

Эффективная работа с портами во FreeBSD

© Сергей А. Осокин, 2002, 2003
<http://ozz.pp.ru/>

1 октября 2003 г.

Порт — это не только то место, куда приходят корабли. В мире компьютеров есть коммуникационные порты, а в мире FreeBSD есть коллекция портов программ различного назначения.

1 Введение

Глава четвёртая руководства по FreeBSD описывает порт следующим образом: это «скелет», состоящий из набора файлов, который указывает вашей операционной системе где взять исходный набор файлов для сборки программы, как собрать и установить эту программу, а также каким образом эту программу можно будет впоследствии безболезненно удалить из системы.

2 Установка коллекции портов

Установить коллекцию портов (или как ещё говорят «порты») можно как в процессе установки самой операционной системы, так и позже. Для установки портов можно воспользоваться утилитой `/stand/sysinstall`, альтернативный вариант — установить порты вручную. Первый вариант достаточно подробно описан в руководстве по FreeBSD, а второй вариант мы рассмотрим ниже.

Итак, мы имеем установленную операционную систему FreeBSD. В этой системе отсутствует коллекция портов, о чём нам говорит пустой каталог `/usr/ports` (возможно отсутствие этого каталога). На сайте проекта FreeBSD портам посвящён отдельный раздел¹. На этой странице можно найти ссылку на архив² текущего (т.е. состояние на сегодня) дерева портов.

Скачаем этот архив³ (ваш компьютер должен быть подключён к Сети), поместим его содержимое в каталог `/usr` (требуется около 120МБ дискового пространства для установки полной коллекции портов).

```
# fetch -o - ftp://ftp.FreeBSD.org/pub/FreeBSD/branches/-current/ports.tar.gz |
tar xfvz - -C /usr
```

3 Работаем с портами

Для работы с нужным вам портом достаточно перейти в соответствующий каталог

```
$ cd /usr/ports/раздел/порт
```

и указать для сборки команду

```
$ make
```

или `make build`.

Для сборки и установки

```
# make install
```

¹<http://www.freebsd.org/ports/>

²<ftp://ftp.FreeBSD.org/pub/FreeBSD/branches/-current/ports.tar.gz>

³На момент написания документа размер файла `ports.tar.gz` составлял 19МБ

Попробуем удалить установленное ПО

```
# make deinstall
```

и установим его ещё раз

```
# make reinstall
```

Для очистки каталога порта после установки

```
# make clean
```

Нам может понадобиться уже собранный нами порт, например, для установки на другую машину. В этом случае существует возможность создания пакета из уже собранного порта.

Следующая команда

```
# make package
```

создаст пакет, который можно будет использовать в другом месте.

Команда `make(1)` понимает несколько аргументов, поэтому более эффективной будет команда

```
# make install package clean
```

которая соберёт порт, создаст пакет и очистит скелет порта.

Пользователь может не знать как называется тот или иной порт в системе портов FreeBSD. Можно воспользоваться командой `make search` для поиска. Например, нам нужно найти порт проигрывателя компании Real, известный как Real Player.

```
$ make search key='Real Player'
```

Хм, пусто. Уменьшим длину аргумента `key` до «Real».

```
$ make search key='Real'
```

Эта команда выдаёт слишком много информации. Можно воспользоваться командой `more(1)`, но попробуем воспользуемся командой `grep(1)`.

```
$ make search key='Real' | grep -i Player
```

Ура! Получилось!

```
Port:   linux-realplayer-8.cs2
Path:   /usr/ports/multimedia/linux-realplayer
Info:   Linux RealPlayer 8.0 from RealNetworks
Info:   Multiplayer graphical arcade and adventure game made for X11
Info:   Server for multiplayer graphical arcade and adventure game for X11
Info:   Multi-player real-time strategy game for X11
Info:   Multi-player real-time strategy game for X11 (beta release)
```

Для установки программы остаётся только перейти в каталог `/usr/ports/multimedia/linux-realplayer` и выполнить команду `make install clean`.

