

## 10.1. Введение в установку пакетов

Несмотря на то что все UNIX-системы подобны друг другу, каждой присущ свой способ установки пакетов программного обеспечения.

В некоторых системах (IRIX, HP-UX) программное обеспечение реализуется только путем компиляции из исходных кодов. Это не всегда удобно. Во-первых, компиляция всегда занимает время. Намного проще установить пакет с уже откомпилированной программой. По сути, *пакет* — это архив, содержащий программу и файлы, необходимые для ее выполнения, а также инструкции по установке программы (в какой каталог устанавливать, какие программы выполнить до и после установки и т.д.). Во-вторых, не всегда можно откомпилировать исходный код. В вашей системе может быть просто не установлен компонент, необходимый для компиляции пакета.

В Solaris поступили проще. Для установки программы нужно найти уже откомпилированный “бинарник” (подходящий для вашей архитектуры) и самостоятельно установить его, т.е. скопировать в определенный каталог файловой системы. Это намного быстрее, чем компилировать программу из исходного текста.

В Linux используется концепция пакетов. Существуют два основных формата пакетов: RPM (дистрибутивы Red Hat, Fedora, Mandriva, openSUSE и др.) и DEB (Debian, Ubuntu и дистрибутивы, основанные на этих двух дистрибутивах). Как уже было отмечено выше, пакет — это специальный архив, содержащий программу, необходимые файлы и инструкции по установке программы. Пакет также может содержать инструкции по удалению программы, а также описание зависимостей и конфликтов (об этом мы поговорим отдельно).

Как правило, в пакете находится уже откомпилированная программа. Установщик (в Linux — это программа `rpm` для RPM-пакетов и `dpkg` для DEB-пакетов) распаковывает архив

(пакет), выполняет указанные в нем действия и копирует файлы в указанные в пакете места файловой системы.

Кроме установки из пакетов, в Linux есть еще два способа установки программ: из исходного кода и установки из пакета, содержащего исходный код. Оба способа требуют установки компилятора `gcc` и еще ряда дополнительных программ. Разница лишь в том, что в первом случае вы сами загружаете, распаковываете архив с исходным текстом, читаете файл `README` (без него лучше не начинать сборку программы, дабы не наделать ошибок), а затем уже вводите команды, описанные в файле `README` (как правило, это `make`; `make install`, но могут быть и отличия), а во втором случае все это делает за вас программа `rpm`.

В FreeBSD также возможны два способа установки программ: из пакетов или из исходного кода (из портов). Порты нужно воспринимать как большую коллекцию исходного кода программ. Установка программы из портов — это, по сути, компиляция исходного текста программы, но в FreeBSD это настолько просто, что вы будете приятно удивлены. Вам можно не читать огромные файлы `README` — нужно ввести только одну команду, указывающую системе, какую программу следует откомпилировать и установить. Конечно, иногда и при установке из портов возникают проблемы, не без этого. В Linux могут возникнуть проблемы на ровном месте даже при установке уже откомпилированных программ, упакованных в пакеты, не говоря уже о компиляции программ из исходного кода.

Управление пакетами в FreeBSD осуществляется с помощью набора инструментов `pkg*`. Во многом управление пакетами в FreeBSD напоминает управление пакетами в Linux. Как и в Linux, в FreeBSD ведется база данных установленного программного обеспечения, в которой хранится информация о названиях и версиях установленных программ. База данных позволяет отслеживать зависимости и конфликты между пакетами. Инструменты `pkg*` могут производить установку (как с локального диска, так и с FTP-сервера), модернизацию и удаление пакета.

Устанавливаемые программы помещаются в каталог `/usr/local`. Вот краткое описание подкаталогов этого каталога:

- `bin` — содержит бинарные файлы, т.е. откомпилированные программы;
- `etc` — файлы конфигурации;
- `include` — содержит `include`-файлы на языке C, они используются для разработки новых программ;
- `info` — различная документация;
- `lib` — совместно используемые библиотеки;
- `libexec` — вспомогательные файлы, которые используются другими программами;
- `man` — страницы руководств для установленных программ;
- `sbin` — системные бинарные файлы установленных программ;
- `share` — файлы, не зависящие от платформы, например, набор картинок, которые будут использовать программы.

## 10.2. Зависимости и конфликты

Предположим, что есть некая библиотека функций `library`. Ее используют разработчики программ Иванов (пусть программа называется И), Петров (программа П) и Сидоров (программа С). Библиотека занимает 20 Мбайт, а программы других разработчиков — 1-2 Мбайт.

Суммарно набор программа плюс библиотека будет занимать 21-22 Мбайт. Если создать три пакета, содержащие программу и библиотеку, то у нас получится три пакета И, П и С общим объемом около 66 Мбайт. А можно пойти другим путем. Можно создать пакет L, содержащий библиотеку, и создать три маленьких пакета И, П и С. Тогда весь набор из четырех пакетов будет занимать всего 26 Мбайт. Так намного рациональнее, правда?

Поскольку программы И, П и С нуждаются в библиотеке `libgdy`, то говорят, что пакеты И, П и С *зависят от пакета L*. Это и есть пример зависимости пакетов. Пока вы не установите пакет L, вы не сможете установить ни один из пакетов И, П и С. Информация о зависимостях помещается в пакеты И, П и С, поэтому менеджер пакетов знает, какой пакет необходим для установки каждого из этих пакетов.

Теперь рассмотрим пример конфликта пакетов. Предположим, что вы хотите установить пакеты `sendmail` и `postfix`. Обе эти программы являются агентами отправки почты (MTA), но в системе может быть только один MTA, поэтому пакеты `sendmail` и `postfix` *конфликтуют друг с другом*.

## 10.3. Просмотр установленных пакетов: `pkg_info`

С помощью программы `pkg_info` можно просмотреть список установленных пакетов. Если запустить программу `pkg_info` без параметров, то вы увидите все установленные пакеты.

```
# pkg_info | more
```

