

Краткий справочник по Debian

Osamu Aoki <osamu\#at\#debian.org>
Перевод Ильи В. Головки <qref\#at\#yandex.ru>
'Authors' на стр. **27**

CVS, Чтв 18. Янв 2007, 11:54:40 UTC

Аннотация

Этот Краткий справочник по Debian (<http://qref.sourceforge.net/>) представляет краткое введение в систему Debian в форме Краткого справочника. Содержит выдержки из Справочник по Debian (<http://qref.sourceforge.net/>).

Сведения об авторских правах

Copyright (c) 2001–2005 by Osamu Aoki <osamu#at#debian.org>.

Этот документ может использоваться на условиях лицензии GNU General Public License версии 2 или выше. (<http://www.gnu.org/copyleft/gpl.html>)

Разрешается создавать и распространять точные копии этого документа с указанием лицензии и с этой заметкой о разрешениях, сохраняемой на всех копиях.

Разрешается копировать и распространять модифицированные версии этого документа на условиях для точного копирования только, если вся итоговая произведенная работа распространяется на условиях лицензии, идентичной этой.

Разрешается копировать и распространять переводы этого документа на другой язык на выше приведенных условиях для модифицированных версий, за исключением того, что эта заметка о разрешениях может включаться в переводы вместо оригинальной английской версии в случае одобрения ее Фондом Свободного Обеспечения.

Оглавление

1	Введение	1
1.1	Соглашения, принятые в документе	1
1.2	Основы дистрибуции Debian	1
2	Обновление до дистрибутива testing	3
2.1	Перевод системы АРТ к ее Woody-версии	3
2.2	Подготовка обновления (от stable к testing)	3
2.3	Обновление системы Debian	4
2.3.1	Самый лучший способ обновления, использующий программу dselect	5
3	Управление пакетами в Debian	7
3.1	Введение	7
3.1.1	Основные инструменты	7
3.1.2	Удобные инструменты	8
3.2	Основы управления пакетами в Debian	8
3.2.1	Установка задачи при помощи программ tasksel или aptitude	8
3.2.2	Установка системы АРТ - нет перевода!!!!	9
3.2.3	Программа dselect	9
3.2.4	Программа aptitude	10
3.2.5	Команды apt-cache и apt-get	11
3.2.6	Tracking a flavor of Debian distribution	12
3.2.7	Откат всех пакетов до уровня stable	12
3.2.8	Обзор файла /etc/apt/preferences	13
3.3	Команды выживания в системе Debian	14

3.3.1	Программные ошибки в Debian и поддержка	14
3.3.2	Разрешение проблем APT upgrade	14
3.3.3	Восстановление через dpkg	15
3.3.4	Восстановление информации о выбранных пакетах	16
3.3.5	Восстановление системы после сбоя каталога /var	16
3.3.6	Установка пакета в систему, которая не загружается	16
3.3.7	Что делать, если команда dpkg не работает	17
3.4	Нирвана в командах Debian	17
3.4.1	Информация о файле	17
3.4.2	Информация о пакете	18
3.4.3	Автоматическая установка при помощи APT	19
3.4.4	Повторное конфигурирование установленных пакетов	19
3.4.5	Операции remove и purge с пакетами	20
3.4.6	Удержание старых пакетов	20
3.4.7	Система на смеси stable/testing/unstable	20
3.4.8	Удаление файлов кэшированных пакетов	21
3.4.9	Запись/копирование конфигурации системы	21
3.4.10	Портирование пакета в систему на базе stable	21
3.4.11	Local package archive	22
3.4.12	Convert or install an alien binary package	23
3.4.13	Verify installed package files	23
3.5	Other Debian peculiarities	23
3.5.1	The dpkg-divert command	23
3.5.2	The equivs package	24
3.5.3	Alternative commands	24
3.5.4	System-V init and runlevels	25
3.5.5	Отключение неиспользуемых сервисов	25
A	Appendix	27
A.1	Authors	27
A.2	Warranties	29
A.3	Feedback	29

Глава 1

Введение

Коротко и ясно (KISS) - принцип, которым я руководствовался.

1.1 Соглашения, принятые в документе

Этот “Краткий справочник по Debian” предоставляет информацию путем коротких скриптов для командного интерпретатора Bash.

Ссылки на:

- Unix man-страницу даются в форме `bash(1)`.
- страницу GNU TEXINFO даются в форме `info libc`.

1.2 Основы дистрибуции Debian

Релизы Debian бывают трех видов :

- `stable` (стабильный) : Хорош для установки на рабочий сервер. Скучноват для рабочей станции (WS).
- `testing` (тестируемый): Хорош для установки на рабочую станцию.
- `unstable` (нестабильный): Не стоит слепо устанавливать.

Изучайте ключевой лист рассылки `debian-devel-announce@lists.debian.org` на предмет изменения статуса Debian.

В марте 2002 этим трем версиям релизов соответствовали кодовые имена Potato (рабочий), Woody (бета-тест, достаточно стабилен при приближении к выпуску), и Sid (альфа-тест). В августе 2002 после выхода Woody эти имена релизов стали соответствовать Woody (рабочий), Sarge (бэ́та-тест, достаточно современен некоторое время), и Sid (вечный альфа-тест).

Когда пакеты в `unstable` не имеют критических для выпуска (RC) ошибок в течение недели или больше, то они автоматически переходят в `testing`.

Существует два подхода, чтобы позволить вам работать на самых последних версиях ПО Debian:

- ‘Обновление до дистрибутива `testing`’ на стр. [3](#) (в основном для рабочих станций)
- ‘Портирование пакета в систему на базе `stable`’ на стр. [21](#) (в основном для серверов)

Глава 2

Обновление до дистрибутива testing

2.1 Перевод системы АРТ к ее Woody-версии

Если Вы все еще работаете в Potato, то возможно следующее обновление по сети системы АРТ и некоторых важных пакетов после включения в файл `/etc/apt/sources.list` ссылок на дистрибутив stable:

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

2.2 Подготовка обновления (от stable к testing)

Перед использованием Debian-дистрибутива testing нужно знать, что заплатки безопасности выходят для него очень медленно.