Обратите внимание на вид полученной информации. Вся информация поделена на несколько групп (таких как `Port`, `Path`, `Info`). Более того, информация поделена на 2 колонки. В левой колонке указано свойство порта, в правой — значение. Имя порта имеет префикс `linux-`, что говорит нам о том, что эта программа работает под FreeBSD в режиме эмуляции ОС GNU/Linux и «собрана» именно под эту ОС. В коллекции портов есть и другие префиксы, такие как: `p5-`, `ruby-` и др.

Следует заметить, что здесь были перечислены наиболее часто встречающиеся «цели», такие как `clean`, `install`, `search`⁴.

⁴Более подробное описание этих и других целей можно найти в руководстве по портам `ports(7)`

4 Обновление коллекции портов

Обновление портов можно производить при помощи CVS/CTM или CVSup. Подробнее о CVSup можно почитать в руководстве `cvsup(8)`, на странице проекта CVSup⁵, в руководстве по FreeBSD.

Для использования CVSup следует установить соответствующий пакет. Попробуем воспользоваться указанными выше методами для поиска.

```
$ make search key='CVSup'
```

Много информации... Попробуем вместе с `grep`.

```
$ make search key='CVSup' | more
```

В дереве портов несколько пакетов CVSup? Конечно же, ответ на первый взгляд очевиден: скорее всего правильный путь к CVSup — `/usr/ports/net/cvsup`.

Найдём все «Path» к портам.

```
$ make search key='CVSup' | grep -i CVSup | grep Path
Path: /usr/ports/net/cvsup
Path: /usr/ports/net/cvsup-mirror
Path: /usr/ports/net/cvsup-without-gui
Path: /usr/ports/net/cvsupit
Path: /usr/ports/sysutils/fastest_cvsup
```

Получается, что в дереве портов есть, как минимум, 5 пакетов, связанных с работой CVSup. Какой из них нам нужен? Прочитаем однострочные комментарии к портам в файлах `pkg-comment`.

```
$ make search key='CVSup' | grep -i CVSup | grep Path |
  awk '{print $2"/pkg-comment";}' | xargs cat
General network file distribution system optimized for CVS (GUI version)
A kit for easily setting up a FreeBSD mirror site using CVSup
cat: /usr/ports/net/cvsup-without-gui/pkg-comment: No such file or directory
CVSup installation and setup front-end package (16.1 / GUI V3)
Finds fastest CVSup server
```

Вывод команды показывает на отсутствие файла `/usr/ports/net/cvsup-without-gui/pkg-comment`. Забавно, не правда ли?

Итак, правильный ответ — `/usr/ports/net/cvsup` (или как ещё говорят `net/cvsup`, `/usr/ports/` пропускается за ненадобностью), за одним небольшим «но». Это GUI, т.е. графическая версия пакета. Если вы собираетесь использовать CVSup на машине без графического интерфейса или хотите сэкономить время на сборку и установку CVSup — воспользуйтесь портом `net/cvsup-without-gui`.

Обратите внимание на этот порт. Он состоит из одного Makefile. А где же описание порта? В `net/cvsup-without-gui` сказано:

```
MASTERDIR= ${.CURDIR}/../cvsup
WITHOUT_X11= yes

.include "${MASTERDIR}/Makefile"
```

Как вы уже наверное догадались, директива `.include` «говорит» команде `make` искать все остальные описания в файле, относительный путь к которому `../cvsup/Makefile`. Чем же тогда отличается `net/cvsup` от `net/cvsup-without-gui`? Практически ничем, за исключением: `net/cvsup-without-gui` устанавливает переменную `WITHOUT_X11` в значение «yes», что позволяет нам собрать и установить программу CVSup без графического интерфейса. Того же эффекта можно было бы добиться выполнив следующую команду

```
# cd /usr/ports/net/cvsup && make WITHOUT_X11=yes install clean
```

Итак, мы установили CVSup и теперь можем им воспользоваться. В `/usr/share/examples/cvsup` лежат примеры файлов конфигурации для обновления базовой системы и коллекции портов. Файл, который нас интересует — `/usr/share/examples/cvsup/ports-supfile`. В этом файле конфигурации содержится описание CVSup-сервера, с которого мы будем получать обновления коллекции портов, список коллекций портов и другие параметры, необходимые для нормального функционирования CVSup.

