# Справочник по Debian

Osamu Aoki <osamu\#at\#debian.org>
Перевод Ильи В. Головко <qref\#at\#yandex.ru>
'Authors' на стр. 253

CVS, Чтв 18. Янв 2007, 11:54:11 UTC

## Аннотация

Этот Справочник по Debian (http://qref.sourceforge.net/) представляет широкий обзор системы Debian в форме руководства пользователя после установки. Он охватывает многие аспекты системного администрирования, используя примеры в командой строке. Содержит руководство по основам системы, советы и многую другую информацию по фундаментальным концепциям Debian, процессу установки, системе управления пакетами, ядру Linux в среде Debian, построению шлюза, текстовым редакторам, системе CVS, программированию и средствам GnuPG для рядовых пользователей и администраторов.

# Оглавление

# Глава 1

# Введение

Этот Справочник по Debian (http://qref.sourceforge.net/) дает широкий обзор системы Debian в виде руководства пользователя после установки. Предназначается для тех, кто не боится работы в командной строке. Я предполагаю, что читатель уже имеет базовые навыки работы в Unix-подобных системах перед прочтением данного документа.

Я сделал осознанное решение не объяснять до мелочей все то, что может быть найдено в man-страницах, info-страницах, или документах HOWTO. Вместо подробных объяснений я попытался дать больше практической информации, предоставив точные последовательности команд как в основном тексте, так и в Примерах скриптов из http://www.debian.org/doc/manuals/debian-reference/examples/. Вы должны понимать содержимое примеров перед вводом команд. И возможно Ваша система может потребовать несколько другой последовательности команд.

Много приведенной здесь информации состоит из указателей на авторитетные ссылки, приведенные в ‘References’ на стр. 245. Это, отчасти, из-за того, что документ изначально разрабатывался в форме “Краткого справочника”.

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

За помощью при аварийном обслуживании системы немедленно переходите к секции ‘Команды выживания в системе Debian’ на стр. 88.

## 1.1 Официальный документ

Самый последний официальный документ располагается в архивах Debian в форме пакета debian-reference-en. Он также доступен по адресу http://www.debian.org/doc/manuals/debian-reference/.

Самая последняя разрабатываемая версия находится по адресу http://qref.sourceforge.net/Debian/. Проект размещается на сайте http://qref.sourceforge.net/, где данный документ доступен для скачивания в форматах “просто текст”, HTML, PDF, SGML и PostScript.

## 1.2 Соглашения, принятые в документе

Этот “Справочник по Debian” предоставляет информацию путем коротких скриптов для командного интерпретатора Bash. Используются следующие соглашения:

```
# команда от имени пользователя root
$ команда от имени обычного пользователя
... описание действия
```

Дополнительную информацию по командному интерпретатору Bash смотрите в ‘Bash – GNU standard interactive shell’ на стр. 221.

Ссылки на:
- Unix man-страницу даются в форме bash(1).
- страницу GNU TEXINFO даются в форме info libc.
- книгу даются в форме Язык программирования Си.
- URL даются в форме http://www.debian.org/doc/manuals/debian-reference/.
- файл в системе даются форме /usr/share/doc/Debian/reference/.

Используются следующие абревиатуры:
- LDP: Linux Documentation Project (проект документации системы Linux) (http://www.tldp.org/)
- DDP: Debian Documentation Project (проект документации системы Debian) (http://www.debian.org/doc/)

В этом справочнике для документов из LDP даются только адреса URL, хотя они (документы) могут быть получены в виде пакетов и установлены в каталог /usr/share/doc/HOWTO/. Смотрите ‘References’ на стр. 245.

## 1.3 Примеры скриптов

Примеры скриптов доступны в каталоге examples subdirectory (http://www.debian.org/doc/manuals/debian-reference/examples/), устанавливаемые в /usr/share/doc/Debian/reference/; для скрытых файлов точка в начале имени файла заменяется на подчеркивание “_”. Когда имеется несколько альтернативных примеров скрипта, то добавляется добавочное расширение имени файла.

## 1.4 Базовая установка

Если систему установлена с минимальным количеством пакетов, то нужно выполнить следующую последовательность команд для установки некоторых нужных пакетов и ключевых документов:

```
# aptitude install mc less ssh vim kernel-package \
        manpages-dev doc-debian doc-linux-text \
        debian-policy developers-reference maint-guide \
        apt-howto harden-doc debian-reference \
        libpam-doc glibc-doc samba-doc exim4-doc-html
```

## 1.5 Основы дистрибуции Debian

Релизы Debian бывают трех видов :

- stable (стабильный) : Хорош для установки на рабочий сервер. Скучноват для рабочей станции (WS). Смотрите 'Дистрибутив stable' на стр. 6.
- testing (тестируемый): Хорош для установки на рабочую станцию. Смотрите 'Дистрибутив testing' на стр. 7.
- unstable (нестабильный): Не стоит слепо устанавливать. Смотрите 'Дистрибутив unstable' на стр. 7.

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 приближению к выпуску), и Sid (альфа-тест). В августе 2002 после выхода Woody эти имена релизов стали соответствовать Woody (рабочий), Sarge (бэта-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. Смотрите 'Архивы Debian' на стр. 5.

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

- 'Обновление до дистрибутива testing' на стр. 77 (в основном для рабочих станций)
- 'Портирование пакета в систему на базе stable' на стр. 96 (в основном для серверов)

Использование testing-дистрибутива Debian имеет побочный эффект, выражающийся в очень медленном исправлении ошибок, связанных с безопасностью, в ПО дистрибутива. Имейте это в виду.

Если Вы смешиваете виды релизов Debian, то использование частей testing или unstable в системе на базе stable приведет к нежелательному вытягиванию многих ключевых пакетов из testing или unstable, которые могут содержать ошибки. Имейте в виду и это.

Использование testing или unstable дистрибутивов Debian увеличивает риск столкнуться с серьезными ошибками. Этого можно избежать путем развертывания бутового загрузчика с возможностью выбора более стабильного дистрибутива Debian, или использования программы chroot для более стабильного дистрибутива (описано в 'chroot' на стр. 130). Это позволит выполнить одновременно разные виды Debian на разных консолях.

После разъяснения основ дистрибутивов Debian в ‘Основы Debian’ на стр. 5, я представлю некоторую базовую информацию, помогающую Вам жить счастливо с самым последним ПО, взяв преимущества Debian-дистрибутивов testing и unstable. Нетерпеливым можно сразу перейти к ‘Команды выживания в системе Debian’ на стр. 88. Счасливых обновлений!

# Глава 2

# Основы Debian

Эта секция обеспечивает основную информацию по системе Debian для пользователей. За авторитетной информацией обращайтесь к документам:

- Руководство по политике Debian
- Руководство по созданию пакетов в Debian
- Справочник Debian-разработчика
- Руководство начинающего Debian-мейнтейнера

ссылки на которые приведены в 'References' на стр. 245.

Если вы ищете менее детализированные описания (в форме HOW-TO), переходите к главе 'Управление пакетами в Debian' на стр. 81 или другие соответствующие главы.

Эта статья основана на документах, взятых из "Debian FAQ", тщательным образом реорганизованного в форму введения для обычного администратора системы Debian.

## 2.1 Архивы Debian

### 2.1.1 Структура каталогов

Программное обеспечение, сформированное в debian-пакеты, доступно в одной из нескольких древовидных структур каталогов на каждом зеркальном сайте Debian Debian mirror site (http://www.debian.org/mirror/) по протоколам FTP или HTTP.