Обновление по сети до testing может быть выполнено следующим образом (или можно запустить скрипт `go-woody` (<http://www.debian.org/doc/manuals/debian-reference/examples/>)):

Очистите существующий файл `sources.list`:

```
# cd /etc/apt
# cp -f sources.list sources.old
# :>sources.list
```

Получите чистый список репозитариев для stable:

```
# cd /
# apt-setup noprobe
... выберите репозитарии, доступные по HTTP или FTP
```

Добавьте секцию testing к новому списку. Закомментируйте строки с deb-src.

```
# cd /etc/apt
# grep -e "^deb " sources.list >sources.deb
# grep -e "^deb-" sources.list >sources.src
# sed -e "s/stable/testing/" sources.deb \
    >>sources.list
# sed -e "s/stable/testing/" sources.src | \
    sed -e "s/^deb-/#deb-/" >>sources.list

# apt-get update
# apt-get install apt apt-utils
# cat >preferences <<EOF
Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50

EOF
```

Опционально, добавьте секцию “unstable”.

```
# sed -e "s/stable/unstable/" sources.deb \
    >>sources.list
# sed -e "s/stable/unstable/" sources.src | \
    sed -e "s/^deb-/#deb-/" >>sources.list
```

Искусство настройки файлов `/etc/apt/sources.list` и `/etc/apt/preferences` можно постичь, изучив раздел ‘Основы управления пакетами в Debian’ на стр. 8.

Теперь Вы готовы к обновлению, сделав это одним из методов следующей секции.

2.3 Обновление системы Debian

После соответствующей установки файлов `/etc/apt/sources.list` и `/etc/apt/preferences` система может быть обновлена до testing. Обращайтесь за базовыми знаниями к разделу ‘Управление пакетами в Debian’ на стр. 7 и к разделу ‘Разрешение проблем APT upgrade’ на стр. 14, если возникли трудности.

2.3.1 Самый лучший способ обновления, использующий программу dselect

Если в системе множество пакетов вида -dev и пр., рекомендуется следующий способ обновления для точного управления пакетами, использующий dselect:

```
# dselect update # всегда выполняется перед обновлением
# dselect select # выберите дополнительные пакеты
```

При запуске dselect выбираются все Ваши текущие пакеты. Программа dselect может предложить Вам установить дополнительные пакеты, основываясь на зависимостях depends, suggests и recommends пакетов. Если Вы не желаете добавлять никакие пакеты, просто нажмите Q для выхода dselect.

```
# dselect install
```

На этой стадии процесса Вам нужно будет ответить на некоторые вопросы, касаемые конфигурации пакетов, поэтому подготовьте Ваши записи и затратьте на это некоторое количество времени. Смотрите ‘Программа dselect’ на стр. 9.

Используйте программу dselect. Она всегда работает :) Если Вам нужно обновиться без использования dselect в системах новее Woody, рассмотрите программу aptitude и прочие.

Глава 3

Управление пакетами в Debian

Эта глава основывается на более старой версии текста английского оригинала. Проверьте английский вариант тоже.

Чтобы снизить нагрузку на репозитории Debian по сети, установите локальный HTTP прокси-сервер squid для кэширования пакетов, скачиваемых системой АРТ, и настройте переменную окружения `http_proxy` или задайте значение параметру `http` в файле `/etc/apt/apt.conf` (для того, чтобы АРТ использовала этот прокси-сервер - прим. переводчика). Эти меры существенно улучшат производительность сетевых обновлений, особенно, если Вы имеете несколько Debian-машин в сети.

Although the pinning feature of `apt_preferences(5)` is powerful, it does not solve all the dependency issues since dependency requirements tend to pull in newer versions of other fundamental program packages.

Использование `chroot`-метода предпочтительно для обеспечения стабильности системы и одновременно доступа к последним версиям программного обеспечения.

Хотя эта глава предназначена для использования на Woody-версии Debian, большая часть информации также применима к Potato-версии (за исключением `apt_preferences(5)` и разделов, связанных с `/etc/apt/preferences`).

3.1 Введение

Если чтение всей документации разработчиков слишком утомительно для Вас, то сначала прочтите эту главу и потом наслаждайтесь всей мощью системы Debian, используя дистрибутивы `testing/unstable :-)`.

3.1.1 Основные инструменты

`dselect` – инструмент для управления пакетами с меню-подобным интерфейсом (высокоуровневый инструмент)

`dpkg` – инсталлятор пакетов (основная утилита для работы с файлами-пакетами)
`apt-get` – инсталлятор пакетов (утилита для работами с репозиториями пакетов, интерфейс командной строки к системе АРТ)
`tasksel` – инсталлятор задач (набора пакетов)
`aptitude` – инсталлятор пакетов (пакеты & задачи, интерфейс к АРТ на базе ncurses)
`deity` – альтернатива ncurses-интерфейса к АРТ
`synaptic`, `gsynaptic` – графические утилиты для работы с АРТ

Данные инструменты принадлежат разным уровням работы с пакетами. `dselect` работает над системой АРТ (команда `apt-get`) и программой `dpkg`.

АРТ использует файлы `/var/lib/apt/lists/*` для отслеживания доступных пакетов, в то время как `dpkg` для этих целей использует файл `/var/lib/dpkg/available`. Если Вы установили пакеты непосредственно используя программу `apt-get` или подобными средствами (`aptitude`), то перед тем, как запускать команды `dselect select`, `tasksel` или `dpkg -l`, не забудьте обновить файл `/var/lib/dpkg/available` через пункт меню [U]pdate программы `dselect` или, запустив в командной строке команду “`dselect update`”.

Так как пакеты имеют зависимости, программа `apt-get` автоматически при установке пакета выбирает к установке дополнительные пакеты, от которых устанавливаемый пакет зависит (`depends`), но она не трогает пакеты, связанные с ним зависимостями “рекомендует” (`recommends`) и “предлагает” (`suggests`). В то же время программа `dselect` через меню предлагает установить пакеты, основываясь на зависимостях вида “зависит” (`depends`), “рекомендует” (`recommends`) и “предлагает” (`suggests`). Программа `aptitude` имеет опцию для автоматической установки всех пакетов, выбираемых этими видами зависимостей.

