то никогда не найдете обновления пакета до следующего release, или пока не найдется брешь в системе безопасности, оправдывающая обновление порта и соответствующего пакета в -stable ветке.

### **ВНИМАНИЕ: НЕ ПУТАЙТЕ версии портов и OpenBSD!**

Делая так, рано или поздно (скорее всего рано) вы погрязнете в самых разнообразных ошибках системы.

*Но, гм, у меня же -current!*

Коллекция портов поддерживается добровольцами. Иногда проект просто не имеет ресурсов разработчиков, чтобы поддерживать актуальность порта. Разработчики зачастую собирают те пакеты, которые считают нужными и имеют возможность испытать на имеющемся оборудовании. Ваши [пожертвования](#) помогают увеличить количество платформ и портов.

Некоторые отдельные порты могут быть не так актуальны в связи с этим. Коллекция портов может содержать январьскую версию программы в то время как новый релиз программы выпущен разработчиком в мае, три месяца назад. Часто это мотивированное решение; новая версия может иметь нерешенные на данный момент мантейнером проблемы работы на OpenBSD, или приложение работает хуже, чем более старая версия: OpenBSD может иметь [цели](#) отличные от целей разработчиков других проектов, которые приводят к результату, неприемлимому с точки зрения разработчиков OpenBSD. Обновление также может быть отложено, исходя из соображений, что в новую версию не было внесено принципиальных изменений по сравнению с предыдущей версией.

Если вы действительно нуждаетесь в последней версии данной программы, то можно обратиться к мантейнеру с соответствующей просьбой (ниже указано, как посмотреть, кто является мантейнером). Будет еще лучше, если вы

ему поможете в этой работе.

### 15.4.3 - Почему нет пакета с моей любимой программой?

Этому может быть несколько причин:

На компакт-дисках OpenBSD недостаточно места, чтобы включить каждый существующий пакет для каждой существующей платформы. Поэтому на CD включены только наиболее часто используемые пакеты. К тому же, некоторое программное обеспечение может распространяться только бесплатно, то есть не может быть включено на CD. Если вы не можете найти пакет на компакт-дисках, то попробуйте поискать в другом источнике, например, на зеркале FTP.

Некоторое программное обеспечение просто не может распространяться в бинарном виде из-за своей лицензии. Другое — например из-за патентного ограничения. Если ваше Самое-Любимое-ПО подпадает под эту категорию, то вам нужно будет использовать порт и собирать пакет из исходников.

Очевидная, но, иногда забываемая причина: для вашего Самого-Любимого-ПО просто не существует порта. Вы можете это проверить, изучив дерево портов. Если там действительно его нет, то ваша помощь в этом приветствуется.

### 15.4.4 - Почему отсутствует порт моей любимой программы?

Коллекция портов поддерживается добровольцами. Процесс разработки портов ограничен количеством людей, принимающих участие в проекте, и их свободным временем. Люди обычно создают новые порты только для программ, которые они используют или которые непосредственно их интересуют.

Однако вы можете изменить эту ситуацию. Создайте свой собственный порт. Существует документация по этому вопросу: [Создание порта OpenBSD](#). Прочтите ее, а потом.. прочтите снова. Особенно ту часть, где описывается процесс сопровождения вашего порта. После этого, попробуйте создать новый порт и тщательно его проверьте, шаг за шагом. Если, наконец-то, он работает на вашей системе без ошибок, то отошлите его в почтовую рассылку [ports@openbsd.org](mailto:ports@openbsd.org). Существует хороший шанс, что вы получите некоторые отзывы и результаты тестов от других людей. Если эти результаты положительны, то ваш порт будет включён в общее дерево портов.

### 15.4.5 - Почему моя Самая Любимая Программа не входит в базовую систему?

Потому что OpenBSD изначально создавалась как небольшая самостоятельная UNIX`оподобная операционная система, и существует определенная грань - что должно входить в ее состав, а что нет. Чтобы приложение было включено в базовый набор, необходимо соблюдение следующих условий:

ПО должно соответствовать общепринятым стандартам, которые соответствуют [целям](#) проекта OpenBSD.