⁵<http://www.cvsup.org/>

4.1 Конфигурационный файл CVSup

Один из самых простых способов обновить «порты» — изменить значение `host` с `CHANGE_THIS.FreeBSD.org` на имя хоста, на котором есть свежая коллекция портов. Подобных хостов достаточно много в Сети, какой из них выбрать? Попробуем выбрать ближайший. Но как его найти? В этом деле нам поможет утилита `sysutils/fastest_cvsup`.

4.2 Утилита `fastest_cvsup`(7)

Утилита `fastest_cvsup` представляет собой скрипт на языке `perl`⁶. Попробуем отыскать эту утилиту в коллекции портов с помощью команды `whereis(1)`

```
$ whereis fastest_cvsup
fastest_cvsup: /usr/ports/sysutils/fastest_cvsup
```

Установим утилиту.

```
# cd /usr/ports/sysutils/fastest_cvsup && make install clean
```

Попробуем воспользоваться этой утилитой, указав в качестве параметра страны «ru».

```
$ fastest_cvsup -c ru
>> Querying servers in countries: ru
....
>> Speed Daemons:
    - 1st: cvsup5.ru.freebsd.org
    - 2nd: cvsup6.ru.freebsd.org
    - 3rd: cvsup.ru.freebsd.org
```

Итак, для автора этой статьи — `cvsup5.ru.freebsd.org` самый быстрый (лучший) CVSup-сервер, с которого можно получать обновления для коллекции портов.

4.3 Получение обновлений

Изменим параметр `host` в конфигурационном файле для CVSup на `cvsup5.ru.freebsd.org`⁷. Если нам нужно обновить всю коллекцию портов — можно начинать процедуру обновления. Если вы хотите заблокировать обновления каких-либо портов или даже разделов — воспользуйтесь списком ответов на часто задаваемые вопросы по CVSup.

Выполним следующую команду

```
# cvsup -g -L 2 /usr/share/examples/cvsup/ports-update
```

Ключ «-g» указывает программе CVSup работать вне графического режима, ключ «-L» указывает программе CVSup уровень «разговорчивости»⁸.

Команда достаточно длинная и процесс обновления хочется как-то упростить. С другой стороны, мы изменили пример конфигурационного файла в директории, содержимое которой может измениться вследствие `make world`. Что делать? Есть альтернативный подход.

4.4 Использование `make.conf`(5)

В файле `/etc/make.conf` содержатся параметры, которые использует уже известная нам команда `make`. Мы знаем где находится нужный нам `supfile`, знаем имя хоста, с которого мы будем получать обновления «портов».

Допишем следующие строки в файл `/etc/make.conf`

```
SUP_UPDATE=    yes
SUP=           /usr/local/bin/cvsup
SUPFLAGS=     -g -L 2
SUPHOST=      cvsup5.ru.FreeBSD.org
PORTSSUPFILE= /usr/share/examples/cvsup/ports-supfile
```

⁶<http://www.perl.org/>

⁷Конфигурационный файл для программы CVSup называют `supfile`

⁸Более подробное описание этих и других ключей вы можете найти в `cvsup(1)`

Первая строка указывает на то, что мы готовы получать обновления. Вторая — команду, с помощью которой мы будем их получать. Остальные строки вам уже известны.

После этих изменений вы можете смело отправляться в каталог `/usr/ports` и набрать магическую команду

```
# make update
```

Обновления портов произойдёт «автоматически».

4.5 «Ненужные» порты

Коллекция портов огромна. Многие порты специфичны и могут оказаться вам не нужны. Что если Вы захотите отказаться от этих «ненужных» портов? Можно ли отказаться от получения обновлений для национальных категорий портов? Ответ: да, конечно, можно.