3.1.2 Удобные инструменты

`apt-cache` – проверка локального кэша пакетов
`dpkg-reconfigure` – повторная конфигурация уже установленного пакета (если он использует `debconf`)
`dpkg-source` – управляет исходными файлами пакета
`dpkg-buildpackage` – автоматизирует сборку пакета
...

3.2 Основы управления пакетами в Debian

Вы можете установить набор пакетов, называемых задачами, установить пакеты по одному или обновить систему, используя инструменты управления пакетами, описанные ниже .

3.2.1 Установка задачи при помощи программ `tasksel` или `aptitude`

Программа `tasksel` является Дебиановским Установщиком Задач, который используется при установке системы в упрощенном “simple” режиме.

Когда Вам в системе нужна некоторая функциональность, которая реализуется множеством пакетов, использование `tasksel` - самый лучший способ это сделать. Последовательность команд должна быть такой:

```
# dselect update
# tasksel
```

Программа `aptitude` также предлагает доступ к задачам. Но она не только дает Вам возможность выбрать для установки задачу, но и позволяет выборочно через меню отменить установку некоторых пакетов из состава задачи.

3.2.2 Установка системы APT - нет перевода!!!!

If you try to track mixed environment as described here, you may likely to hit some package dependency conflicts. It is good idea not to mix flavors. Followings are for people who is willing to experiment knowing some risks.

For selective upgrade while tracking the testing distribution, the APT system (>Woody) must be set up as in ‘Перевод системы APT к ее Woody-версии’ на стр. 3 to use `apt_preferences(5)` features.

First, add the sources for stable, testing, and unstable to your `/etc/apt/sources.list`. Then, edit `/etc/apt/preferences` to set the proper `Pin-Priority`.¹

```
Package: *
Pin: release a=stable
Pin-Priority: 500
```

```
Package: *
Pin: release a=testing
Pin-Priority: 600
```

```
Package: *
Pin: release a=unstable
Pin-Priority: 50
```

3.2.3 Программа dselect

После старта программа `dselect` автоматически выбирает все пакеты с приоритетами “Required” (требующиеся), “Important” (важные) и “Standard” (стандартные). В системе Potato некоторые большие пакеты вроде `teTeX` или `Emacs` попадали в этот список, и самым

¹I kept the description around here as is to maintain consistency with other parts. If you are really tracking testing or unstable, you should actually remove references to stable in `/etc/apt/sources.list` and `/etc/apt/preferences`. This is because testing starts as a copy of stable.

лучшим был их пропуск при начальной установке системы путем ручного снятия отметки соответствующих пакетов (нажатие клавиши ‘_’). В Woody же эти пакеты уже помечены приоритетом “Optional” (опициональный).

Программа dselect имеет несколько необычный интерфейс пользователя. Существует четыре несколько странных команды (регистр букв важен!):

Клавиша	Действие
Q	Выход. Подтвердить текущий выбор пакетов и выйти из программы. (не принимать во внимание зависимости)
R	Вернуть все назад! Я не имел это ввиду.
D	Проклятье! Мне не волнует, что думает dselect. Просто сделай это!
U	установить все пакеты как рекомендуемые

При помощи D и Q Вы можете выбрать конфликтующие пакеты (на свой страх и риск). С этими командами следует обращаться осторожно. В настоящее время dselect является достаточно продвинутым инструментом с меню-интерфейсом, который предлагает тонкий контроль над выбором пакетов, в частности, по зависимостям suggests (предлагает) и recommends (рекомендует).

Для снижения “разговорчивости” программы добавьте в файл /etc/dpkg/dselect.cfg строку с опцией “expert”. Для медленных машин Вы можете запускать dselect на другой более быстрой машине для поиска нужных пакетов, а на медленной машине уже использовать apt-get install для их установки.

dselect не дает доступ к пакетам, которые не preferred by the Pin-Priority.

3.2.4 Программа aptitude

aptitude - это новый инсталлятор пакетов с меню-интерфейсом, похожий на dselect. Он также может использоваться как альтернативна консольной команде apt-get. Изучите страницу aptitude(1).

Программа aptitude обрабатывает одно-клавишные команды, вводимые в нижнем регистре

Клавиша	Действие
F10	Меню
?	Помощь по клавишам управления
u	Обновить информацию об доступных пакетах в репозитории
g	Скачать и установить выбранные пакеты
q	Выйти из текущего экрана с сохранением изменений
x	Выйти без сохранения изменений
Enter	Просмотр информации о пакете

Программа aptitude опционально может автоматически устанавливать пакеты, основываясь на зависимостях не только depends, но recommends и suggests. Вы можете изменять ее поведение, выбрав в меню F10 -> Options -> Dependency handling.

Программа aptitude предлагает доступ ко всем версиям пакета.

3.2.5 Команды apt-cache и apt-get

While tracking testing as described in the above example, мы можем управлять системой следующими командами:

- `apt-get -u upgrade`
This tracks the testing distribution and upgrades all the packages on the system while installing their dependencies from testing.
- `apt-get -u dist-upgrade`
This tracks the testing distribution and upgrades all the packages on the system while installing and resolving their dependencies from testing.
- `apt-get -u dselect-upgrade`
This tracks the testing distribution and upgrades all the packages on the system according to the selections of dselect.
- `apt-get -u install package`
This installs package and its dependencies from the testing distribution.
- `apt-get -u install package/unstable`
Устанавливает пакет `package` из `unstable`-дистрибутива, при этом пакеты, от которых он зависит, берутся из `testing`-дистрибутива.
- `apt-get -u install -t unstable package`
Устанавливает пакет `package` из `unstable`-дистрибутива, при этом, пакеты от которых он зависит, также берутся из `unstable` при помощи установки `Pin-Priority unstable` в `990`.
- `apt-cache policy foo bar ...`
Проверяет статус пакетов `foo bar ...`.
- `apt-cache show foo bar ... | less`
Проверяет информацию по пакетам `foo bar ...`.
- `apt-get install foo=2.2.4-1`
Устанавливает конкретную версию `2.2.4-1` пакета `foo`.
- `apt-get -u install foo bar-`
Устанавливает пакет `foo` и удаляет пакет `bar`.
- `apt-get remove bar`
Удаляет пакет `bar`, но не трогает его конфигурационные файлы.
- `apt-get remove --purge bar`
Удаляет пакет `bar` вместе со всеми его конфигурационными файлами.