Следующие каталоги могут быть найдены на каждом зеркале Debian в каталоге debian:

dists/: Этот каталог содержит "дистрибутивы" и используется для канонического пути для доступа к имеющимся (в настоящее время) пакетам в релизах и пре-релизах Debian. Некоторые старые пакеты и файлы Packages.gz могут быть до сих пор и здесь.

pool/: Это новое физическое расположение всех пакетов релизов и пре-релизов Debian.

tools/: DOS-утилиты для создания загрузочных дискет, разбиения вашего жесткого диска, сжатия/распаковки файлов и загрузки Linux.

doc/: Это основная документация по Debian, такая как FAQ, инструкции по системе оповещения об ошибках и т.д.

indices/: The Maintainers file and the override files.

project/: материалы, в основном, для разработчиков. Это:

project/experimental/: Этот каталог содержит пакеты и инструменты, которые находятся в разработке или даже в альфа-тестировании. Пользователи не должны использовать эти пакеты, так как они могут быть опасны и вредны даже для достаточно опытных.

project/orphaned/: Здесь находятся пакеты, которые 'осиротели' (т.е. остались без мейнтейнера) и были изъяты из дистрибутива.

### 2.1.2 Дистрибутивы Debian

Обычно существует три дистрибутива Debian в каталоге dists. Это дистрибутив stable, дистрибутив testing и дистрибутив unstable. Иногда может быть еще и frozen. Каждый дистрибутив определяется как символическая ссылка на реальный каталог под кодовым именем в каталоге dists.

### 2.1.3 Дистрибутив stable

Пакеты stable дистрибутива Debian Sarge (3.1r0) записываются в каталог stable (символическая ссылка на каталог sarge/):

- stable/main/: Этот каталог содержит пакеты, которые формально составляют самый свежий релиз системы Debian.

  Эти пакеты полностью отвечают Рекомендациям об открытом ПО в Debian (DFSG) (http://www.debian.org/social_contract#guidelines) (смотрите также файл /usr/share /doc/debian/social-contract.txt из пакета debian-doc) и являются полностью свободными к использованию и распространению.

- stable/non-free/: Этот каталог содержит пакеты, распространение которых ограничено требованиями ряда копирайтов.

  Например, некоторые пакеты имеют лицензию, которая запрещает коммерческую дистрибуцию. Другие могут распространяться в составе дистрибутива, но в действительности являются shareware и несвободным ПО. Лицензия каждого из этих пакетов должна быть изучена и учтена перед тем, как пакет будет включен в дистрибутив на CDROM и т.д.

- stable/contrib/: Этот каталог содержит пакеты, которые сами по себе являются свободными (отвечают DFSG) и могут свободно распространяться, но неким образом зависят от несвободного пакета из non-free секции.

Теперь, в дополнение к выше указанным расположениям пакетов, пакеты физически располагаются в каталоге pool (‘Каталог pool’ на стр. 9).

Текущее состояние по найденным ошибкам в stable дистрибутиве отображается на странице Stable Problems (http://ftp-master.debian.org/testing/stable_probs.html).

### 2.1.4 Дистрибутив testing

Пакеты для дистрибутива testing, Debian Etch, записываются в testing каталог (символическая ссылка на etch/) после того, как они пройдут некоторое тестирование в unstable. Физически пакеты располагаются в каталоге pool’ (‘Каталог pool’ на стр. 9). В каталоге testing/ также имеются подкаталоги main, contrib и non-free, которые выполняют те же функции, что и в дистрибутиве stable/.

Для всех архитектур, под которые собираются пакеты дистрибутива testing, обеспечивается синхронность версий, также эти пакеты не должны иметь зависимостей, которые могли бы привести к невозможности их удалить, и должны иметь меньше критических ошибок, чем версия, находящаяся сейчас в unstable. Таким образом, мы надеемся, что testing всегда близок, чтобы стать кандидатом в релиз. Подробности о механизме тестирования смотрите в документе http://www.debian.org/devel/testing.

Самое последнее состояние дистрибутива testing публикуется на этих сайтах:
- update excuses (http://ftp-master.debian.org/testing/update_excuses.html)
- проблемы testing (http://ftp-master.debian.org/testing/testing_probs.html)
- критические ошибки (http://bugs.debian.org/release-critical/)
- ошибки в базовой системе (http://bugs.qa.debian.org/cgi-bin/base.cgi)
- ошибки в стандартных пакетах и пакетах задач (http://bugs.qa.debian.org/cgi-bin/standard.cgi)
- замечания по другим ошибкам и их исправлениям (http://bugs.qa.debian.org/)

### 2.1.5 Дистрибутив unstable

Пакеты для unstable дистрибутива, который всегда имеет кодовое имя “Sid”, сохраняются в каталоге unstable (символическая ссылка на sid/) сразу после того, как их закачают в Debian-архив и они находятся там до их перемещения в testing/. Сами пакеты размещаются в каталоге pool (‘Каталог pool’ на стр. 9). В каталоге unstable также существуют подкаталоги main, contrib и non-free, которые выполняют те же функции, что и в дистрибутиве stable /. Дистрибутив unstable содержит снимок разрабатываемой в настоящий момент системы. Вы можете использовать и тестировать эти пакеты, осознавая состояние их готовности. Преимущество от использования дистрибутива unstable в том, что вы всегда используете самое последнее ПО из проекта Debian — оно является и самым нестабильным.

Текущее состояние ошибок в дистрибутиве unstable отображается на странице Unstable Problems (http://ftp-master.debian.org/testing/unstable_probs.html).

### 2.1.6 Дистрибутив frozen

Когда дистрибутив testing достаточно созреет, он замораживается, означая тем самым, что новый код более не принимается, а только исправления ошибок, если таковые необходимы. Также, создается новое тестируемое дерево в каталоге dists с новым кодовым названием. Замороженный дистрибутив проходит в течение нескольких месяцев фазу тестирования с нерегулярными обновлениями и глубокими заморозками, называемыми тест-циклами.

Мы ведем запись ошибок в замороженном дистрибутиве, которые могут задержать какой-либо пакет от прохождения его в релиз или ошибок, которые могут задержать весь релиз от выхода. В момент, когда количество ошибок понизится до максимально приемлемого значения, frozen становится стабильным, он выпускается, а предыдущий стабильный дистрибутив становится obsolete (устаревшим)( и переносится в архив).

### 2.1.7 Кодовые имена дистрибутивов Debian

Реальные имена каталогов в каталоге dists, такие как sarge/ и etch/ являются просто кодовыми именами. Когда дистрибутив Debian находится в стадии разработки, он не имеет номера версии, зато имеет кодовое имя. Такое назначение кодовых имен упрощает зеркалирование дистрибутива Debian(если реальное имя каталога unstable менялось бы на stable/ (дистрибутив стал стабильным - прим. переводчика), то пришлось бы скачивать повторно большое количество файлов).

В настоящее время stable/ - это символическая ссылка на каталог sarge/, а testing/ - ссылка на etch/. Это означает, что Sarge является текущим стабильным дистрибутивом, а Etch находится в стадии тестирования.

unstable/ - это постоянная символическая ссылка на каталог sid/, так как Sid является всегда нестабильным дистрибутивом.

### 2.1.8 Кодовые имена прошлого

Вот кодовые имена, которые уже использовались: “Buzz” для релиза 1.1,“Rex” для релиза 1.2, “Bo” для релизов 1.3.x, “Hamm” для релиза 2.0, “Slink” для релиза 2.1 и “Potato” для релиза 2.2.

### 2.1.9 Источник кодовых имен

Имена дистрибутивов до сих пор были именами персонажей из фильма Toy Story от студии Pixar.