Можно просмотреть список пакетов, относящихся к графической среде `gnome`.

```
# pkg_info | grep gnome
```

```
evolution-webcal-2.26.0_1 Webcal(endar) handler for gnome
gnome-applets-2.26.3_1 Applets components for the GNOME 2 Desktop Environment
gnome-audio-2.22.2 Sound files for use with GNOME 2
gnome-backgrounds-2.24.1 A collection of backgrounds for GNOME 2
gnome-control-center-2.26.0_2 Control center for GNOME 2 project
gnome-desktop-2.26.3_1 Additional UI API for GNOME 2
gnome-doc-utils-0.16.1_1 GNOME doc utils
gnome-games-2.26.3_1 The game applications package for the GNOME 2 Desktop Envir
gnome-icon-theme-2.26.0_2 A collection of icons for the GNOME 2 desktop
gnome-keyring-2.26.3_1 A program that keeps passwords and other secrets
gnome-mag-0.15.8 GNOME screen magnifier
gnome-media-2.26.0_4 Multimedia applications for the GNOME 2 desktop
gnome-menus-2.26.2 Implementation of the FreeDesktop Desktop Menu Spec
gnome-mime-data-2.18.0_3 A MIME and Application database for GNOME
gnome-mount-0.8_3 A front-end to mount, umount, and eject using HAL
gnome-netstatus-2.26.0_1 A GNOME applet that shows network status
gnome-nettool-2.26.2_1,1 A GNOME utility that provides network information sta-
tistic
gnome-panel-2.26.3_1 Panel component for the GNOME 2 Desktop
gnome-power-manager-2.24.4_5 Power management system for the GNOME Desktop
gnome-screensaver-2.26.1_1 GNOME screen saver and locker
gnome-session-2.26.2_1 Session component for the GNOME 2 desktop
```

```

gnome-settings-daemon-2.26.1_2 GNOME 2 settings daemon
gnome-speech-0.4.25 GNOME text-to-speech API
gnome-spell-1.0.8_3 A GNOME/Bonobo component for spell checking
gnome-system-monitor-2.26.2_1 GNOME 2 system monitor program
gnome-system-tools-2.22.2_1 Tools to configure workstations
gnome-terminal-2.26.3.1_2 Terminal component for the GNOME 2 Desktop
gnome-themes-2.26.3.1_1 A collection of themes and icons for GNOME2
gnome-user-docs-2.26.2 GNOME 2 users guide
gnome-user-share-2.26.0_1 Personal file-sharing service for the GNOME Desktop
gnome-utils-2.26.0_2,1 GNOME 2 support utilities
gnome-vfs-2.24.1_1 GNOME Virtual File System
gnome2-2.26.3 The "meta-port" for the GNOME 2 integrated X11 desktop
gnome_subr-1.0 Common startup and shutdown subroutines used by GNOME scrip
gnomehier-2.3_12 A utility port that creates the GNOME directory tree
...

```

Вывод я немного сократил, так как не вижу необходимости приводить его полностью. Для просмотра информации о конкретном пакете используется следующая команда.

```
# pkg_info <имя пакета>
```

Пример.

```
# pkg_info bash-4.0.24
```

```
Information for bash-4.0.24:
```

```
Comment:
```

```
The GNU Project's Bourne Again SHell
```

```
Required by:
```

```

rarian-0.8.1
gnome-doc-utils-0.16.1_1
policykit-gnome-0.9.2_2
gnome-mount-0.8_3
gvfs-1.2.3_2
libgnome-2.26.0_1
libbonoboui-2.24.1_1
libgnomeui-2.24.1_1
evolution-data-server-2.26.3_1
bug-buddy-2.26.0_1
py26-gnome-2.26.1_1
gnome-desktop-2.26.3_1
nautilus-2.26.3_1
evince-2.26.2_2
gnome-media-2.26.0_4
gnome-panel-2.26.3_1
gtksourceview-1.8.5_5
zenity-2.26.0_1
metacity-2.26.0_1
totem-pl-parser-2.26.2_1
py26-gnome-desktop-2.26.0_1
alacarte-0.12.1_1
brasero-2.26.3_1
cheese-2.26.3_1

```

dasher-4.10.1\_1,2  
deskbar-applet-2.26.2\_1  
ekiga-2.0.11\_6  
empathy-2.26.2\_1  
eog-2.26.3\_1  
gnome-themes-2.26.3.1\_1  
epiphany-2.26.3\_3  
gnome-spell-1.0.8\_3  
gtkhtml3-3.26.3\_1  
evolution-2.26.3\_1  
evolution-exchange-2.26.3\_1  
evolution-webcal-2.26.0\_1  
file-roller-2.26.3\_1,1  
gcalctool-5.26.3\_1,2  
gconf-editor-2.26.0\_1,1  
gnome-settings-daemon-2.26.1\_2  
gnome-session-2.26.2\_1  
gdm-2.26.1\_6  
gedit-2.26.3\_1  
gucharmap-2.26.3.1\_1  
gnome-applets-2.26.3\_1  
gnome-control-center-2.26.0\_2  
gnome-games-2.26.3\_1  
gnome-netstatus-2.26.0\_1  
gnome-nettool-2.26.2\_1,1  
gnome-power-manager-2.24.4\_5  
gnome-screensaver-2.26.1\_1  
gnome-system-monitor-2.26.2\_1  
gnome-system-tools-2.22.2\_1  
gnome-terminal-2.26.3.1\_2  
gnome-user-docs-2.26.2  
gnome-user-share-2.26.0\_1  
gnome-utils-2.26.0\_2,1  
gok-2.26.0\_1,1  
hamster-applet-2.26.3\_1  
libgail-gnome-1.20.1\_1  
mousetweaks-2.26.3\_1  
orca-2.26.3\_1  
seahorse-2.26.2\_2  
seahorse-plugins-2.26.2\_2  
sound-juicer-2.26.1\_1  
totem-2.26.3\_1  
vinagre-2.26.2\_1  
vino-2.26.2\_2  
yelp-2.26.0\_1  
gnome2-2.26.3

**Description:**

This is GNU Bash. Bash is the GNU Project's Bourne Again SHell, a complete implementation of the POSIX.2 shell spec, but also with interactive command line editing, job control on architectures that support it, csh-like features such as history substitution and brace expansion, and a slew of other features.

WWW: <http://cnswww.cns.cwru.edu/~chet/bash/bashtop.html>

Теперь разберемся, какие поля есть в выводе:

- `Comment` — комментарий, краткое описание пакета;
- `Required by` — список пакетов, которые зависят от данного пакета;
- `Description` — полное описание пакета;
- `WWW` — адрес сайта разработчика пакета.

Если вы хотите получить полную информацию о пакете, тогда используйте параметр `-v`.

```
# pkg_info -v <имя пакета>
```

Пример.

```
# pkg_info -v bash-4.0.24
```

Полный вывод я сократил (уж очень он большой), но зато добавил в него свои комментарии (они выделены жирным), так что внимательно прочитайте его.

#### **Информация о пакете bash-4.0.24**

Information for bash-4.0.24:

#### **Краткий комментарий:**

Comment:

The GNU Project's Bourne Again SHell

#### **Полное описание пакета:**

Description:

This is GNU Bash. Bash is the GNU Project's Bourne Again SHell, a complete implementation of the POSIX.2 shell spec, but also with interactive command line editing, job control on architectures that support it, csh-like features such as history substitution and brace expansion, and a slew of other features.