В приведенных выше примерах использование опции `-u` гарантирует, что `apt-get` выведет список всех пакетов, подлежащих обновлению, и запросит подтверждение у пользователя перед началом своих действий. Вот так можно установить опцию `-u` в режим по умолчанию:

```
$ cat >> /etc/apt/apt.conf << .  
// Всегда показывать пакеты для обновления (-u)  
APT::Get::Show-Upgraded "true";  
.
```

Используйте опцию `-s` для имитации обновления без выполнения каких-либо действительных операций.

3.2.6 Tracking a flavor of Debian distribution

Depending on your preference for a flavor of Debian to track, you can change the sample `/etc/apt/preferences` in ‘Установка системы АРТ - нет перевода!!!!’ на стр. 9 to fit your needs:

```
track stable:           change Pin-Priority of testing to 50  
track testing:         keep settings as shown above  
track testing(unstable): change Pin-Priority of unstable to 500  
track unstable(testing): change Pin-Priority of unstable to 800
```

A guideline for the choice of Pin-Priority is to move from the top to bottom in the above table as the time moves from a time immediately after a distribution release to a time of freeze for the next release.

Caution: tracking the testing flavor of Debian has a side effect of providing very slow security fixes.

If you mix flavors of Debian, testing with stable or unstable with stable, you will be likely to pull in unintentionally key packages from testing or unstable that may be buggy. So be warned.

Examples of `/etc/apt/preferences` which lock some key packages to the more mature version while tracking the less mature version for other nonessential packages are available in the examples subdirectory (<http://www.debian.org/doc/manuals/debian-reference/examples/>) as `preferences.testing` and `preferences.unstable`. On the other hand, `preferences.stable` forces all packages to be downgraded to stable.

3.2.7 Откат всех пакетов до уровня stable

Чтобы откатить все пакеты до уровня `stable`, отредактируйте файл `/etc/apt/preferences` следующим образом:

```
Package: *  
Pin: release a=stable  
Pin-Priority: 1001
```


и запустите команду `apt-get upgrade`, которая вызовет принудительный откат из-за значения `Pin-priority > 1000`. Нужно предупредить, что могут возникнуть небольшие накладки с зависимостями.

3.2.8 Обзор файла `/etc/apt/preferences`

В примере файла `/etc/apt/preferences`, представленного в разделе ‘Установка системы АРТ - нет перевода!!!!’ на стр. 9, строка `Package: *` в первой строчке означает, что эта секция применяется ко всем пакетам. Вы можете использовать имя пакета вместо `*` для указания значения `Pin-Priority` этого пакета.

Во второй строчке запись `Pin: release a=stable` означает, что `apt-get` получает файл `Package.gz` из архива Debian, у которого в файле `Release` есть строка `Archive: stable`.

В последней строчке запись `Pin-Priority: 500` указывает значение `Pin-Priority` равное 500.

Если существует несколько пакетов с одним и тем же именем, устанавливается пакет с наибольшим значением `Pin-Priority`

В частности, значения `Pin-Priority` имеют следующий смысл:

- 1001 и выше: Downgradable priorities.
Пакеты с `Pin-Priority` из этого диапазона могут быть понижены в версии.
- от 100 до 1000: Стандартные приоритеты.
Такие пакеты нельзя понижать в версии. Вот несколько ключевых значений `Pin-Priority`:
 - 990: приоритет установлен через `—target-release` или опцию `-t`. Подробнее на странице `apt-get(8)`.
 - 500: приоритет всех пакетов по умолчанию.
 - 100: приоритет установленных пакетов в настоящее время.
- 0 - 99: неавтоматические приоритеты. (Они используются только в случае, если пакет не установлен и нет другой доступной версии пакета.)
- менее нуля: Версия пакета никогда не выбирается.

Эффект подобный от `—target-release` может быть получен от установки файла `/etc/apt/apr.conf` следующим образом:

```
# echo 'АРТ::Default-Release "testing";' >> /etc/apt/apr.conf
```

Опция `—target-release` в командой строке и установки в файле `/etc/apt/apr.conf` имеют больший приоритет установок в файле `/etc/apt/preferences`. Будьте внимательны с этим побочным эффектом при работе с файлом `/etc/apt/preferences`.

3.3 Команды выживания в системе Debian

Обладая этими знаниями, Вы можете пребывать в вечном "обновлении" :-)

3.3.1 Программные ошибки в Debian и поддержка

Если у Вас возникли проблемы с определенным пакетом, исследуйте следующие сайты перед тем, как обращаться за помощью или оформлять отчет об ошибке. (страницы одинаково хорошо будут выглядеть в браузерах lynx, links и w3m):

```
$ lynx http://bugs.debian.org/  
$ lynx http://bugs.debian.org/имя-пакета # если известно имя пакета  
$ lynx http://bugs.debian.org/номер-ошибки # если известен номер ошибки
```

Зайдите на Google (www.google.com) для поиска слов, включив фразу "site:debian.org".