- Buzz (Buzz Lightyear) был космонавтом
- Rex был тиранозавром,
- Bo (Bo Peep) была девочкой, ухаживающей за овечкой,
- Hamm был piggy bank
- Slink (Slinky Dog) был игрушечной собакой,

- Sarge был предводителем армии людей Зеленого Пластика,
- Potato был, конечно, мистером Картошечья Голова,
- Woody был ковбоем,
- Sid был парнем по-соседству, который сломал игрушки.

### 2.1.10 Каталог pool

Исторически пакеты хранились в каталоге дистрибутива, для которого они были созданы. Сам каталог дистрибутива находился в каталоге dists. Это вызывало различного рода проблемы, в том числе и большой расход пропускной способности канала на зеркалирующих серверах, когда производились большие изменения.

Теперь пакеты хранятся в большом “пуле”, структурированном в соответствии с именем исходного пакета. Чтобы сделать пул управляемым, он поделен на секции (main, contrib, и non-free) и далее по первым буквам имени исходного пакета. Эти каталоги содержат несколько файлов: бинарные пакеты для каждой архитектуры и пакеты с исходными текстами, на основе которых эти бинарные пакеты были собраны.

Вы можете выяснить, где размещается каждый пакет, выполнив команду типа apt-cache showsrc имя_моего_пакета и посмотрев на строку “Directory:”. Например, пакеты apache хранятся в pool/main/a/apache/. Так как существует очень большое количество пакетов библиотек lib*, то они обрабатываются несколько иначе. Например, пакеты libpaper хранятся в pool/main/libp/libpaper/.

Каталоги dists все еще используются индексными файлами для программ типа apt. Также, на момент написания этого документа, старые дистрибутивы еще не сконвертированы использовать пулы, и поэтому в поле Directory заголовка пакета Вы увидите пути, содержащие имена дистрибутивов типа potato или woody.

Обычно Вам не нужно беспокоиться об этом, так как новая программа apt и, вероятно, старые dpkg-ftp (смотрите ‘Способы обновления системы Debian’ на стр. 20) обработают это без проблем. Если Вам нужно больше информации, смотрите RFC: Реализация пула пакетов (http://lists.debian.org/debian-devel-announce/2000/debian-devel-announce-200010/msg00007.html).

### 2.1.11 Историческая справка относительно Sid

Когда не существовало современного Sid, организация сайта с архивом Debian имела одни большой недостаток: существовало предположение, что когда создавалась архитектура в текущем unstable/, она могла быть также выпущена, когда сам дистрибутив переходил в фазу релиза или stable. Для множество архитектур это было неверным, что приводило к необходимости переноса соответствующих каталогов в момент релиза (в unstable - прим. пер.). Такие перемещения были непрактичными из-за повышенного расхода пропускной способности канала.

Администраторы архива решали эту проблему в течение нескольких лет путем помещения бинарных файлов для архитектур, которая не готова к выпуску, в специальный каталог, называемый sid. Когда такая архитектура входила в фазу релиза, то в первое время существовала ссылка из текущего stable/ на sid/, and from then on they were created inside the unstable/tree as usual. Это расположение вводило в заблуждение пользователей.

С приходом пула пакетов (смотрите 'Каталог pool' на стр. 9) в процессе разработки дистрибутива Woody, бинарные пакеты начали хранить в каноническом месте пула вне зависимости от дистрибутива, и таким образом выпуск дистрибутива больше не вызывает повышенного расхода пропускной способности канала на зеркалах (существует, однако,a lot of gradual bandwidth consumption throughout the development process).

### 2.1.12 Пакеты, закаченные в incoming/

Закаченные пакеты, находящиеся в http://incoming.debian.org/ проходят проверку для гарантии того, что они в действительности отправлены разработчиком Debian (в противном случае, т.е. случае закачки пакета лицом, не являющимся мейнтейнером - закачка NMU, пакет помещается в под-каталог DELAYED). Один раз в день пакеты из incoming/ перемещаются в unstable/.

В крайних случаях Вы можете пожелать установить пакеты из incoming/ до того, как они достигнут unstable/.

### 2.1.13 Доступ к старым пакетам

В то время, как самые последние дистрибутивы Debain хранятся в каталоге debian на каждом зеркале Debian (http://www.debian.org/mirror/), архивы старых дистрибутивов типа Slink хранятся по адресу http://archive.debian.org/ или в каталоге debian-archive на каждом зеркале Debian.

Пакеты старых testing и unstable дистрибутивов могут быть найдены по адресу http://snapshot.debian.net/.

### 2.1.14 Секции архитектрур

Внутри каждого основного дерева каталогов (dists/stable/main, dists/stable/contrib, dists/stable/non-free, dists/unstable/main/, и пр.), бинарные пакеты размещаются в подкаталогах, чьи имена обозначают архитектуру процессора, под которую эти пакеты были собраны.

- binary-all/ - для архитектурно-независимых пакетов. Сюда относятся, например, скрипты на языке Perl, документация.

- binary-platform/ - для пакетов, которые могут исполняться на определенной аппаратной платформе.

Обратите внимание, что бинарные пакеты дистрибутивов testing и unstable теперь размещаются не в описанных каталогах, а каталоге верхнего уровня pool. Индексные файлы типа (Packages и Packages.gz) тем не менее хранятся для обратной совместимости.

В каждом дистрибутиве для поддерживаемых аппаратных архитектур существуют Release Notes. Их можно найти на странице Release Notes соответствующего дистрибутива: stable (http://www.debian.org/releases/stable/releasenotes) и testing (http://www.debian.org/releases/testing/releasenotes).

### 2.1.15 Исходный код

Для всех компонентов системы Debian имеется исходный код. Более того, условия лицензий большинства программ в системе требуют, чтобы исходный код распространялся вместе с программой, либо был бы способ его получить.

Обычно исходный код располагается в каталогах source, находящиеся по-соседству с каталогами архитектур, или, с недавнего времени, в каталоге pool (смотрите ‘Каталог pool’ на стр. 9). Чтобы получить исходный код пакета, не вникая в структуру архива Debian, используйте команду apt-get source mypackagename.

Некоторые пакеты, например pine, доступны только в виде исходных текстов из-за ограничений лицензии. (Недавно был представлен пакет pine-tracker для облегчения установки Pine.) Процедуры, описанные в ‘Портирование пакета в систему на базе stable’ на стр. 96 и ‘Packaging’ на стр. 238 описывают способы ручной сборки пакетов.

Исходный код для пакетов из каталогов contrib и non-free, может отсутствовать, так как пакеты из этих каталогов формально не являются часть системы Debian.

## 2.2 Система управления пакетами в Debian

### 2.2.1 Обзор пакетов Debian

Пакеты, как правило, содержат все необходимые файлы для реализации какого-либо набора команд или возможностей. Существует два типа пакетов Debian:

- Бинарные пакеты, которые содержат исполняемые и конфигурационные файлы, страницы руководств в формате man/info, информацию о копирайтах и другую документацию. Эти пакеты распространяются в специальном архивном формате Debian (смотрите ‘Формат Debian пакета’ на стр. 12) и обычно выделяются наличием .deb расширения файлов. Бинарные пакеты могут быть распакованы при помощи утилиты Debian dpkg; подробности приведены в ее man-странице.

- Пакеты с исходным текстом, которые состоят из .dsc файла, описывающего пакет (включая имена далее идущих файлов), файла .orig.tar.gz, который содержит немодифицированный исходный код в формате tar и упакованный программой gzip, и обычно файл .diff.gz, который содержит изменения исходного текста, специфичные для Debian. Утилита dpkg-source упаковывает и распаковывает пакеты Debian с исходными текстами; подробности смотрите в ее man-странице.