Достаточно создать файл `/usr/sup/ports-all/refuse`⁹, в котором описать «ненужные» коллекции портов.

```
ports/chinese
ports/french
ports/german
ports/hebrew
ports/hungarian
ports/japanese
ports/korean
ports/polish
ports/portuguese
ports/vietnamese
```

Вышеуказанный `refuse`-файл заблокирует приём обновлений для китайской, французской, немецкой, еврейской, венгерской, японской, корейской, польской, португальской и вьетнамской коллекций портов соответственно.

4.6 Обновление порта

Представим себе следующую ситуацию. Есть некий программный продукт, который уже присутствует в коллекции портов. У порта есть владелец, который его поддерживает. Узнать владельца можно с помощью команды `make`, указав ключу `-V` интересующую нас переменную `MAINTAINER`. Например,

```
$ make -f /usr/ports/www/apache13/Makefile -V MAINTAINER
ache@freebsd.org
```

или

```
$ (cd /usr/ports/www/apache13 && make maintainer)
ache@freebsd.org
```

Теперь допустим, что Фонд программного обеспечения Apache выпустил новую версию продукта `apache-1.3.27`, а в «портах» до сих старая предыдущая версия.

```
$ make -f /usr/ports/www/apache13/Makefile -V PORTVERSION
1.3.26
```

Что делать?

Попытаемся обновить порт своими руками. Воспользуемся командой `cvs(1)`.

```
$ export CVSROOT=:pserver:anoncvs@anoncvs.FreeBSD.org:/home/ncvs
$ cvs login
Logging in to :pserver:anoncvs@anoncvs.freebsd.org:2401/home/ncvs
CVS password: anoncvs
$ cvs co ports/www/apache13
...
```

⁹Подробности см. в CVSup ЧаВо

Добудем новую версию популярного WWW-сервера и положим её в специальный каталог для дистрибутивных файлов `/usr/ports/distfiles`¹⁰

Обычно, достаточно сделать две вещи: исправить версию программного продукта в `Makefile` порта и изменить контрольную сумму MD5 в файле `distinfo`. При необходимости следует убрать (или закомментировать) «лишние» описания из `Makefile` (такие как `PORTREVISION`, потому что это первая, вернее нулевая, ревизия новой версии порта) и исправить файл `pkg-plist`, в котором содержится список устанавливаемых файлов порта.

Изменим версию в `ports/www/apache13/Makefile` с 1.3.26 на 1.3.27. Изменим контрольную сумму дистрибутива в `distinfo`

```
$ cd ports/www/apache13 && make makesum
```

и соберём получившийся порт, установив переменную `LOCALBASE` в значение `/tmp/local`. Если порт собрался - его можно установить, исправить `pkg-plist`. Если порт устанавливается и удаляется нормально, с помощью команды `diff(1)` сохраним изменения в патч-файле.

```
$ cvs diff -u ports/www/apache13 > patch-apache13
```

Используя команду `send-pr(1)` составим отчёт о проблеме, включив в него наш патч.

5 Утилита `portupgrade(1)`

Утилита `portupgrade` представляет собой скрипт на языке `ruby`¹¹. Это достаточно мощная утилита, позволяющая обновлять уже установленные порты с помощью коллекции портов или пэкеджей.

5.1 Установка

```
# cd /usr/ports/sysutils/portupgrade && make install clean
```

5.2 Обновление установленных пакетов

В этой главе рассказывается о том, как обновить уже установленное ПО с помощью коллекции портов и утилиты `portupgrade`.