Если что-то не ясно, изучите замечательное руководство. Установите переменную окружения CDPATH следующим образом:

```
export CDPATH=./usr/local:/usr/share/doc
```

и наберите

```
$ cd ИмяПакета  
$ pager README.Debian # если файл существует  
$ mc
```

3.3.2 Разрешение проблем APT upgrade

Проблемы с зависимостями пакетов могут случаться при обновлениях в unstable/testing, описанных в 'Обновление системы Debian' на стр. 4. Чаще всего это случается потому, что обновляемый пакет имеет зависимости, которые нельзя удовлетворить. Подобные проблемы решаются так:

```
# apt-get dist-upgrade
```

Если это не работает, то повторяйте одну из следующих команд до разрешения проблемы:

```
# apt-get upgrade -f # продолжение обновления (upgrade) даже после возникнове-  
ния ошибки  
... или  
# apt-get dist-upgrade -f # продолжение обновления дистрибутива (dist-upgrade) даже по-  
сле возникновения ошибки
```

Некоторые действительно неправильные скрипты могут привести к постоянной проблеме. Обычно лучше разрешать подобные ситуации, изучив файлы `/var/lib/dpkg/info/packageName.{post,pre}{inst,rm}` скриптов сбойного пакета и затем запустить команду:

```
# dpkg --configure -a # настраивает все ненастроенные пакеты
```

Если скрипт жалуется на отсутствие какого-нибудь конфигурационного файла, поищите его в `/etc`. Если он существует, но с расширением `.new` (или подобным), измените его имя на имя без этого расширения (команда `mv`).

Проблема зависимостей пакетов может возникать при установке пакета в смесь `unstable/testing`. Существует несколько путей обойти зависимости.

```
# apt-get install -f package # не принимать во внимание сломанные зависимости
```

Альтернативный метод исправить эту ситуацию заключается в использовании пакета `equivs`. Смотрите `/usr/share/doc/equivs/README.Debian`.

3.3.3 Восстановление через `dpkg`

В крайних случаях восстановление неработоспособной программы `dselect` (APT) на серьезно поврежденной системе может быть выполнено при помощи программы `dpkg` без использования APT:

```
# cd /var/cache/apt/archives
# dpkg -i libc6* libdb2* perl*
# dpkg -i apt* dpkg* debconf*
# dpkg -i * # до тех пор, пока ошибки не перестанут появляться
```

Если какого-либо пакета нет, получите его с используя:

```
# mc # используйте "FTP-соединение", указав Debian FTP-сервер
```

Стоит заметить, что с недавнего времени сами файлы пакетов на HTTP/FTP-сервере располагаются в новом каталоге `/pool`, а не в классическом месте - каталоге `/dist`.

Затем устанавливаем пакеты:

```
# dpkg -i /var/cache/apt/archives/packagefile.deb
```

Сломанные зависимости или исправляем, или поступаем таким образом:

```
# dpkg --ignore-depends=пакет1,... -i имя_файла_пакета.deb
# dpkg --force-depends -i имя_файла_пакета.deb
# dpkg --force-depends --purge пакет
# dpkg --force-confmiss -i имя_файла_пакета.deb # Установка отсутствующих conf-файлов
```

3.3.4 Восстановление информации о выбранных пакетах

Если файл `/var/lib/dpkg/status` оказался поврежден по какой-либо причине, система Debian теряет информацию о выбранных пакетах и попадает в затруднительное положение. Ищите старую версию файла `/var/lib/dpkg/status` под именем `/var/lib/dpkg/status-old` или `/var/backups/dpkg.status.*`.

Неплохой идеей является размещение каталога `/var/backups/` на отдельном разделе жесткого диска, так как он содержит массу важных системных данных.

Если старый файл `/var/lib/dpkg/status` недоступен, Вы все же можете восстановить потерянную информацию из каталогов из `/usr/share/doc/`:

```
# ls /usr/share/doc | \
grep -v [A-Z] | \
grep -v '^texmf$' | \
grep -v '^debian$' | \
awk '{print $1 " install"}' | \
dpkg --set-selections
# dselect --expert # reinstall system, de-select as needed
```

3.3.5 Восстановление системы после сбоя каталога `/var`

Так как каталог `/var` содержит регулярно обновляемые данные, такие как почта, он подвержен разного рода порче. Выделение его на независимый раздел жесткого диска ограничивает риск. Если сбой все же произошел, то для восстановления Вашей Debian системы Вам понадобится восстановить и этот каталог.

Получите скелет содержимого каталога `/var` от минимальной рабочей системы Debian, основанной на той же или более древней версии Debian, например такой, `var.tar.gz` (<http://people.debian.org/~osamu/pub/>), и поместите его в корневой каталог сломанной системы. Далее:

```
# cd /
# mv var var-old # если все же что-то полезное осталось
# tar xvzf var.tar.gz # используйте скелет от Woody
# aptitude # или dselect
```

Это должно воссоздать рабочую систему. Вы можете облегчить восстановление информации о выбранных пакетах, используя прием из раздела ‘Восстановление информации о выбранных пакетах’ на стр. 16. ([FIXME]: This procedure needs more experiments to verify.)

3.3.6 Установка пакета в систему, которая не загружается

Загрузите Linux со спасательного диска/дискеты Debian или альтернативного раздела мультизагрузочной Linux-системы. Подмонтируйте незагружающуюся систему к каталогу `/target` и используйте у `dpkg chroot`-режим установки.

```
# dpkg --root /target -i имя_пакета.deb
```

Затем настройте пакеты и исправьте проблемы.

Между прочим, если причиной нормальной загрузки является нарушенный загрузчик lilo, Вы можете загрузить систему с использованием стандартного спасательного диска Debian. На загрузочное приглашение boot: введите (подразумевается, что корневой раздел Вашей Linux-системы - /dev/hda12, что нужно войти на третий уровень исполнения системы):

```
boot: rescue root=/dev/hda12 3
```

После этого Вы загружаете практически полностью функционирующую системы на ядре с гибкого диска. (возможно, при загрузке будут некоторые проблемы с отсутствием поддержки некоторых возможностей в ядре или каких-либо модулей).

3.3.7 Что делать, если команда dpkg не работает

Неработоспособность программы dpkg приводит к невозможности установки любого .deb файла. Следующая процедура поможет Вам выйти из подобной ситуации. (в следующей команде можно заменить "links" на Ваш любимый браузер)

```
$ links http://http.us.debian.org/debian/pool/main/d/dpkg/  
... скачиваем пакет dpkg_версия_архитектура.deb  
$ su  
password: *****  
# ar x dpkg_version_arch.deb  
# mv data.tar.gz /data.tar.gz  
# cd /  
# tar xzfv data.tar.gz
```

Для архитектуры i386 также можно использовать следующий адрес:
<http://packages.debian.org/dpkg>.