Установка программного обеспечения при помощи пакетной системы использует так называемые “зависимости”, которые тщательно спроектированы мейнтейнерами пакета. Эти зависимости задокументированы в файле control, связанным с каждым пакетом. Например, пакет, содержащий компилятор GNU C (gcc) “зависит” от пакета binutils, который включает в себя компоновщик и транслятор. Если пользователь попытается установить gcc без предварительной установки binutils, система управления пакетами (dpkg) выведет сообщение об ошибке, что также необходима установка binutils, и прервет установку gcc. (Однако, это поведение может быть изменено по требованию пользователя; смотрите dpkg(8).) Дополнительные подробности смотрите ‘Зависимости пакетов’ на стр. 16 ниже.

Инструменты для работы с пакетами Debian можно использовать для того, чтобы:

- манипулировать и управлять пакетами или их частями,
- помочь пользователю разбить пакеты, которые нужно передать на носителях ограниченного размера (флоппи-диски),
- помочь разработчикам в создании архивов пакетов,
- помочь пользователям в установке пакетов, размещающихся на удаленном сайте архива Debian.

### 2.2.2 Формат Debian пакета

“Пакет” Debian или архивный файл Debian содержит исполняемые файлы, библиотеки и документацию, ассоциированные с некоторым программмным комплектом или набором связанных программ. Обычно архивный файл Debian имеет имя файла с расширением .deb. [1]

Внутренние подробности этого формата бинарного пакета Debian описывются на deb(5) man-странице. Так как внутренний формат может изменяться (от релиза к релизу Debian), всегда используйте страницу dpkg-deb(1) для манипулирования .deb файлами.

До дистрибутива Sarge включительно со всеми архивными файлами Debian можно работать стандартными командами Unix типа ar и tar, даже когда не доступны команды dpkg.

[1] Проект debian-installer ввел имена файлов пакетов, которые заканчиваются на .udeb. Коротко, это микро-.deb формат, который точно не следует политике Debian, не содержит документации и должен использоваться только debian-installer, который является новым установщиком Debian, разработанным для Sarge. Формат файла .udeb идентичен формату .deb. Программа udpkg, используемая для работы с .udeb пакетами имеет ограниченные возможности по сравнению с dpkg и меньше поддерживает связи пакетов. Разница в имени существует из-за того, что мейнтейнерам архива Debian не нравилось присутствие в архиве пакетов .deb, которые не отвечали политике, поэтому для них было выбрано другое имя, чтобы подчеркнуть это и не допустить их непреднамеренную установку на рабочую систему. Пакеты .udeb используются на initial ram-диске при установке базовой системы, только чтобы создать очень ограниченную систему Debian.

### 2.2.3 Соглашение об именах файлов Debian-пакетов

Имена файлов пакетов Debian подчиняются следующему соглашению:

foo_НомерВерсии-номерDebianРевизии.deb

где foo является именем пакета. Для проверки можно определить имя пакета, связанное с некоторым архивным файлом (.deb file) одним из следующих способов:

- Изучить файл “Packages” в каталоге дистрибутива на архивном сайте Debian. Этот файл содержит записи, описывающие каждый пакет; первое поле в них - это формальное имя пакета.
- Использовать команду dpkg —info foo_VVV-RRR.deb (где VVV и RRR - номера версии и ревизии запрашиваемого пакета, соответственно). Команда отображает, помимо других данных, имя пакета, соответствующее изучаемому архивному файлу.

Компонент VVV - это номер версии, указанный разработчиком программы (upstream developer). Не существует стандарта о нумерации версий, поэтому они могут иметь различные форматы типа “19990513” или “1.3.8pre1”.

Компонент RRR - это ревизионный номер Debian, который указывается разработчиком Debian (или частным пользователем, если он пожелал собрать пакет самостоятельно). Этот номер соответствует уровню ревизии пакета Debian, поэтому новый ревизионный номер обычно означает изменения в Debian makefile (debian/rules), файле Debian control (debian /control), скриптах установки или удаления (debian/p*), или конфигурационных файлах, используемых в пакете.

### 2.2.4 Сохранение текущей конфигурации

Сохранение файлов настройки реализуется через механизм “conffiles” системы Debian. Файлы настройки (обычно размещаемые в /etc/) указываются в файле conffiles системы пакетов Debian. Система управления пакетами гарантирует, что при обновлении пакета эти файлы не будут перезаписаны.

When it is possible to configure the system without modifying files that belong to various Debian packages, it is usually a good idea not to modify them even if they are “conffiles”. This ensures faster and smoother upgrade operations.

Чтобы точно определить, какие файлы будут сохранены при обновлении пакета, выполните:

```
dpkg --status package
```

и смотрите под строкой “Conffiles:”.

Подробности, связанные с содержимым Debian-файла conffiles, представлены в руководстве по политике Debian, секция 11.7 (смотрите ‘References’ на стр. 245).

### 2.2.5 Скрипты сопровождения Debian

Скрипты сопровождения Debian это исполняемые скрипты, автоматически выполняемые перед или после установки пакета. Вместе с файлом control, эти файлы являются частью секции “control” архивного файла Debian.

В частности, такими файлами являются:

preinst Этот скрипт выполняется до распаковки пакета, к которому он принадлежит, из архивного файла Debian (.deb). Многие “preinst” скрипты останвливают сервисы обновляемых пакетов до окончания установки или обновления (с последующим успешным выполнением скрипта “postinst”).

postinst Этот скрипт обычно завершает конфигурирование пакета после его распаковки из архивного файла Debian (.deb). Часто скрипт “postinst” запрашивает у пользователя некоторую информацию и/или предупреждает пользователя что, если он принимает значения по умолчанию, то нужно будет не забыть переконфигурировать пакет, как это требуется. Многие скрипты “postinst” затем выполняют команды, необходимые для запуска или перезапуска сервиса после установки или обновления пакета.

prerm Этот скрипт обычно останавливает какие-либо демоны (сервисы - прим. переводчика), связанные с пакетом. Он выполняется перед удалением файлов пакета.

postrm Этот скрипт обычно модифицирует ссылки или другие файлы, связанные с пакетом, и/или удаляет файлы, созданные им. (Смотрите также ‘Виртуальные пакеты’ на стр. 15.)

В настоящее время все control-файлы могут быть найдены в каталоге /var/lib/dpkg/info. Файлы, относящиеся к пакету foo начинаются с имени “foo” и, соответственно, имеют расширение файла типа “preinst”, “postinst”, и так далее. Файл foo.list в этом каталоге описывет все файлы, установленные с пакетом foo. (Заметьте, что месторасположение этих файлов является внутренней особенностью программы dpkg, и поэтому может меняться.)

### 2.2.6 Приоритеты пакетов

В помощь системе управления пакетами каждому пакету Debian мейнтейнерами дистрибутива назначается приоритет priority. Приоритеты бывают:

- Required (требующиеся) пакеты, которые необходимы для правильного функционирования системы.

  Сюда относятся все инструменты, которые необходимы для исправления повреждений системы. Вы не должны удалять эти пакеты, иначе Ваша система может стать неработоспособной, и Вы даже не сможете использовать программу dpkg, чтобы вернуть все назад. Системы с установленными только пакетами Required могут не соответствовать для решения большинства задач, но они имеет достаточно функций для того, чтобы позволить системному администратору загрузить и установить дополнительное программное обеспечение.

- Important (важные) пакеты должны быть найдены на любой Unix-подобной системе.

  Пакеты с таким приоритетом - это остальные пакеты, без которых система не будет нормально работать или не будет удобной. Сюда не относятся пакеты типа Emacs, X11, TeX или любые другие крупные приложения. Эти пакеты составляют минимальную инфраструктуру.

- Standard (стандартные) пакеты являются стандратными на любой Linux системе, включая разумно небольшую, но не ограниченную систему текстового режима.