#### **Страничка разработчиков:**

WWW: <http://cnswww.cns.cwru.edu/~chet/bash/bashtop.html>

#### **Список упаковки:**

Packing list:

Comment: PKG_FORMAT_REVISION:1.1	<b>версия формата</b>
Package name: bash-4.0.24	<b>имя пакета</b>
Package origin: shells/bash	<b>группа и имя пакета</b>
CWD to /usr/local	<b>перейти в /usr/local при установке</b>

#### **Этот пакет зависит всего от двух пакетов: libiconv и gettext**

Dependency: libiconv-1.13.1

dependency origin: converters/libiconv

Dependency: gettext-0.17\_1

dependency origin: devel/gettext

#### **Пакет конфликтует со старыми версиями bash**

Conflicts: bash-[0-35-9].\*

Conflicts: bash-static-[0-9]\*

Далее идет список устанавливаемых файлов, я его сократил

```
File: man/man1/bash.1.gz
  Comment: MD5:434b5c71eac4d944c3a51fd23873ec78
File: man/man1/bashbug.1.gz
  Comment: MD5:bb4d94374b9a4e0da99fbc3052c0f6dc
UNEXEC — это команда для удаления файла
UNEXEC 'rm -f %D/man/cat1/bash.1.gz %D/man/cat1/
bash.1.gz %D/man/cat1/bash.1.gz.gz %D/man/cat1/bash.1.gz.bz2'
UNEXEC 'rm -f %D/man/cat1/bashbug.1.gz %D/man/cat1/bashbug.1 %D/man/cat1/
bashbug.1.gz %D/man/cat1/bashbug.1.gz.gz %D/man/cat1/bashbug.1.gz.bz2'
  Comment: $FreeBSD: ports/shells/bash/pkg-plist,v 1.25 2009/03/12 09:54:01
osa Exp $
File: bin/bash
  Comment: MD5:a2c80d02ea340deecc4c6021179d6b57
...

```

#### Сценарий установки:

```
Install script:
#!/bin/sh
#
# $FreeBSD: ports/shells/bash/pkg-install,v 1.6 2007/08/04 11:41:18 gabor Exp $
#

```

```
BASH="$(echo ${PKG_PREFIX-/usr/local}/bin/bash | /usr/bin/sed -e 's|/||g')"
RBASH="$(echo ${PKG_PREFIX-/usr/local}/bin/rbash | /usr/bin/sed -e 's|/||g')"
SHELLS="/etc/shells"
```

```
case $2 in
POST-INSTALL)
  if [ -d "${SHELLS%/*}" ] && ! grep -qs "^$BASH$" "$SHELLS"; then
    if [ 'id -u' -eq 0 ]; then
      echo "$BASH" >> "$SHELLS"
    else
      echo "Not root, please add $BASH to $SHELLS manually"
    fi
  fi
  if [ -d "${SHELLS%/*}" ] && ! grep -qs "^$RBASH$" "$SHELLS"; then
    if [ 'id -u' -eq 0 ]; then
      echo "$RBASH" >> "$SHELLS"
    else
      echo "Not root, please add $RBASH to $SHELLS manually"
    fi
  fi
fi
;;
esac

```

#### Сценарий удаления:

```
De-Install script:
#!/bin/sh
#
# $FreeBSD: ports/shells/bash/pkg-deinstall,v 1.5 2007/06/29 01:06:38 obrien
Exp $
#

```

```
BASH="$(echo ${PKG_PREFIX-/usr/local}/bin/bash | /usr/bin/sed -e 's|/|/|g')"
```

```
RBASH="$(echo ${PKG_PREFIX-/usr/local}/bin/rbash | /usr/bin/sed -e 's|/|/|g')"
```

```
SHELLS="${PKG_DESTDIR-}/etc/shells"
```

```
case $2 in
```

```
DEINSTALL)
```

```
  if grep -qs "^$BASH\$" "$SHELLS"; then
```

```
    if [ 'id -u' -eq 0 ]; then
```

```
      TMPSHELLS='mktemp -t shells'
```

```
      grep -v "^$BASH\$" "$SHELLS" > "$TMPSHELLS"
```

```
      cat "$TMPSHELLS" > "$SHELLS"
```

```
      rm "$TMPSHELLS"
```

```
    else
```

```
      echo "Not root, please remove $BASH from $SHELLS manually"
```

```
    fi
```

```
  fi
```

```
  if grep -qs "^$RBASH\$" "$SHELLS"; then
```

```
    if [ 'id -u' -eq 0 ]; then
```

```
      TMPSHELLS='mktemp -t shells'
```

```
      grep -v "^$RBASH\$" "$SHELLS" > "$TMPSHELLS"
```

```
      cat "$TMPSHELLS" > "$SHELLS"
```

```
      rm "$TMPSHELLS"
```

```
    else
```

```
      echo "Not root, please remove $RBASH from $SHELLS manually"
```

```
    fi
```

```
  fi
```

```
;;
```

```
esac
```

**Далее описывается дерево каталогов (mtree), но для нас оно не представляет никакой ценности.**

Как видите, пакет `bash` содержит описание зависимостей и конфликтов, а также сценарии установки и удаления. Кстати, сценарий установки добавляет оболочку `bash` в файл `/etc/shells`, а сценарий удаления удаляет оболочку `bash` из этого файла.

Теперь давайте узнаем, какие пакеты нуждаются обновления. Для этого введите такую команду.

```
# pkg_version -v | grep need
```

Я перенаправил вывод команды на фильтр `grep` по строке `'need'`. Дело в том, что напротив пакетов, которые нуждаются в обновлении, будет указана строка `'needs updating'` (нуждается в обновлении), а напротив актуальных версий пакетов будет строка `'up to date with port'` (соответствует версии порта) — таких пакетов будет значительно больше, поэтому я и использовал фильтр по строке `'need'`.