Представим себе задачу. Есть операционная система, которая была установлена достаточно давно. Плюс, используя коллекцию портов, в эту систему было установлено ПО прочих производителей. Коллекцию портов можно обновить с помощью `CVSup`, систему — с помощью `CVSup` и `make world`. А как же быть с уже установленными портами? Задача сама по себе нетривиальная, потому что сначала надо выяснить какие версии и каких пакетов установлены, потом вытащить свежие файлы дистрибутивов, собрать свежие версии портов, попутно переустанавливая зависимые пакеты, и только потом можно будет приостановить работу имеющихся программ, удалить старые программы и заменить их новыми. Помня, что современная оконная система и «окружения рабочего стола» (такие как `GNOME`, `KDE`) состоит из множества пакетов — решение подобной задачи может стать далеко не самым простым.

Всю эту работу за вас сможет сделать `portupgrade`!

В руководстве `portupgrade` можно найти не только большое количество различных ключей и их описаний, но и технические подробности функционирования этой утилиты. Есть и примеры использования.

Если вы хотите обновить все пакеты — укажите ключ «-а».

Допустим, что нужно обновить порт программы `gmake`. Некоторые технические подробности. Из файла `devel/gmake/Makefile` вы можете узнать, что этот порт зависит от пакета `devel/gettext`, который в свою очередь (смотрим `devel/gettext/Makefile`) зависит от `converters/libiconv` и `textproc/expat2`. Подобного рода зависимости могут продолжаться сколь угодно много. Чтобы не ходить по `Makefile`'ам в директориях всей коллекции портов, можно воспользоваться следующей командой

```
$ cd /usr/ports/devel/gmake && make pretty-print-build-depends-list
This port requires package(s) "expat-1.95.5 gettext-0.11.5_1
libiconv-1.8_2" to build.
```

Какие порты стоит обновлять? Только один `gmake`? Или может быть следует обновить и те порты, от которых он зависит? Вопрос очень интересный, одновременно простой и сложный.

¹⁰Этот каталог регулируется с помощью переменной `DISTDIR`

¹¹<http://www.ruby-lang.org/>

6 Порты против пакетов

Эта глава расскажет вам о том, в каких случаях стоит использовать порты, а в каких случаях лучше использовать пакеты и почему именно так стоит поступать.

Что представляет собой пакет? Пакет представляет собой уже готовый к использованию продукт. Команда `pkg_add(1)` устанавливает пакет. Сразу же после удачной установки пакет можно использовать. Необходимо отметить, что установка пакета занимает меньше времени, нежели сборка соответствующего этому пакету порта. Так как пакет это уже готовый к употреблению продукт (такой как маринованный огурец в банке), следовательно, появление новых свойств в уже собранном кем-то пакете невозможно.

Рассмотрим на примере. На компакт-диске с FreeBSD 4.8-RELEASE есть пакет, абсолютный путь к которому `packages/All/mutt-1.4.1.tgz`. В дереве портов присутствует соответствующий этому пакету порт `mail/mutt`. Заглянем внутрь `Makefile`. Директива `PACKAGE_BUILDING`

```
.if defined(PACKAGE_BUILDING)
WITH_SLANG= yes
BUILD_DEPENDS= ispell:${PORTSDIR}/textproc/ispell
RUN_DEPENDS= ispell:${PORTSDIR}/textproc/ispell \
urlview:${PORTSDIR}/textproc/urlview
.endif
```

указывают на то, что при «автоматической» сборке пакета из порта, пакет будет собран с поддержкой интерпретируемого языка `S-Lang` (установлена переменная `WITH_SLANG`) и интерактивного пакета проверки правописания `textproc/ispell`. Какие ещё переменные можно установить? Изменятся ли свойства приложения после установки других переменных?

Получим список переменных.

```
$ grep defined mail/mutt/Makefile | awk '{ print $2; }' | sort -u | \
  sed -e 's|[a-z]([A-Z]*\)|\1|g'
!(NOPORTDOCS)
(MUTT_CONFIGURE_ARGS)
(PACKAGE_BUILDING)
(WITH_LOCALES_FIX)
(WITH_MUTT_CYRUS_SASL)
(WITH_MUTT_XFACE)
(WITH_NCURSES_PORT)
(WITH_SGML_DOCS)
(WITH_SLANG)
```