3.4 Нирвана в командах Debian

Осознавшие смысл этих команд спасутся от вечных мучений ада обновлений и достигнут нирваны в Debian.

3.4.1 Информация о файле

Чтобы найти пакет, к которому некий файл принадлежит, сделайте:

```
$ dpkg {-S|--search} образец # найти слово образец в установленных пакетах
$ zgrep -e образец /local/copy/of/debian/woody/Contents-i386.gz
#... найти образец (имя файла) в архиве Debian
```

Или используйте специально предназначенную для этого команду:

```
# apt-get install dlocate
$ dlocate имя_файла # более быстрая альтернатива командам dpkg -L и dpkg -S
...
# apt-get install auto-apt # инструментарий для установки пакетов по запросу
# auto-apt update # создать базу данных для auto-apt
$ auto-apt search образец # поиск образца во всех пакета, установленных и нет.
```

3.4.2 Информация о пакете

Поиск и вывод информации из архива пакетов. Убедитесь, что система АРТ настроена на соответствующий архив(файл `/etc/apt/sources.list`). Если Вы хотите увидеть, как пакеты установлены из дистрибутивов `testing/unstable`, пользуйтесь командой `apt-cache policy`— очень удобно.

```
# apt-get check # обновить кэш и проверить его на предмет сломанных пакетов
$ apt-cache search образец # поиск пакета по текстовому образцу
$ apt-cache policy образец # информация по пакету priority/dists
$ apt-cache show -a образец # показать описание пакета во всех дистрибутивах
$ apt-cache showsrc образец # показать описание пакетов с исходным текстом
$ apt-cache showpkg образец # отладочная информация по пакету
# dpkg --audit|-C # поиск неполностью установленных пакетов
$ dpkg {-s|--status} пакет ... # описание установленного пакета
$ dpkg -l пакет ... # вывести статус установленного пакета (по одной строчке на пакет)
$ dpkg -L пакет ... # вывести список файлов, установленных пакетом
```

Команда `apt-cache showsrc` не заявлена для релиза Woody, но она работает:)

Вы также можете найти информацию по пакету в следующих файлах (я использую `ms` для их просмотра):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Сравнение следующих файлов дает информацию о том, что же произошло в последний сеанс установки.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

3.4.3 Автоматическая установка при помощи АРТ

Для автоматической установки добавьте следующую строку в файл `/etc/apt/apt.conf`:

```
Dpkg::Options {"--force-confold";}
```

Это эквивалентно запуску команды `apt-get -q -у packagename`. Так как на все вопросы дается утвердительный ответ "да", то иногда это может вызвать проблемы, поэтому используйте эту возможность с осторожностью. Смотрите страницы `apt.conf(5)` и `dpkg(1)`.

Позже Вы можете повторно конфигурировать любой конкретный пакет путем 'Повторное конфигурирование установленных пакетов' на стр. 19.

3.4.4 Повторное конфигурирование установленных пакетов

Используйте следующие команды для повторного конфигурирования любого уже установленного пакета.

```
# dpkg-reconfigure --priority=medium пакет [...]
# dpkg-reconfigure --all # повторно конфигурировать все пакеты
# dpkg-reconfigure locales # генерировать дополнительные локали
# dpkg-reconfigure --p=low xserver-xfree86 # повторно настроить X-сервер
```

Do this for debconf if you need to change the debconf dialog mode permanently.

Некоторые программы идут со специальными настроенными скриптами.

```
apt-setup      - создает список /etc/apt/sources.list
install-mbr    - устанавливает менеджер главной загрузочной записи
tzconfig       - устанавливает временную зону
gpmconfig      - настроить демон мыши gpm
eximconfig     - настроить Exim (MTA)
texconfig      - настроить TeTeX
apacheconfig   - настроить Apache (httpd)
cvsconfig      - настроить CVS
sndconfig      - настроить звук
...
update-alternatives - set default command, e.g., vim as vi
update-rc.d     - System-V init script management
update-menus    - Debian menu system
...
```

3.4.5 Операции remove и purge с пакетами

Удаление пакета с сохранением его конфигурации:

```
# apt-get remove пакет ...
# dpkg --remove пакет ...
```

Удаление пакета, включая его конфигурацию:

```
# apt-get remove --purge пакет ...
# dpkg --purge пакет ...
```

3.4.6 Удержание старых пакетов

Например, удержание пакетов `libc6` и `libc6-dev` при работе с командами `dselect` и `apt-get -u upgrade package` может быть выполнено так:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

Однако, для команды `apt-get -u install пакет` этот "hold" не будет помехой. Для того, чтобы удержать пакет от обновления путем автоматического отката в командах `apt-get -u upgrade пакет` или `apt-get -u dist-upgrade`, добавьте следующие строки в файл `/etc/apt/preferences`:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

Здесь элемент "Package:" не может использовать запись вида "libc6*". Если Вам нужно удержать все бинарные пакеты, связанные с пакетом с исходными текстами `glibc` в синхронном состоянии, Вам нужно перечислить всех их здесь явным образом.