Вот как выглядит результат выполнения вышеуказанной команды.

```
bn-freebsd-doc-20090820 < needs updating (port has 20090913)
da-freebsd-doc-20090820 < needs updating (port has 20090913)
dbus-1.2.4.6 < needs updating (port has 1.2.4.6_1)
de-freebsd-doc-20090820 < needs updating (port has 20090913)
el-freebsd-doc-20090820 < needs updating (port has 20090913)
en-freebsd-doc-20090820 < needs updating (port has 20090913)
epiphany-2.26.3_3 < needs updating (port has 2.26.3_4)
```

```

es-freebsd-doc-20090820 < needs updating (port has 20090913)
farsight2-0.0.14 < needs updating (port has 0.0.15)
fr-freebsd-doc-20090820 < needs updating (port has 20090913)
gdm-2.26.1_6 < needs updating (port has 2.26.1_7)
gio-fam-backend-2.20.4 < needs updating (port has 2.20.5)
glib-2.20.4 < needs updating (port has 2.20.5)
gnome-media-2.26.0_4 < needs updating (port has 2.26.0_5)
gnome-screensaver-2.26.1_1 < needs updating (port has 2.26.1_3)
gnupg-2.0.12 < needs updating (port has 2.0.13)
gstreamer-plugins-bad-0.10.13,3 < needs updating (port has 0.10.14,3)
gstreamer-plugins-dts-0.10.13,3 < needs updating (port has 0.10.14,3)
gstreamer-plugins-esound-0.10.15,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-flac-0.10.15,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-gconf-0.10.15_1,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-good-0.10.15,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-hal-0.10.15,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-libpng-0.10.15,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-soup-0.10.15_1,3 < needs updating (port has 0.10.16,3)
gstreamer-plugins-xvid-0.10.13,3 < needs updating (port has 0.10.14,3)
gtk-2.16.5_1 < needs updating (port has 2.16.6)
hal-0.5.11_25 < needs updating (port has 0.5.11_26)
hu-freebsd-doc-20090820 < needs updating (port has 20090913)
iso-codes-3.10.2 < needs updating (port has 3.10.3)
it-freebsd-doc-20090820 < needs updating (port has 20090913)
ja-freebsd-doc-20090820 < needs updating (port has 20090913)
libaudiofile-0.2.6 < needs updating (port has 0.2.6_1)
libgphoto2-2.4.6_1 < needs updating (port has 2.4.7)
liboobs-2.22.1 < needs updating (port has 2.22.2)
librsvg2-2.26.0_1 < needs updating (port has 2.26.0_2)
libsigc++-2.2.3 < needs updating (port has 2.2.4.2)
libxml2-2.7.3_1 < needs updating (port has 2.7.4_1)
mn-freebsd-doc-20090820 < needs updating (port has 20090913)
nautilus-2.26.3_1 < needs updating (port has 2.26.3_2)
nl-freebsd-doc-20090820 < needs updating (port has 20090913)
nss-3.11.9_3 < needs updating (port has 3.12.4)
openldap-client-2.4.17 < needs updating (port has 2.4.18)
p5-Compress-Raw-Zlib-2.020 < needs updating (port has 2.021)
pl-freebsd-doc-20090820 < needs updating (port has 20090913)
png-1.2.38 < needs updating (port has 1.2.40)
policykit-0.9_4 < needs updating (port has 0.9_5)
pt-freebsd-doc-20090820 < needs updating (port has 20090913)
...

```

Как видите, выводятся не только имена пакетов, нуждающихся в обновлении, но и версии новых пакетов, до которых можно обновиться (версия указывается после строки 'port has').

На диске пакеты хранятся в виде файлов `.tbz`. Например, `имя_версия.tbz`. В имени Linux-пакетов обязательно указывается платформа, для которой предназначен пакет. В FreeBSD формат имени пакета проще — указывается только имя пакета, версия пакета и `.tbz` (`.tbz` означает, что архив сначала был упакован с помощью команды `tar`, а потом сжат с помощью архиватора `bzip`). Такая практика упаковки стандартна для среды UNIX: вы сначала упаковываете каталог в один файл с помощью утилиты `tar` (без сжатия), а потом сжимаете его с помощью архиватора (программа `bzip`). Чтобы просмотреть информацию о пакете,

вам нужно указать его имя и версию (см. выше), одного имени недостаточно. Зато можно не указывать `.tbz` — хоть так проще.

## 10.4. Установка пакетов с помощью конфигуратора `sysinstall`

Если вы установили коллекцию портов при установке системы, то просмотреть программы, доступные для установки, можно в каталоге `/usr/ports` (рис. 10.1).

```
ls /usr/ports
```

```
den# ls /usr/ports
.cvsignore      arabic           emulators       mbone           shells
CHANGES        archivers       finance         misc            sysutils
COPYRIGHT       astro           french          multimedia     textproc
GIDs            audio           ftp             net             ukrainian
INDEX-8         benchmarks     games           net-im         vietnamese
KNOBS           biology         german          net-mgmt       www
LEGAL           cad             graphics        net-p2p        x11
MOVED           chinese        hebrew         news           x11-clocks
Makefile        comms          hungarian      palm           x11-drivers
Mk              converters     irc            polish         x11-fm
README          databases      japanese       ports-mgmt     x11-fonts
Templates       deskutils      java           portuguese     x11-servers
Tools           devel          korean         print          x11-themes
UIDs            distfiles      lang           russian        x11-toolkits
UPDATING        dns            mail           science        x11-wm
accessibility   editors        math           security
```

Рис. 10.1. Содержимое каталога `/usr/ports`

Но установка из портов занимает много времени, поэтому намного проще и удобнее использовать конфигуратор `sysinstall` для установки пакетов.

```
# sysinstall
```

В появившемся окне (рис. 10.2) выберите команду `Configure`, а затем — `Packages`. Далее нужно выбрать источник установки (рис. 10.3). В большинстве случаев вы выберете команду `Install from a FreeBSD CD/DVD` для установки пакетов с установочного CD/DVD. Если вы подключены к Интернету, можете выбрать команду `Install from an FTP server` для установки пакетов с FTP-сервера.