Лицензия данного приложения не будет чересчур ограничивать его использование, и она должна быть совместимой с лицензией BSD.

Приложение не должно быть слишком большим, дабы обеспечить компактность базовой системы.

Более подробные ответы на этот вопрос можно также найти в [FAQ 1](#).

## 15.4.6 - Что я должен использовать: пакеты или порты?

Вообще, мы **настоятельно советуем** для установки приложений использовать пакеты, а не порты. Команда разработчиков портов OpenBSD полагает, что конечным результатом их работы будут пакеты, а не непосредственное применение портов.

Сборка приложения из исходников не является простой задачей. Должно быть скомпилировано не только само приложение, но и используемые им инструментальные средства и зависимости. К сожалению, совместно работающие OpenBSD, инструменты, приложения развиваются и часто возникает следующая ситуация. В один момент изменения, сделанные в той или иной части, от которой зависит данный порт, могут привести к тому, что порт окажется нерабочим. Каждые шесть месяцев, с выходом каждого релиза OpenBSD, делается проверка на сборку каждого порта на каждой платформе, но в процессе развития системы, вероятно, порты могут оказаться нерабочими.

В дополнение к наличию проблемы работы зависимых составляющих присутствует и тот факт, что на сборку из исходников тратятся время и ресурсы. Для примера, [CVSup](#), инструмент для [отслеживания исходного дерева OpenBSD](#). Для установки CVSup на средней платформе с быстрым каналом в интернет в случае использования пакета потребуется примерно десять секунд -- это время на загрузку и распаковку 779kB-ного пакета. Для сравнения, сборка CVSup на той же машине из исходников длительная и серьезная задача, требующая использования многих инструментов работы компилятора, на ее выполнение потребуется как минимум полчаса. Другие приложения, как [Mozilla](#) или [KDE](#) потребуют часы работы и очень большое количество дискового пространства, ОЗУ и swap. Зачем тратить собственное время и работу, когда приложения уже откомпилированы, готовы к использованию и находятся на CD-ROM или FTP-зеркале?

Конечно, есть несколько серьезных оснований использовать порты:

- Лицензия не позволяет включить в состав пакетов OpenBSD.

- Вы желаете модифицировать или отладить приложение, либо изучить его исходный код.

- Вам требуется flavor или порт, не поддерживаемый разработчиками портов OpenBSD.

- Вы желаете изменить размещение файлов (то есть изменить PREFIX или SYSCONFDIR).

Однако, для большинства людей и большинства приложений, использование пакетов проще, поэтому мы настоятельно рекомендуем устанавливать приложения в OpenBSD используя пакеты.

## 15.4.7 - Как мне настроить порты для достижения максимальной производительности?

OpenBSD заботится о стабильности и безопасности. Поэтому, подобно ядру GENERIC, которое поставляется по умолчанию и является единственно поддерживаемым ядром, группа людей, занимающихся портами, всего лишь следит за тем чтобы порт был рабочим и стабильным. Если вы хотите использовать дополнительные опции компилятора при сборке, то это ваше право. Но, пожалуйста, не задавайте потом в почтовых рассылках вопросов о том, что то, или иное приложение не работает, когда вы пытаетесь включить какие-то скрытые опции для ускорения его работы. По большому счёту, весь этот тюнинг отнюдь не является необходимым для более чем 99% пользователей, и скорее всего это будет пустой тратой времени для вас, как для пользователя, равно как и для разработчиков, которые прочтут о ваших "проблемах", в то время как, в действительности, их могло бы и не быть.