  Это то, что устанавливается по умолчанию, если пользователь ничего не выбрал. "Standard" не включает многих крупных приложений, но уже включает Emacs (Это больше часть инфрастурктуры, чем приложение.) и разумное подмножество систем Tex и LaTeX (разумное настолько, чтобы не требовать X11).

- Optional (опциональные) пакеты включает все те, которые вы можете захотеть установить, даже если с ними не знакомы, и если не имеете специальных требований.

  Это включает X11, полностью TeX и множество приложений.

- Extra пакеты, которые или конфликтуют с другими более высокими приоритетами, несут малую пользу для пользователей, с ними незнакомыми, или имеют специальные требования, делающие их неподходящими к "Optional".

Пожалуйста, заметьте разницу между "Priority: required", "Section: base" и "Essential: yes" в описании пакета. "Section: base" означает, что пакет устанавливается перед чем бы то ни-было на новой системе. Большинство пакетов с "Section: base" имеют приоритет "Priority: required" или "Priority: important", и многие из них помечены как "Essential: yes". "Essential: yes" означает, что при удалении этого пакета из системы необходимо указать дополнительную форсирующую опцию системе управления пакетами dpkg. Например, пакеты libc6, mawk и makedev являются пакетами с приоритетом "Priority: required" из секции "Section: base", но не являются "Essential: yes".

### 2.2.7 Виртуальные пакеты

Виртуальный пакет - это общее имя, употребляемое к любому из пакетов некой группы, все пакеты из которой обеспечивают подобную базовую функциональность. Например, пакеты tin и trn являются программами для чтения news-конференций, и любая из них должна, следовательно, удовлетворять зависимость программы, требующей для нормального своего функционирования установленной программы чтения news-конференций. Эти пакеты, таким образом, обеспечивают "виртуальный пакет", называемый news-reader.

Подобным образом, оба пакета exim и sendmail обеспечивают функциональность почтового транспортного агента. Следовательно, они предоставляют виртуальный пакет mail-transport-agent. Если один из них установлен, то любая программа, зависящая от наличия установленного почтового транспортного агента будет удовлетворена существованием этого виртуального пакета.

Debian имеет такой механизм, что если в системе установлено более одного пакета, предоставляющего виртуальный пакет, то системный администратор может установить один из них, как предпочитаемый. Для этого используется команда update-alternatives, описываемая далее 'Alternative commands' на стр. 99.

### 2.2.8 Зависимости пакетов

Система пакетов в Debian имеет ряд "зависимостей" пакетов, которые предназначены для того, чтобы показать (одним признаком) уровень, на котором программа A может работать независимо от существования программы B в данной системе:

- Пакет A зависит (depends) от пакета B, если пакет B безусловно должен быть установлен для выполнения пакета A. В некоторых случаях, пакет A зависит не только от пакета B, а еще и от конкретной версии пакета B. В этом случае зависимость от версии пакета означает нижний предел, в том смысле, что пакет A зависит от любой версии пакета B, не старее указанной версии.

- Пакет A рекомендует (recommends) пакет B, если мейнтейнер пакета решил, что большинство пользователей не захотят иметь пакет A без функциональности, предоставляемой пакетом B.

- Пакет A предлагает (suggests) пакет B, если пакет B содержит файлы, относящиеся к (или обычно улучшающие) функциональности пакета A.

- Пакет A конфликтует (conflicts) с пакетом B, когда пакет A не будет работать, если пакет B установлен в системе. Более частый случай конфликта, когда пакет A содержит файлы, которые являются улучшением тех, что находятся в пакете B. Статус "конфликтует" часто комбинируется с "заменяет".

- Пакет A заменяет (replaces) пакет B, когда установленные файлы пакета B удаляются и, в некоторых случаях перезаписываются файлами пакета A.

- Пакет A обеспечивает (provides) пакет B, когда все файлы и функциональность пакета B включается в состав пакета A. Этот механизм предоставляет для пользователей с ограниченным дисковым пространством способ получить только реально нужную им часть пакета A.

Более подробную информацию об использовании каждого из этих терминов можно найти в Руководстве по созданию пакетов и в Руководстве по политике.

Заметьте, что программа dselect имеет более тонко настроенный контроль над пакетами, указываемыми зависимостями вида рекомендует и предлагает, нежели программа apt-get, которая просто вытаскивает все пакеты, указанные в зависимости зависит и не трогает все пакеты, указанные в зависимостях пакета рекомендует и предлагает. Обе программы в современном виде используют за основу систему APT.

### 2.2.9 Значение “pre-depends”

“Pre-depends” - специальный вид зависимости. В случае обычного пакета программа dpkg распаковывает архивный файл пакета (т.е., файл .deb) вне зависимости от существования в системе файлов, от которых зависит данный пакет. Распаковка же, по существу, означает, что программа dpkg извлечет из архива .deb файлы, подлежащие установке на Вашу файловую систему и разместит их на своих местах. Если данный пакет зависит (depends) от наличия неких других пакетов на Вашей системе, то программа dpkg откажется завершить установку (выполняя действие “configure”) до тех пор, пока требуемые пакеты не будут установлены.

Однако, существуют некоторые пакеты, даже распаковка которых при помощи программы dpkg будет отменена, пока не будут разрешены определенные зависимости. Считается, что такие пакеты критично зависят (“pre-depend”) (а как еще перевести “pre-depends? - прим. переводчика) от присутствия некоторых других пакета(ов). Проект Debian предоставил этот механизм, чтобы поддержать безопасное обновление систем с формата a.out на ELF формат, когда был критичен порядок, в котором пакеты распаковываются. Бывают еще другие ситуации крупных обновлений, где этот метод полезен, например, в случае пакетов с приоритетом ”Required“ и их зависимостями от libc.

Замечу, что более детальная информация об этом может быть найдена в Руководстве по созданию пакетов.

### 2.2.10 Статус пакета

Статус пакета может быть ”unknown“ (неизвестный), ”install“ (установить), ”remove“ (удалить), ”purge“ (очистить) или ”hold“ (удержать). Эти предписывающие флаги показывают то, что пользователь желает сделать с пакетом (делая свой выбор в секции ”Select“ программы dselect, или прямо вызывая программу dpkg).

Смысл эти значений:

- unknown (неизвестный) - пользователь никогда не показывал свою потребность в пакете.
- install (установить) - пользователь желает, чтобы пакет был установлен или обновлен.
- remove (удалить) - пользователь желает, чтобы пакет был удален, но не желает удалять никакие конфигурационные файлы.
- purge (очистить) - пользователь желает, чтобы пакет был полностью удален, включая его конфигурационные файлы.
- hold (удержать) - пользователь желает, чтобы этот пакет не обрабатывался, т.е. он хочет сохранить его текущую версию с его текущим состоянием, каким бы оно не было.

### 2.2.11 Удержание пакетов от обновления

Существует два механизма для удержания пакета от обновления: через программу dpkg или, начиная с Woody, через систему APT.

В случае с dpkg, сначала экспортируем список выбора пакетов:

```
dpkg --get-selections \* > selections.txt
```

Затем отредактируем получившийся файл selections.txt, изменив строку, содержащую пакет, который хотим удержать, например libc6, следующим образом:

```
libc6        install
```

заменяем на:

```
libc6        hold
```

Сохраням файл и перезагружаем его в базу данных dpkg таким способом:

```
dpkg --set-selections < selections.txt
```

Или, если вы знаете имя пакета для удержания, просто выполните:

```
echo libc6 hold | dpkg --set-selections
```

Эта процедура удерживает пакеты в процессе установки каждого пакета.