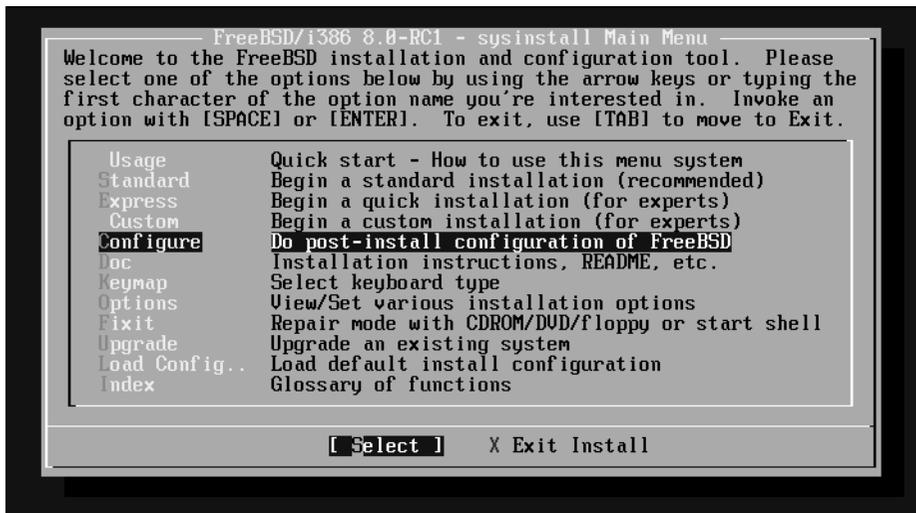


Рис. 10.2. Конфигуратор sysinstall

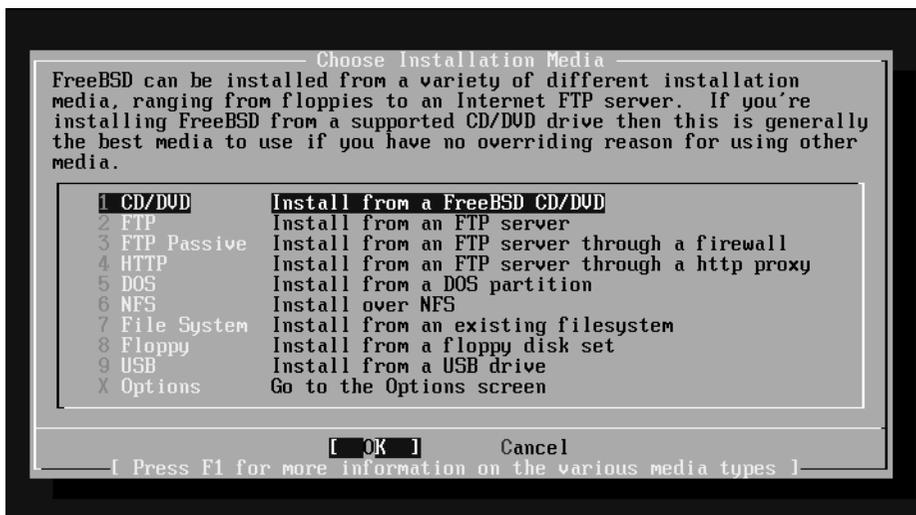


Рис. 10.3. Источник пакетов

Далее вы увидите список категорий пакетов (рис. 10.4). Выберите категорию и нажмите клавишу <Enter>.

Выберите категорию (категория All отображает все доступные пакеты), и Sysinstall отобразит список пакетов выбранной категории (рис. 10.5). Пакеты, отмеченные крестиком, уже установлены. Для установки пакета отметьте его крестиком (выберите пакет стрелками и нажмите <Пробел>). На рис. 10.5 выделен пакет mutt — удобный почтовый клиент с текстовым интерфейсом. Аналогично, сняв крестик напротив имени пакета, вы помечаете его для удаления.

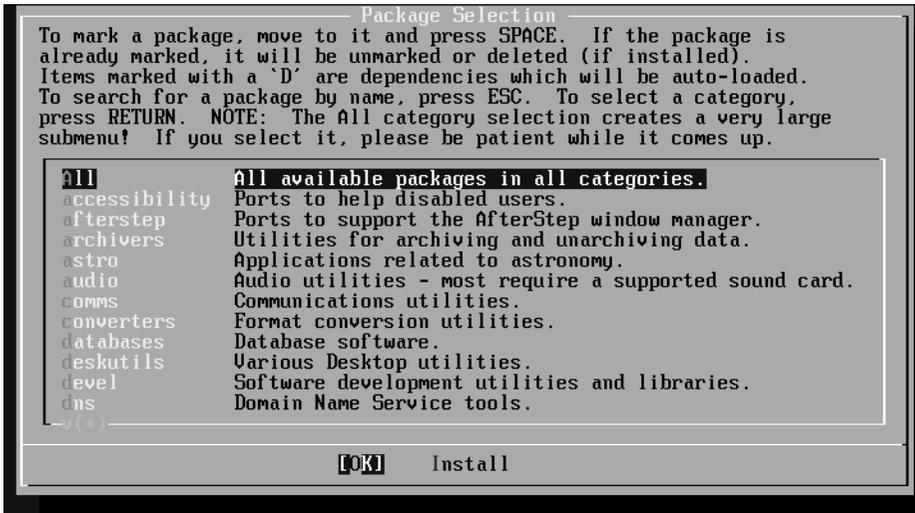


Рис. 10.4. Список категорий пакетов

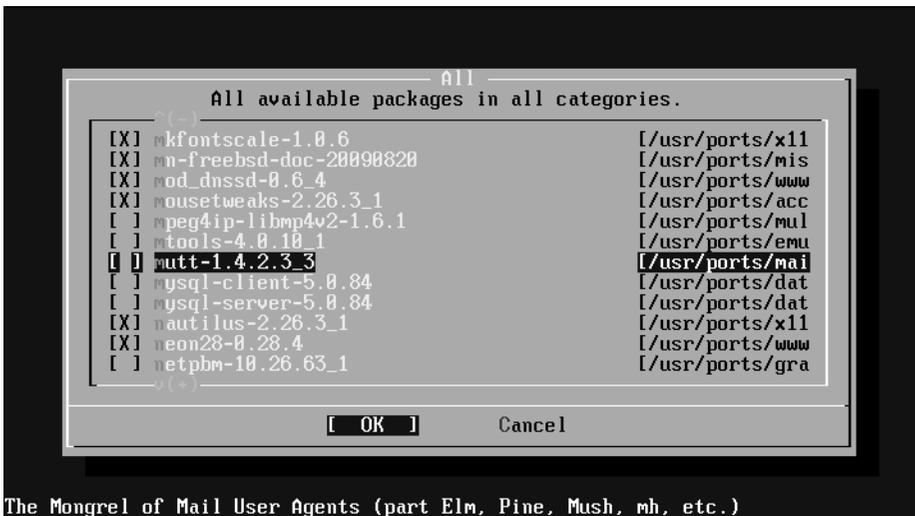


Рис. 10.5. Выбор пакетов для установки

Выбрав необходимые вам пакеты, нажмите клавишу <Tab> для перехода к кнопке OK, после чего нажмите клавишу <Enter>. Далее опять нажмите <Tab> для перехода к кнопке Install (см. рис. 10.4) и нажмите <Enter> для установки выбранных пакетов. Конфигуратор отобразит список выбранных пакетов (рис. 10.6). Для их установки щелкните на кнопке OK.

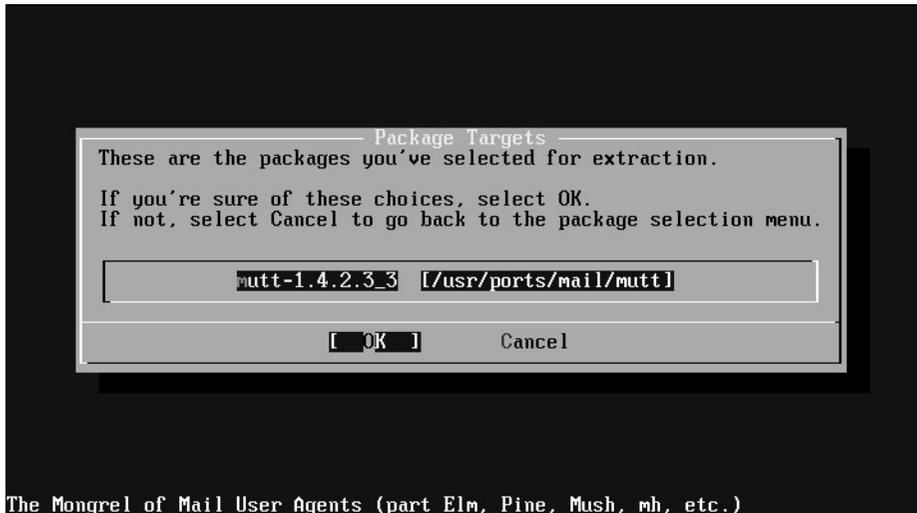


Рис. 10.6. Пакеты, отмеченные для установки

Далее вы увидите процесс установки пакета (рис. 10.7).