## 15.4.8 - Я отослал новый порт/обновление несколько недель назад. Почему он до сих пор еще не принят?

Группа, занимающаяся портами, очень малочисленна, и это значит, что в течение того времени ни один коммитер еще не успел просмотреть ваш порт/обновление. Это может быть слегка обидно, но лучше просто проигнорируйте данный факт. Заботьтесь о своём порте, шлите обновления, и в конечном счёте кто-то может быть позаботится о нём. Ранее так случалось, что у коммитеров вдруг появляется свободное время, которое они могут посвятить приёму портов, или сфера их интересов охватывает новые области деятельности, и ваша старая отсылка становится им интересна (если она запомнилась).

## 15.5 - Сообщение о проблемах

Если вы имеете проблемы с существующим портом, пожалуйста, сообщите по e-mail мантейнеру порта. Просмотреть, кто является мантейнером, можно введя нижеприведенную команду. Пример:

```
$ cd /usr/ports/archivers/unzip
$ make show=MAINTAINER
```

в противном случае, если мантейнера нет, либо вы не смогли с ним связаться, отправьте e-mail в почтовую рассылку OpenBSD ports, [ports@openbsd.org](mailto:ports@openbsd.org). Пожалуйста, НЕ используйте рассылку [misc@openbsd.org](mailto:misc@openbsd.org) для сообщений о портах.

В любом случае, пожалуйста, включите информацию:

вашу версию OpenBSD включая примененные патчи. Версию ядра, выполнив: `sysctl -n kern.version`  
версию дерева портов: если есть файл `/usr/ports/CVS/Tag`, отправьте его содержание. Если файл отсутствует, вы используете `-current` дерево портов.

исчерпывающее описание проблемы. Не волнуйтесь, если окажется длинным. Упомяните все предпринятые вами шаги до возникновения проблемы. Проблема повторяется? Чем более подробно вы проблему опишете, тем больше вероятность получения помощи.

Для портов, сборка из которых некорректна, полный вывод процесса сборки обязателен. Для этого можно использовать скрипт `postlogger`, его можно найти в `/usr/ports/infrastructure/build`. Пример выполнения `portslogger`:

```
$ mkdir ~/portslogs
$ cd /usr/ports/archivers/unzip
$ make clean install 2>&1 | /usr/ports/infrastructure/build/portslogger \
    ~/portslogs
```

После этих действий в `~/portslogs` директории вы получите logfile, который надо отправить мантейнеру порта. Удостоверьтесь также, не использовалось ли при сборке особых настроек, например, в `/etc/mk.conf`.

Как вариант, можно

Использовать [script\(1\)](#) для создания описания сборки. Не удаляйте конфигурационную информация. присоедините вывод [pkg\\_info\(1\)](#) if it seems even remotely relevant. ошибки встроенного компилятора [gcc\(1\)](#) следует отправлять в рассылку gcc mailinglist. It does save time if you follow their direction, и предоставите, как минимум, различные файлы, созданные командой gcc -save-temps.

## 15.6 - Как помочь нам

Существует много способов нам помочь. Они приведены ниже, в порядке возрастания сложности.

Сообщить нам о испытываемых вами проблемах.

Периодически проверять порты, сообщать о проблемах и предлагать улучшения. Только перед этим ознакомьтесь с [руководством по проверке портов](#).

Проверять появляющиеся в рассылке обновления портов.

Прислать изменения или патчи мантейнеру, или, если порт не имеет мантейнера - в почтовую рассылку. Будет замечательно, если ваши патчи не приведут к пустой трате времени мантейнера, а, наоборот, помогут ему сэкономить свое время.

Создать новый порт. Если вы действительно хотите знать все о [портировании приложений на OpenBSD](#), то создание порта — это отличная отправная точка.

**Внимание:** Для всех работ по созданию и проверке обновлений вы **должны работать в версии -current!** В общем-то, это нежелательно, потому что окружение такой системы постоянно изменяется, поэтому продолжайте в ней работать только если вы действительно уверены в том, что будете вносить изменения в -current.

Перевод соответствует \$OpenBSD: faq15.html,v 1.50 2008/05/01 20:54:49 steven Exp \$

---