Переменная	Описание возможностей
<code>NOPORTDOCS</code>	установка документации, поставляемой в составе дистрибутива
<code>WITH_LOCALES_FIX</code>	исправления в локализации (<code>isprint(3)</code>)
<code>WITH_NCURSES_PORT</code>	сборка с пакетом <code>devel/ncurses</code> (вместо системного <code>curses</code>)
<code>WITH_SGML_DOCS</code>	дополнительная документация (в формате HTML и txt)
<code>WITH_MUTT_XFACE</code>	поддержка X-Face
<code>WITH_MUTT_CYRUS_SASL</code>	поддержка SASL-аутентификации

Замечу, что порт `mail/mutt-devel` имеет ещё больше различных ключей, один из которых (`WITH_MUTT_NNTP`) позволяет сделать `mutt` полноценным NNTP-клиентом.

Итак, устанавливая уже собранный пакет вы экономите время на сборку, но можете лишиться других возможностей, которые предоставляет сборка из порта.

7 Советы

Очень часто, когда выходит новая версия какого-либо дистрибутива возникает вопрос: как быстро появится обновлённая версия порта для новой версии? Что можно сделать для того, чтобы ускорить этот процесс?

Первое, что следует сделать, написать письмо по адресу, указанному в поле `MAINTAINER` интересующего порта, сообщив о вашей заинтересованности. Но, владелец порта не робот и не может заниматься только портированием приложений, по-этому следует относиться с пониманием к его проблемам.

Второе. В тех случаях, когда в поле `MAINTAINER` указан адрес `ports@FreeBSD.org`, явный владелец порта отсутствует, а поддержкой таких портов занимается команда разработчиков портов FreeBSD.

Третье. При написании письма владельцу порта следует указать не только причины, по которым следует немедленно обновить тот или иной порт (появление новых возможностей, устранение проблем с безопасностью), но и указать как это можно сделать, т.е. прислать патч ¹².

Четвёртое. Воспользуйтесь web-интерфейсом базы данных отчётов о проблемах¹³ для поиска интересующей вас проблемы. Возможно, кто-то уже заявил о найденной вами проблеме, а решение существует.

Пятое. Если владелец порта долгое время не отвечает на ваш запрос, следует воспользоваться утилитой `send-pr(1)` для отправки запроса в базу данных отчётов о проблемах GNATS.

8 Послесловие

«Порты» оказались очень работоспособной и удобной в работе. Системные администраторы во многих уголках мира высоко ценят возможности, предоставляемые системой портов.

Для операционных систем NetBSD¹⁴ и OpenBSD¹⁵ существуют собственные коллекции портов, а создатели популярного дистрибутива Gentoo GNU/Linux¹⁶ использовали идеи коллекции портов для создания собственной системы портеджей.

9 Литература

При подготовке были использованы «FreeBSD Handbook»¹⁷, «Руководство пользователя FreeBSD»¹⁸ (в стадии перевода), «Руководство FreeBSD по созданию портов»¹⁹, «CVSup. Часто задаваемые вопросы»²⁰.

10 Полезные ссылки

По адресу <http://cvsweb.freebsd.org/ports/> можно получить доступ к коллекции портов через интерфейс CVSup. На страницах <http://freshports.org/> можно отслеживать изменения в коллекции портов.

11 Колофон

При подготовке этой статьи использовались операционная система FreeBSD, текстовый редактор VIM (`editors/vim`) и издательская система L^AT_EX (`print/teTeX`).

¹² Вопрос о том как сделать патч был частично рассмотрен в главе «Обновление порта»

¹³<http://www.freebsd.org/cgi/query-pr-summary.cgi?query>

¹⁴<http://www.NetBSD.org/>

¹⁵<http://www.OpenBSD.org/>

¹⁶<http://www.gentoo.org/>

¹⁷<http://www.freebsd.org/handbook/>

¹⁸<http://www.freebsd.org.ru/handbook/>

¹⁹<http://www.freebsd.org.ru/porters-handbook/>

²⁰<http://ozz.pp.ru/cvsup.html>