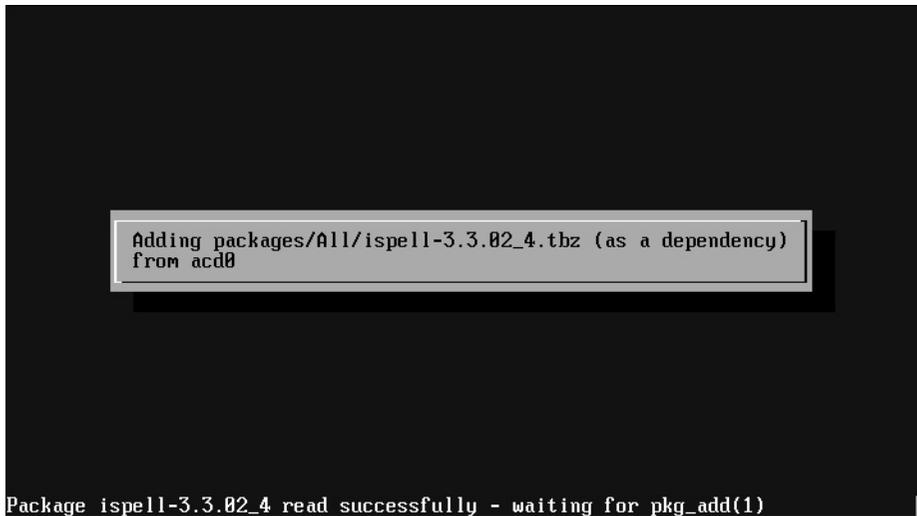


Рис. 10.7. Установка пакета для разрешения зависимости

Обратите внимание на то, что `sysinstall` умеет разрешать зависимости. Например, выбранный пакет `mutt` зависит от пакета `ispell`, поэтому конфигуратор сначала установит этот пакет, отметив, что пакет устанавливается для разрешения зависимости (*as a dependency*) (см. рис. 10.7), а затем установит пакет `mutt` (рис. 10.8).

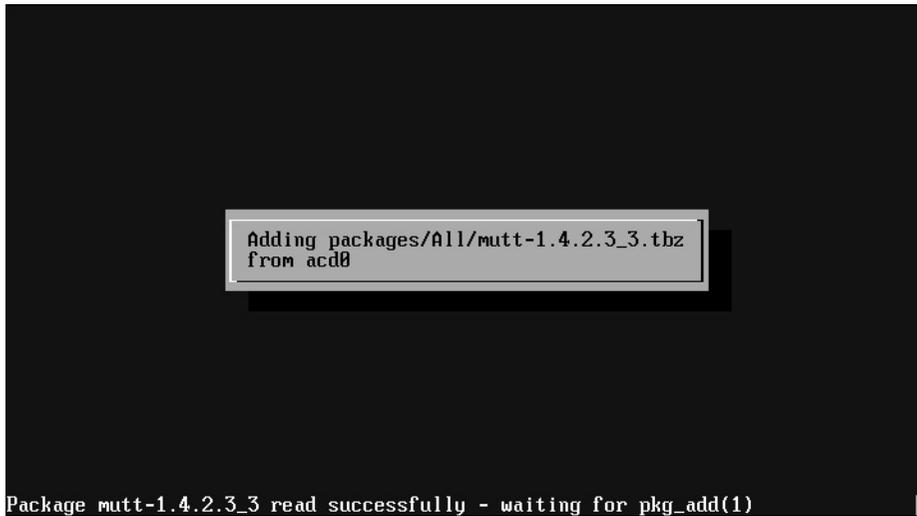


Рис. 10.8. Установка пакета mutt

После установки пакетов вы опять вернетесь в окно выбора категории пакетов (см. рис. 10.4). Как выйти из него? Щелкните на кнопке **Install** и согласитесь, что вы ничего не выбрали, щелкнув на кнопке **OK** (рис. 10.9), после этого выберите команду **Exit** (рис. 10.10), а потом — **Exit Install** (рис. 10.11).