Тот же самый эффект можно получить, используя программу dselect. Просто войдите на экран [S]elect, найдите пакет, который желаете удержать в его настоящем состоянии и нажмите клавишу '=' (или 'H'). Изменения войдут в силу сразу после выхода из экрана [S]elect.

Система APT в дистрибутиве Woody имеет новый альтернативный механизм для удержания пакетов во время процесса выборки архива, используя Pin-Priority. Смотрите man-страницу apt_preferences(5), а также http://www.debian.org/doc/manuals/apt-howto/ или пакет apt-howto; глава 'Обзор файла /etc/apt/preferences' на стр. 87 тоже содержит краткое объяснение.

### 2.2.12 Пакеты с исходным текстом

Пакеты с исходным текстом располагаются в каталоге source, либо Вы можете скачать их вручную или использовать APT

```
apt-get source foo
```

чтобы их получить (смотрите man-страницу apt-get(8), чтобы настроить APT для этого).

### 2.2.13 Сборка бинарного пакета из пакета с исходным текстом

Чтобы скомпилировать исходный текст пакета foo, Вам нужны будут все следующие файлы - foo_*.dsc, foo_*.tar.gz и foo_*.diff.gz (заметьте, что не существует файла .diff.gz для самого пакета Debian).

Как только Вы их получите, и если у Вас установлен пакет dpkg-dev, команда

```
$ dpkg-source -x foo_версия-ревизия.dsc
```

распакует пакет с исходным текстом в каталог под именем foo-версия.

Вызовите следущую команду, чтобы собрать бинарный пакет:

```
$ cd foo-версия
$ su -c "apt-get update ; apt-get install fakeroot"
$ dpkg-buildpackage -rfakeroot -us -uc
```

Далее,

```
# su -c "dpkg -i ../foo_версия-ревизия_архитектура.deb"
```

чтобы установить только что собранный бинарный пакет. Смотрите 'Портирование пакета в систему на базе stable' на стр. 96.

### 2.2.14 Создание новых пакетов Debian

За детальной информацией по созданию новых пакетов читайте Руководство новым мейнтейнерам (New Maintainers' Guide), имеющееся в пакете maint-guide или по адресу http://www.debian.org/doc/manuals/maint-guide/.

## 2.3 Обновление системы Debian

Одной из задач Debian является обеспечение совместимого пути обновления и надежного процесса обновления, и мы всегда делали все от нас зависящее по созданию таких новых релизов, которые способны гладко обновить предыдущие. Пакеты предупредят пользователя о важных моментах в процессе обновления и, зачастую, предоставят решение возможных проблем.

Вам также нужно изучить документ Release Notes, который описывает подробности конкретных обновлений, размещающийся на всех компакт-дисках с Debian, или по адресам http://www.debian.org/releases/stable/releasenotes и http://www.debian.org/releases/testing/releasenotes.

Практическое руководство по обновлению представлено в 'Управление пакетами в Debian' на стр. 81. Эта секция лишь описывает основные принципы.

### 2.3.1 Способы обновления системы Debian

Всегда можно просто, используя анонимный доступ по FTP или через программу wget зайти в архив Debian, внимательно изучить каталоги, найти желаемый файл, скачать его, и, наконец, установить его при помощи программы dpkg. (Заметьте, что программа dpkg установит файлы обновления на свое место даже на работающей системе.) Иногда, однако, обновленный пакет потребует установки новой обновленной версии еще одного пакета, и пока он не будет установлен, установка не будет завершена.

Много людей сочли, что этот ручной способ расходует массу времени, так как Debian развивается очень быстро — обычно каждую неделю закачивается множество новых пакетов. И их число увеличивается перед новым главным релизом. Чтобы иметь дело с таким потоком, пользователи предпочитают использовать автоматизированную программу для обновления. Для этого существует несколько специальных инструментов управления пакетами.

### 2.3.2 Обзор инструментов управления пакетами

Система управления пакетами в Debian имеет две цели: манипулирование собственно файлами пакетов и выборка файлов пакетов из архива Debian. Программа dpkg выполняет первую задачу, система APT and программа dselect последнюю.

### 2.3.3 Программа dpkg

Это основная программа для манипулирования файлами пакетов; за полным описанием обращайтесь к man-странице dpkg(8).

Программа dpkg идет вместе с несколькими простыми дополнительными программами:

- dpkg-deb: Манипулирует файлами .deb. dpkg-deb(1)
- dpkg-ftp: Старая команда для выборки файлов пакетов. dpkg-ftp(1)
- dpkg-mountable: Старая команда для выборки файлов пакетов. dpkg-mountable(1)
- dpkg-split: Разбивает большой пакет на более мелкие файлы. dpkg-split(1)

Программы dpkg-ftp и dpkg-mountable были замещены введением системы APT.

### 2.3.4 Система APT

Система APT (Улучшенный инструмент для работы с пакетами) представляет собой улучшенный интерфейс к системе управления пакетами в Debian, и состоит из нескольких программ, обычно начинающихся со префикса "apt-". Программы apt-get, apt-cache и apt-cdrom - это инструменты командой строки для манипулирования пакетами. Они также функционируют как пользовательские программы-платформы (back end) для других инструментов, таких как dselect и aptitude.

Для дополнительной информации установите пакет apt и прочтите соответствующие man-страницы: apt-get(8), apt-cache(8), apt-cdrom(8), apt.conf(5), sources.list(5), apt_preferences(5) (Woody), а также /usr/share/doc/apt/guide.html/index.html.

Альтернативным источником информации является страница APT HOWTO (http://www.debian.org/doc/manuals/apt-howto/). Она может быть установлена из пакета apt-howto, файл /usr/share/doc/Debian/apt-howto/.

Команды apt-get upgrade и apt-get dist-upgrade обновляют только пакеты, описываемые как "Depends:" и просматривают все пакеты с зависимостями "Recommends:" и "Suggests:". Чтобы избежать этого, используйте программу dselect.

### 2.3.5 Программа dselect

Это программа представляет собой управляемый с помощью меню интерфейс к системе управления пакетами в Debian. Она особенно полезна в первых установках или крупномасштабных обновлениях. Смотрите 'Программа dselect' на стр. 84.

Для дополнительной информации установите пакет install-doc и изучите файл /usr/share/doc/install-doc/dselect-beginner.en.html или страницу по адресу dselect Documentation for Beginners (http://www.debian.org/releases/woody/i386/dselect-beginner).

### 2.3.6 Обновление работающей системы

Ядро (файловая система) в Debian системах поддерживает замену файлов даже во время их использования.

Мы также предоставляем программу, называемую start-stop-daemon, которая используется для запуска демонов на этапе загрузки или для их останова, когда изменяется уровень исполнения ядра (например, при переходе из многопользовательского в однопользовательский режим или в состояние "останов" /halt/). Та же программа используется установочными скриптами, когда новый пакет содержит устанавливаемый демон, для остановки работающих демонов и их рестарта, если это необходимо.

Заметьте, что система Debian не требует использования однопользовательского режима для обновления работающей системы.

### 2.3.7 Скаченные и кэшированные архивные файлы .deb

Если Вы вручную скачали файлы пакетов на Ваш жесткий диск (что, кстати, совершенно не нужно; смотрите описание программ dpkg-ftp или APT выше), то после установки пакетов Вы можете удалить файлы .deb из системы.

Если же используется система APT, то эти файлы кэшируются в каталоге /var/cache/apt/archives. Вы можете стереть их после установки (командой apt-get clean) или скопировать их в каталог /var/cache/apt/archives другой машины с целью экономии входящего трафика при последующих установках.