Следующая команда выведет всех удержанные от обновления пакеты:

```
dpkg --get-selections "*" | grep -e "hold$"
```

3.4.7 Система на смеси stable/testing/unstable

Команда `apt-show-versions` выводит список версий доступных пакетов по дистрибутивам.

```
$ apt-show-versions | fgrep /testing | wc
... как много пакетов мы имеет в testing
$ apt-show-versions -u
... список пакетов, подлежащих обновлению
$ apt-get install 'apt-show-versions -u -b | fgrep /unstable'
... обновляет пакеты из unstable на их более новые версии
```


3.4.8 Удаление файлов кэшированных пакетов

В процессе установки через АРТ в каталоге `/var/cache/apt/archives` кэшируются файлы пакетов, и их можно удалить.

```
# apt-get autoclean # удаляет только неиспользованные файлы пакетов
# apt-get clean    # удаляет все файлы из кэша
```

3.4.9 Запись/копирование конфигурации системы

Для создания локальной копии списка состояния выбранных пакетов сделайте:

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >мой_выбранные_пакеты # или используйте \* вместо *
```

Использование "*" приводит к помещению в файл `мой_выбранные_пакеты` записей о пакетах, подлежащие удалению ("purge").

Вы можете перенести этот файл на другой компьютер и установить по нему системы:

```
# dselect update
# debconf-set-selections < debconfsel.txt
# dpkg --set-selections <мой_выбранные_пакеты
# apt-get -u dselect-upgrade # или dselect install
```

3.4.10 Портирование пакета в систему на базе stable

Для частичного обновления системы на базе `stable` необходимо пересобрать пакет в окружении `stable` системы, используя соответствующий пакет с исходным текстом. Это позволит избежать большого количества обновлений пакетов, от которых устанавливаемый пакет зависит. Во-первых, добавьте следующие записи в файл `to /etc/apt/sources.list`:

```
deb-src http://http.us.debian.org/debian testing \
main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
main contrib non-free
```

Здесь каждая запись `deb-src` разбита на две строки для удобства показа, однако в файле `sources.list` такое разбиение на строки недопустимо.

Далее получаем исходный текст и собираем пакет:

```

$ apt-get update # обновляем кэш списка пакетов
$ apt-get source пакет
$ dpkg-source -x пакет.dsc
$ cd пакет-версия
... изучаем требующиеся пакеты (зависимости Build-depends в .dsc файле) и
  устанавливаем их также. Вам также понадобится пакет "fakeroot".

$ dpkg-buildpackage -rfakeroot

...или (без сигнатуры)
$ dpkg-buildpackage -rfakeroot -us -uc # если нужно, позже воспользуйтесь "debsign"

...Затем для установки пакетов выполняем
$ su -c "dpkg -i имя_файла_пакета.deb"

```

Обычно нужно установить несколько пакетов с суффиксом "-dev" для удовлетворения зависимостей пакета. Программа debsign находится в пакете devscripts. Программа auto-apt способна упростить удовлетворение таких зависимостей. Используйте программу fakeroot, чтобы не заниматься сборкой пакетов под root.

В Woody эти вопросы с зависимостями могут быть упрощены. Например, для компиляции пакета pine, который поставляется только в виде исходного текста, выполните:

```

# apt-get build-dep pine
# apt-get source -b pine

```

3.4.11 Local package archive

In order to create a local package archive which is compatible with APT and the dselect system, Packages needs to be created and package files need to be populated in a particular directory tree.

A local deb repository similar to an official Debian archive can be made in this way:

```

# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # physical packages are located here
# install -d dists/unstable/main/binary-i386
# ls -l pool | sed 's/_{.*}/ priority section/' | uniq > override
# editor override # adjust priority and section
# dpkg-scanpackages pool override /usr/local/ \
  > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
Archive: unstable
Version: 3.0

```

```
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local/unstable/main" \
  >> /etc/apt/sources.list
```

Alternatively, a quick-and-dirty local deb repository can be made:

```
# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /some/where/package.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
# echo "deb file:/usr/local/debian ./" >> /etc/apt/sources.list
```

These archives can be remotely accessed by providing access to these directories through either HTTP or FTP methods and changing entries in `/etc/apt/sources.list` accordingly.

3.4.12 Convert or install an alien binary package

`alien` enables the conversion of binary packages provided in Redhat rpm, Stampede slp, Slackware tgz, and Solaris pkg file formats into a Debian deb package. If you want to use a package from another Linux distribution than the one you have installed on your system, you can use `alien` to convert it to your preferred package format and install it. `alien` also supports LSB packages.

3.4.13 Verify installed package files

`debsums` enables verification of installed package files against MD5 checksums. Some packages do not have available MD5 checksums. A possible temporary fix for sysadmins:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

per Joerg Wendland <joergland@debian.org> (untested).

3.5 Other Debian peculiarities

3.5.1 The `dpkg-divert` command

File diversions are a way of forcing `dpkg` not to install a file into its default location, but to a diverted location. Diversions can be used through the Debian package scripts to move a file away

when it causes a conflict. System administrators can also use a diversion to override a package's configuration file, or whenever some files (which aren't marked as conffiles) need to be preserved by dpkg, when installing a newer version of a package which contains those files.

```
# dpkg-divert [--add] filename # add "diversion"
# dpkg-divert --remove filename # remove "diversion"
```

It's usually a good idea not to use dpkg-divert unless it is absolutely necessary.

3.5.2 The equivs package

If you compile a program from source, it is best to make it into a real local debianized package (*.deb). Use equivs as a last resort.

```
Package: equivs
Priority: extra
Section: admin
Description: Circumventing Debian package dependencies
 This is a dummy package which can be used to create Debian
 packages, which only contain dependency information.
```

3.5.3 Alternative commands

To make the command vi run vim, use update-alternatives:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
 Selection    Command
-----
      1      /usr/bin/elvis-tiny
      2      /usr/bin/vim
*+    3      /usr/bin/nvi
```

Enter to keep the default[*], or type selection number: 2

Items in the Debian alternatives system are kept in /etc/alternatives as symlinks.

To set your favorite X Window environment, apply update-alternatives to /usr/bin/x-session-manager and /usr/bin/x-window-manager.

/bin/sh is a direct symlink to /bin/bash or /bin/dash. It's safer to use /bin/bash to be compatible with old Bashism-contaminated scripts but better discipline to use /bin/dash to enforce POSIX compliance. Upgrading to a 2.4 Linux kernel tends to set this to /bin/dash.

3.5.4 System-V init and runlevels

The default runlevel to boot into can be set in `/etc/inittab`.

Unlike other distributions, Debian makes the management of runlevel completely the sysadmin's responsibility. Management of System-V style init on Debian is intended to be performed through `update-rc.d` scripts.