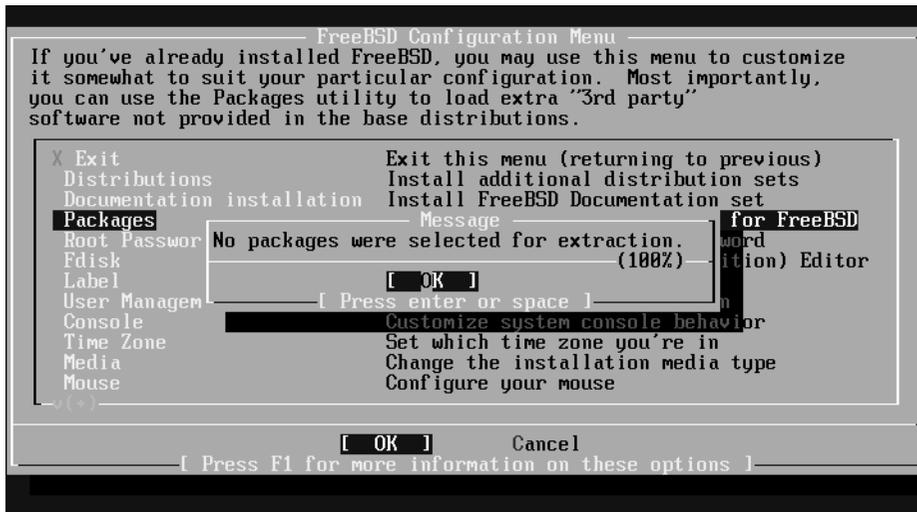


Рис. 10.9. Щелкните на кнопке **OK**

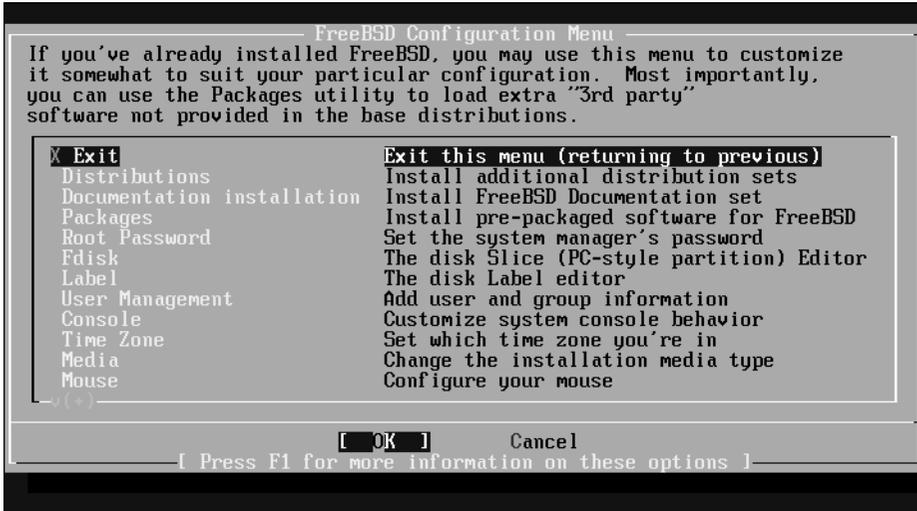


Рис. 10.10. Выберите команду Exit

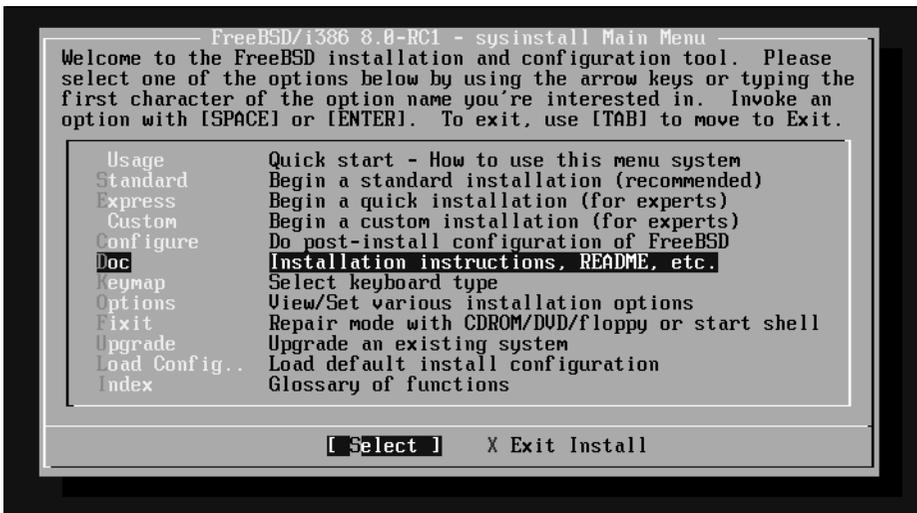


Рис. 10.11. Щелкните на кнопке Exit Install

Могу сказать только одно: установка пакетов в FreeBSD осуществляется проще, чем выход из конфигулятора sysinstall!

На рис. 10.12 изображен установленный нами почтовый клиент mutt.

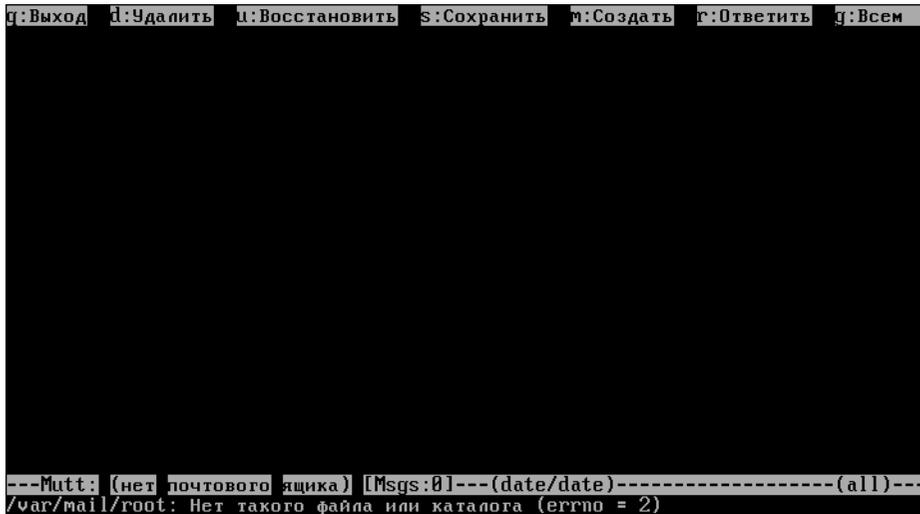


Рис. 10.12. Почтовый клиент mutt

## 10.5. Установка пакетов с помощью программы pkg\_add

Использовать конфигурактор sysinstall удобно, когда вы устанавливаете пакеты из коллекции портов. Но когда вам нужно установить пакет, полученный из другого источника, используйте программу `pkg_add`.

Первым делом загрузите пакет. Это можно сделать вручную, например, с помощью команды `fetch`.

```
fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/8.0-RC3/packages/mail/
pgen-0.2.tbz
```

После того как пакет будет получен, введите такую команду.

```
pkg_add <имя пакета>
```

Пример.

```
pkg_add pgen-0.2.tbz
```

Учитывая, что программа `pkg_add` умеет использовать протокол FTP, вы можете сразу ввести такую команду.

```
pkg_add ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/8.0-RC3/packages/
mail/pgen-0.2.tbz
```

Программа `pkg_add` сначала загрузит пакет, а затем — установит его. Вы можете загружать пакеты с любого источника, а не только с FTP-сервера `ftp.freebsd.org`.

Хотя можно сразу выполнить команду `pkg_add`, я предпочитаю сначала получить файл с помощью команды `fetch`, а затем с помощью команды `md5` проверить его контрольную сумму.

```
md5 <имя файла>
```

Если контрольная сумма полученного файла совпадет с контрольной суммой, которую можно узнать на сайте источника пакета, то пакет можно устанавливать — он загружен корректно. В противном случае нужно загрузить пакет еще раз.

## 10.6. Обновление и удаление пакета

Удалить пакет можно с помощью команды `pkg_delete`.

**# pkg\_delete пакет**

Имя пакета нужно указывать вместе с версией, иначе удалить пакет не получится.

```
pkg_delete pgen-0.2      # правильно
pkg_delete pgen         # неправильно
```

При удалении пакета не нужно указывать “расширение” (слово “расширение” в кавычках, потому что в UNIX нет такого понятия, как расширение файла, зато читателю понятнее) `.tbz`.

А вот обновление пакетов в FreeBSD происходит не так, как бы нам этого хотелось. В Linux менеджер пакетов все делает автоматически: предлагает новую версию пакета, если она доступна; и, если вы согласились на обновление, то он удаляет старую версию и устанавливает новую. В FreeBSD нужно все делать вручную. Допустим, мы хотим обновить пакет `package-1.0` до версии 1.1. Для этого нужно ввести такую команду.

```
pkg_delete package-1.0
```

Затем нужно ввести команду установки новой версии пакета.

```
pkg_add package-1.1.tbz
```

Обратите внимание: при установке нужно обязательно указывать “расширение” файла `.tbz`.