### 2.3.8 Журналирование (record-keeping) процесса обновления

Программа dpkg хранит записи о пакетах, которые были распакованы, сконфигурированы, удалены, и/или очищены, но не хранит (в настоящее время) журнал действий пользователя на терминале во время работы с пакетами.

Самый простой путь исправить это - это запускать сессию программы dpkg, dselect, apt-get, и пр. с программой script(1).

## 2.4 Процесс загрузки Debian

### 2.4.1 Программа init

Как и все Unix-подобные операционные системы, Debian загружается, выполняя программу init. Конфигурационный файл для init (/etc/inittab) указывает, что первый скрипт, который должен быть выполнен, - это скрипт /etc/init.d/rcS. Данный скрипт запускает все скрипты из каталога /etc/rcS.d/ по порядку (как перевести by sourcing ? - прим. перев.) или порождая подчиненные процессы (в зависимости от расширения имени файла) с целью инициализации системы, как то проверка и монтирование файловых систем, загрузка модулей, запуск сетевых сервисов, установка системных часов и пр. Далее (для совместимости) этот скрипт выполняет файлы из каталога /etc/rc.boot/ (кроме тех, что имеют '.' в имени файла). Любые скрипты из последнего каталога обычно предназначаются для использования системным администратором, и применение их в пакетах не допускается. За дополнительной информацией обращайтесь к 'System initialization' на стр. 135 или странице System run levels and init.d scripts (http://www.debian.org/doc/debian-policy/ch-opersys#s-sysvinit) Руководства по политике Debian.

### 2.4.2 Уровни выполнения

По окончании процесса загрузки программа init выполняет все стартовые скрипты в каталоге, определяемым уровнем выполенения по умолчанию (это уровень выполнения задается элементом id в файле /etc/inittab). Как и большинство System V - совместимых Unix-систем, Линукс имеет 7 уровней выполнения:

- 0 (останов системы),
- 1 (однопользовательский режим),
- 2 - 5 (различные многопользовательские режимы), и
- 6 (перезагрузка системы).

Системы Debian идут с id=2, который показывает, что при входе в многопользовательский режим уровень выполнения по умолчанию - второй, и поэтому требуется выполнить скрипты из каталога /etc/rc2.d/.

Скрипты в любом из каталогов /etc/rcN.d/, по сути, являются символическими ссылками на скрипты из каталога /etc/init.d/. Однако, сами имена файлов в каждом /etc/rcN.d/ каталоге определяют способ, которым будут запущены скрипты из /etc/init.d/. Конкретнее, перед входом в любой уровень исполнения сначала запускаются все скрипты, начинающиеся с буквы ‘K’; данные скрипты останавливают сервисы. Далее выполняются все скрипты, начинающиеся в буквы ‘S’; эти скрипты запускают сервисы. Двузначное число после буквы ‘K’ или ‘S’ показывает порядок, в котором выполняются скрипты. Скрипты с меньшим номером выполняются первыми.

Это способ работает, так как все скрипты из каталога /etc/init.d/ принимают аргумент с одним из возможных значений ”start“, ”stop“, ”reload“, ”restart“ или ”force-reload“, и выполняют задачу, соответствующую значению данного аргумента. Эти скрипты могут также использоваться после загрузки системы для управления различными процессами.

Например, следующая команда с аргументом ”reload“

```
# /etc/init.d/sendmail reload
```

посылает демону sendmail сигнал, побуждающий его перечитать конфигурационный файл.

### 2.4.3 Настройка процесса загрузки

Debian не использует свойственный BSD каталог rc.local для настройки процесса загрузки; вместо этого он предоставляет следующий механизм.

Предположим, системе нужно на этапе загрузки или при входе в определенный (System V) уровень выполнения исполнить скрипт foo. Тогда системный администратор должен сделать:

1 Разместить скрипт foo в каталоге /etc/init.d/.

2 Выполнить Debian-команду update-rc.d с соответствующими аргументами, чтобы создать символические ссылки между каталогами rc?.d (задаваемый в командой строке) и файлом /etc/init.d/foo. Здесь ? - это номер от 0 до 6, который соответствует одному из System V уровней выполнения.

3 Перезагрузить систему.

Команда update-rc.d установит ссылки между файлами в каталоге rc?.d и скриптом из /etc/init.d/. Каждая ссылка будет начинаться с ‘S’ или ‘K’ с последующим номером и именем скрипта. Когда система входит в уровень выполнения N, из каталога /etc/rcN.d/ скрипты, начинающиеся с ‘K’, запускаются с аргументом stop, а потом оттуда же скрипты, начинающиеся с ‘S’, запускаются с аргументом start.

Например, можно настроить, чтобы скрипт foo выполнялся при загрузке, разместив его в каталог /etc/init.d/ и установив ссылки при помощи команды update-rc.d foo defaults 19. Аргумент defaults ссылается на уровни выполнения по умолчанию, которые могут быть от 2 до 5. Аргумент 19 обеспечивает, что скрипт foo вызывается до любых других с номером 20 или больше.

## 2.5 Поддержание многообразия программного обеспечения

Debian предлагает несколько путей удовлетворения любых потребностей системного администратора без нарушений в системе:

- dpkg-divert, смотрите 'The dpkg-divert command' на стр. 99.
- equivs, смотрите 'The equivs package' на стр. 99.
- update-alternative, смотрите 'Alternative commands' на стр. 99.
- make-kpkg поддерживает множество системных загрузчиков. Смотрите страницы make-kpkg(1) и 'Debian standard method' на стр. 101.

Любые файлы в каталоге /usr/local/ принадлежат системному администратору и Debian их не трогает. Большинство (или все) файлы в каталоге /etc являются конфигурационными файлами и Debian их не будет перезаписывать при обновлениях, пока системный администратор явным образом этого не запросит.

## 2.6 Интернационализация

Система Debian интернациональна и обеспечивает отображение и ввод символов на многих языках как в консоли, так в системе X Window. Множество документов, man-страницы и системные сообщения переведены на многие языки, и их (языков) число растет. В процессе установки Debian просит пользователя выбрать язык инсталляции ( и даже иногда его местную вариацию).

Если Ваша установленная система не поддерживает все нужные Вам языковые особенности, или Вам нужно изменить язык или установить другую раскладку клавиатуры для Вашего языка, смотрите 'Localization' на стр. 167.

## 2.7 Debian и ядро

Смотрите 'The Linux kernel under Debian' на стр. 101.

### 2.7.1 Компиляция ядра, полученного не из Debian

Необходимо понимать политку Debian в отношении заголовочных файлов (headers).

Библиотеки языка C в Debian собраны с самым свежим стабильным релизом заголовков ядра.

Например, релиз Debian 1.2 использовал версию заголовков 5.4.13. Эта практика отличается от той, что в пакетах исходных текстов ядра Linux, распространяемых на всех FTP-архивах Linux, где используются самые свежие версии заголовочных файлов. Заголовочные файлы ядра распространяются вместе с ядром и находятся в каталоге /usr/include/linux /include/.

Если Вам нужно скомпилировать программу с более новыми заголовочными файлами, чем те, что предоставляются пакетом libc6-dev, то Вы должны добавить опцию -I/usr/src/linux/include/ к Вашей командной строке при компиляции. Например, это важно при сборке пакета демона автомонтирования (пакет amd). Когда новые ядра изменяют некоторые внутренние особенности работы с NFS, программе amd нужно знать об этом. Это и требует включения самых последних заголовочных файлов.

### 2.7.2 Инструменты для построения ядер, настроенных пользователем

Пользователям, желающим собрать настроенное ими ядро, предлагается скачать пакет kernel-package. Данный пакет содержит скрипт для построения пакета с ядром и обеспечивает такую возможность посредством запуска команды:

```
# make-kpkg kernel_image
```

в каталоге верхнего уровня исходных текстов ядра. При помощи следующей команды можно получить справку:

```
# make-kpkg --help
```

также доступную на man-странице make-kpkg(1) и 'The Linux kernel under Debian' на стр. 101.