Starting `/etc/init.d/name` in runlevel 1,2,3 and stopping in 4,5 with sequencing priority number 20 (normal) can be done by:

```
# update-rc.d name start 20 1 2 3 . stop 20 4 5 .
```

Removing symbolic links while the script in `init.d` still exists can be done by:

```
# update-rc.d -f name remove
```

For editing runlevels, I cheat. I edit entries manually using the `mv` command at the shell prompt of `mc` while copying link entries using `Alt-Enter`. For example:

```
# mv S99xdm K99xdm # disable xdm (X display manager)
```

I even disable a daemon by inserting `exit 0` at the start of an `init.d` script as a quick hack. These are `conffiles` after all.

3.5.5 Отключение неиспользующихся сервисов

Дистрибутив Debian серьезно относится к вопросам безопасности и ожидает участие компетентного администратора. Поэтому иногда простота использования переходит на второй план, и многие сервисы идут с наивысшим уровнем безопасности, а сама система с минимальным набором работающих сервисов после установки по умолчанию.

Если Вы имеете какие-то сомнения относительно сервисов (Exim, DHCP, ...), запустите команду `ps aux` или проверьте содержимое файлов `/etc/init.d/*` и `/etc/inetd.conf`. Также проверьте файл `/etc/hosts.deny`. Очень полезная команда `pidof` (смотрите `pidof(8)`).

С недавних пор в Debian система X11 не позволяет удаленные TCP/IP подключения. Проброс трафика системы X Window в тунеле SSH также запрещена.

Приложение А

Appendix

A.1 Authors

Краткий справочник по Debian was initiated by Osamu Aoki <osamu\#at\#debian.org> as a personal installation memo that was eventually called "Quick Reference ...". Many contents came from the archives of the "debian-user" mailing list. Also "Debian Installation Manual" and "Debian Release Notes" were referenced.

Following a suggestion from Josip Rodin, who is very active with the Debian Documentation Project (<http://www.debian.org/doc/ddp>) (DDP) and is the current maintainer of "The Debian FAQ", this document was renamed as "Справочник по Debian" and was merged with several chapters from the "The Debian FAQ" with reference-like contents. Then "Краткий справочник по Debian" was formed as an excerpt.

This document has been edited, translated, and expanded by the following QREF team members:

- English originals for original "Quick Reference..."
 - Osamu Aoki <osamu\#at\#debian.org> (leader: all contents)
- English proofreading and additional contribution
 - David Sewell <dsewell\#at\#virginia.edu> (leader: en style)
 - Thomas Hood <jdthood\#at\#yahoo.co.uk> (network related)
 - Brian Nelson <nelson\#at\#bignachos.com> (especially X related)
 - Jan Michael C Alonzo <jmalonzo\#at\#spaceants.net>
 - Daniel Webb <webb\#at\#robust.colorado.edu>
 - Feedback from all translators
- French translation
 - Guillaume Erbs <gerbs\#at\#free.fr> (leader: fr)
 - Rénaud Casagraude <rcasagraude\#at\#interfaces.fr>
 - Jean-Pierre Delange <adeimantos\#at\#free.fr>
 - Daniel Desages <daniel\#at\#desages.com>
- Italian translation
 - Davide Di Lazzaro <mc0315\#at\#mclink.it> (leader: it)
- Portuguese (Brazil) translation
 - Paulo Rogério Ormenese <pormenese\#at\#uol.com.br> (leader: pt-br)

- Andre Luis Lopes <andrelp\#at\#ig.com.br>
- Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
- Rildo Taveira de Oliveira <to_rei\#at\#yahoo.com>
- Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
- Gustavo Noronha Silva <kov\#at\#debian.org> (coordinator)
- Spanish translation
 - Walter Echarri <wecharri\#at\#infovia.com.ar> (leader: es)
 - José Carreiro <ffx\#at\#urbanet.ch>
- German translation
 - Jens Seidel <tux-master\#at\#web.de> (leader: de)
 - Willi Dyck <wdyck\#at\#gmx.net>
 - Stefan Schröder <stefan\#at\#fkp.uni-hannover.de>
 - Agon S. Buchholz <asb\#at\#kefk.net>
- Polish translation—the following members of PDDP (<http://debian.linux.org.pl>):
 - Marcin Andruszkiewicz
 - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
 - Bartosz Feński <fenio\#at\#debian.linux.org.pl> (leader: pl)
 - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
 - Bartosz 'Xebord' Janowski
 - Jacek Lachowicz
 - Rafał Michaluk
 - Leonard Milcin, Jr.
 - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
 - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
 - Tomasz Piękoś
 - Jacek Politowski
 - Mateusz Prichacz <mateusz\#at\#debian.linux.org.pl>
 - Marcin Rogowski
 - Paweł Różański
 - Mariusz Strzelecki
 - Krzysztof Ścierski
 - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
 - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
 - Cezary Uchto
 - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
 - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- Chinese (simplified) translation
 - Hao "Lyoo" LIU <iamlyoo\#at\#163.net>
 - Ming Hua <minghua\#at\#rice.edu>
 - Xiao Sheng Wen <atzlinux\#at\#163.com> (leader: zh-cn)
 - Haifeng Chen <optical.dlz\#at\#gmail.com>
 - Xie Yanbo <xieyanbo\#at\#gmail.com>
 - easthero <easthero\#at\#gmail.com>
- Chinese (traditional) translation
 - Tang Wei Ching <wctang\#at\#csie.nctu.edu.tw> (leader: zh-tw)
- Japanese translation

– Shinichi Tsunoda <tsuno\#at\#ngy.lst.ne.jp> (leader: ja)

A.2 Warranties

Since I am not an expert, I do not pretend to be fully knowledgeable about Debian or Linux in general. Security considerations I use may only be applicable for home use.

This document does not replace any authoritative guides.

All warranties are disclaimed. All trademarks are property of their respective trademark owners.

A.3 Feedback

Comments and additions to this document are always welcome. Please send email to the Debian BTS system (<http://bugs.debian.org/>) under the debian-reference package or under the respective translation packages. Use of reportbug makes it easy to file a thorough bug report. You may still send email to Osamu Aoki (<http://people.debian.org/~osamu/>) at <osamu\#at\#debian.org> in English or to each translator in their respective language.