При обновлении пакета не нужно волноваться, что будут удалены его конфигурационные файлы, которые вы изменили для себя. При удалении пакета обычно сравниваются конфигурационные файлы (с помощью команды `UNEXEC`): если конфигурационный файл на диске отличается от файла по умолчанию (если вы его изменили), то он не будет удален.

## 10.7. Обновление пакета с помощью портов

Помните, чуть выше мы проверяли актуальность наших портов. Допустим, у нас есть пакеты, которые не актуальны (доступны новые версии этих пакетов).

```
gdm-2.26.1_6 < needs updating (port has 2.26.1_7)
```

Мы видим, что установлена версия `2.26.1_6` пакета `gdm`, а в портах есть версия `2.26.1_7`. Чтобы не тратить время на удаление и установку, можно использовать утилиту `portupgrade`. Использовать утилиту очень просто. Нужно передать ей в качестве параметра только имя пакета (без версии) — новую версию утилита найдет и установит автоматически.

```
# /usr/local/sbin/portupgrade gdm
```

Вот только утилита `portupgrade` не устанавливается по умолчанию, и ее нужно установить самостоятельно (рис. 10.13). Процесс обновления пакета `gdm` (точнее последние строки вывода) изображен на рис. 10.14.

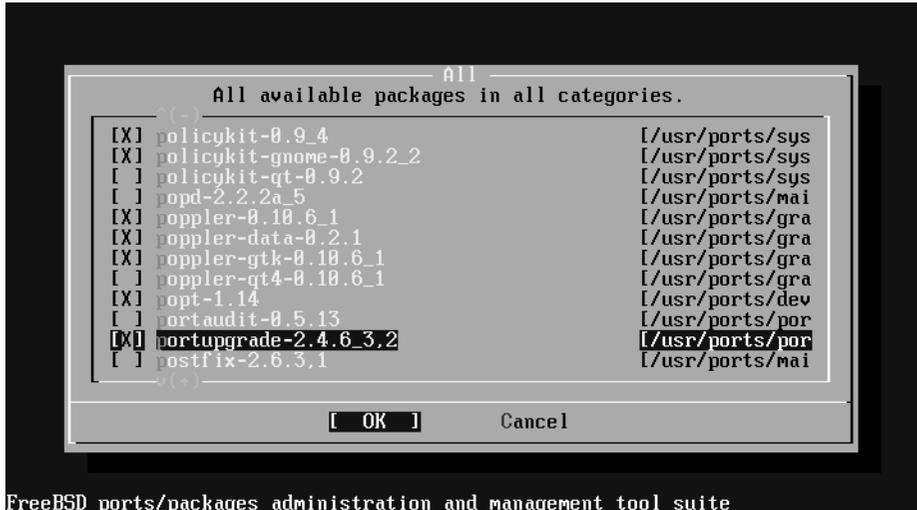


Рис. 10.13. Установка утилиты `portupgrade`

```
ath /usr/local/share/gdm/gconf.path
install -o root -g wheel -m 444 /usr/ports/x11/gdm/work/gdm-2.26.1/data/session
-setup.entries /usr/local/share/gdm/session-setup.entries
install -o root -g wheel -m 444 /usr/ports/x11/gdm/work/gdm-2.26.1/data/gdm.con
f-custom /usr/local/etc/gdm/custom.conf.default
/bin/mkdir -p /usr/local/share/xsessions
install -o root -g wheel -m 444 /usr/ports/x11/gdm/work/gdm.pam /usr/local/etc/
pam.d/gdm
GDM is installed.

Do NOT use /etc/ttys to start gdm at boot time. This will result in gdm
hanging or restarting constantly. Instead, add gdm_enable="YES" to
/etc/rc.conf. GDM will be started automatic on the next reboot.

==> Installing rc.d startup script(s)
==> Running ldconfig
/sbin/ldconfig -m /usr/local/lib
==> Registering installation for gdm-2.26.1_7
==> Cleaning for gmake-3.81_3
==> Cleaning for intltool-0.40.6
==> Cleaning for gdm-2.26.1_7
--> Cleaning out obsolete shared libraries
[Updating the pkgdb <format:bdb_btree> in /var/db/pkg ... - 488 packages found (
-0 +1) . done]
den# █
```

Рис. 10.14. Пакет `gdm` успешно обновлен

После обновления всех интересующих вас пакетов введите следующую команду.

```
# /usr/local/sbin/portsclean
```

Данная команда удаляет весь мусор, образовавшийся в результате компилирования программ из портов.

## 10.8. Обновление портов

Сами порты тоже нужно обновлять, иначе в них просто не будет смысла. Вы можете установить все имеющиеся обновления из портов, но поскольку вы не обновляете сами порты, то из “ниоткуда” новые программы в портах не появятся. Для обновления портов используется инструмент CVSup. Этот инструмент не установлен по умолчанию, поэтому лучше всего раздобыть его пакет (cvsup) и установить его с помощью команды `pkg_add`, поскольку сборка этого пакета из портов займет много времени. После того как `cvsup` установлен, нужно отредактировать его конфигурационный файл `/usr/local/etc/ports-supfile` так.

```
*default host=cvsup12.FreeBSD.org
*default base=/usr
*default prefix=/usr
*default release=cvs
*default tag=.
*default delete use-rel-suffix
ports-all
```

После этого запустите процесс обновления портов с помощью команды.

```
# /usr/local/bin/cvsup -L 2 /usr/local/etc/ports-supfile
```

Параметр `-L` задает уровень детализации выводимой информации. 2 — это самый высокий уровень, но вы можете вообще не указывать параметр `-L`.

Если вы хотите постоянно поддерживать вашу систему в актуальном состоянии, рекомендуется обновлять порты не реже, чем раз в неделю. После обновления портов нужно запустить утилиту `pkg_version` для поиска пакетов, нуждающихся в обновлении.

## 10.9. Резюме

Цель данного раздела — систематизировать полученные знания. Итак, в FreeBSD есть два способа установки программ — *из источника пакетов*, что мы и сделали с помощью конфигууратора `sysinstall`, и *из портов*. Конфигуратор `sysinstall` устанавливает из источника пакетов уже откомпилированные программы. Все, что нужно, — это распаковать пакет, и программа готова к работе. Этим и занимается `sysinstall`.

Порты удобнее использовать для обновления системы. Устанавливать программы из портов дело неблагодарное — на установку одной программы уходит много времени — ведь нужно загрузить ее исходный код, распаковать его и откомпилировать.

Если вы не хотите, чтобы установка программ занимала вечность, тогда установите необходимые вам пакеты с помощью `sysinstall`. А затем, когда появятся новые версии пакетов и при наличии у вас свободного времени и желания, можно обновить некоторые пакеты с помощью утилиты `portupgrade`. После обновления порта не забудьте выполнить команду `portsclean` для освобождения дискового пространства.