Пользователи должны кроме этого скачать исходный код наиболее свежего ядра (или же ядра, которое им требуется) с любого предпочитаемого ими архивного сайта Linux в случае недоступности пакета kernel-source-версия (где версия обозначает версию ядра). Загрузочный скипт initrd в Debian требует специального патча к ядру, называющийся initrd; смотрите http://bugs.debian.org/149236.

Подробные инструкции по использованию пакета kernel-package даны в файле /usr/share /doc/kernel-package/README.gz.

### 2.7.3 Специальное положение о том, как работать с модулями

Пакет modconf в Debian предоставляет скрипт командного интерпретатора (/usr/sbin /modconf), который можно использовать для настройки конфигурации модулей. Этот скрипт имеет меню-интерфейс, через который пользователь опрашивается об особенностях загружаемых драйверов устройств в системе. Указанная пользователем информация используется для настройки файла /etc/modules.conf (который содержит список алиасов и другие аргументы, используемые в сочетании с различными модулями), при этом анализируются файлы из каталога /etc/modutils/ и файл /etc/modules (содержащий список модулей, требуемых своей загрузки при старте системы).

Подобно файлам Configure.help, призванным помочь в создании пользовательских ядер, пакет modconf поставляется с набором файлов справки (в каталоге /usr/lib/modules_help/), которые предоставляют подробную информацию о соответствующих аргументах каждого модуля. Примеры смотрите в 'The modularized 2.4 kernel' на стр. 103.

### 2.7.4 Удаление из системы старого пакета с ядром

Скрипт kernel-image-NNN.prerm (запускаемый перед удалением пакета с ядром - прим. переводчика) проверяет на совпадение ядро, которое сейчас работает, с ядром, которое Вы пытаетесь удалить из системы. Таким образом, Вы можете безопасно удалять ненужные пакеты с ядром, используя данную команду:

```
dpkg --purge --force-remove-essential kernel-image-NNN
```

(Конечно же, замените NNN на номер версии и ревизии Вашего ядра.)

# Глава 3

# Советы по установке системы Debian

Официальная документация по установке Debian располагается по адресу http://www.debian.org/releases/stable/, и http://www.debian.org/releases/stable/installmanual.

Версии документов, находящихся в разработке, располагаются по адресу http://www.debian.org/releases/testing/, и http://www.debian.org/releases/testing/installmanual (поскольку работа еще продолжается, иногда документ может быть не доступен).

Хотя эта глава была изначально написана во время существования программы установки Potato, большая часть текста была обновлена для использования установщика Woody (кроме этого, они очень похожи). А так как Sarge будет использовать совершенно новую программу установки, используйте этот документ лишь как отправную точку для установщика Sarge.

## 3.1 Общие советы по установке системы Linux

Если Вы ищете компактный CD-образ установщика Debian, не забудьте изучить содержимое страницы http://www.debian.org/CD/netinst/.

Использование testing или unstable дистрибутивов Debian увеличивает риск столкнуться с серьезными ошибками. Этого можно избежать путем развертывания бутового загрузчика с возможностью выбора более стабильного дистрибутива Debian, или использования программы chroot для более стабильного дистрибутива (описано в 'chroot' на стр. 130). Это позволит выполнить одновременно разные виды Debian на разных консолях.

### 3.1.1 Основы совместимости аппаратного обеспечения

Linux совместим с множеством аппартного обеспечения персональных компьютеров и может быть установлен почти на любую систему. Для меня это также легко, как установить Windows 95/98/Me. Список совместимого аппаратного обеспечения постоянно растет.

Если Вы имеете ноутбук, изучите информацию Linux on Laptops (http://www.linux-laptop.net/) на предмет ссылок по установке системы на определенную модель выбранного производителя.

Моя рекомендация по выбору аппартного обеспечения для персонального компьютера - "Просто будь консервативным":

- Жесткие диски - на работе SCSI, для частного использования можно IDE/ATAPI.
- IDE/ATAPI CD-ROM (или CD-RW).
- платы PCI, а не ISA, особенно для сетевых плат (NIC).
- Используйте дешевые сетевые карты. Хороши Tulip для PCI, NE2000 для ISA.
- На ноутбуках в случае первой установки избегайте PCMCIA.
- Никаких USB клавиатур, мышей, ... если не хотите трудностей.

Если у Вас медленная машина, хорошей идеей будет снять жесткий диск и поключить его к другой более быстрой машине на момент установки.

### 3.1.2 Определение аппаратного обеспечения ПК и набора микросхем

В процессе установки задаются вопросы об оборудовании или наборе микросхем ПК. Иногда данную информацию не так просто найти. Вот один из методов:

1. Раскройте корпус Вашего ПК и загляните внутрь.
2. Запишите идентификационные коды продуктов на больших микросхемах видеокарты, сетевого адаптера, микросхемах около последовательных и IDE портов.
3. Запишите названия карт, напечатанных сзади на PCI и ISA картах.

### 3.1.3 Определение состава оборудования ПК при помощи Debian

Следующие команды на Linux-системе должны дать некоторую информацию об имеющемся оборудовании и его конфигурации.

```
$ lspci -v |pager
$ pager /proc/pci
$ pager /proc/interrupts
$ pager /proc/ioports
$ pager /proc/bus/usb/devices
```

Эти команды могут быть выполнены в процессе установки системы из экрана консоли, перейти на который можно нажатием Alt-F2.

Типовое использование прерываний:

- IRQ0: прерывание от таймера (8254)
- IRQ1: прерывание от контроллера клавиатуры
- IRQ2: применяется для каскадирования контроллера прерываний IRQ8–IRQ15 на PC-AT
- IRQ3: второй последовательный порт (io-port=0x2F8) (/dev/ttyS1)

- IRQ4: первый последовательный порт (io-port=0x3F8) (/dev/ttyS0)
- IRQ5: свободно (ранее использовалось для прерываний от жесткого диска на PC/XT)
- IRQ6: прерывание от контроллера гибких дисков (io-port=0x3F0) (/dev/fd0, /dev/fd1)
- IRQ7: паралелльный порт (io-port=0x378) (/dev/lp0)
- IRQ8: часы реального времени
- IRQ9: программное прерывание (int 0x0A), направленное на IRQ2
- IRQ10: свободно
- IRQ11: свободно
- IRQ12: мышь PS/2
- IRQ13: свободно (ранее использовалось 80287 математическим сопроцессором)
- IRQ14: первый IDE контроллер (/dev/hda, /dev/hdb)
- IRQ15: второй IDE контроллер (/dev/hdc, /dev/hdd)

Для устройств USB классы устройств описаны в /proc/bus/usb/devices в виде Cls=nn:
- Cls=00 : Не используется
- Cls=01 : Аудио (динамики и пр.)
- Cls=02 : Устройства связи (модем, сетевая плата, ...)
- Cls=03 : HID (Human Interface Device: клавиатуры, мыши, джойстики)
- Cls=07 : Принтер
- Cls=08 : Устройство хранения данных (дисководы, устройства CD/DVD, жесткие диски, флеш-карты, ...)
- Cls=09 : Концентратор (USB концентраторы)
- Cls=255 : тип определяется изготовителем

Если класс устройства не равен 255, Linux поддерживает это устройство.

### 3.1.4 Определение состава оборудования ПК при помощи других операционных систем

Информация об оборудовании может быть так получена от других ОС:

Установите другой коммерческий дистрибутив Linux. Пока определение оборудования на них лучше, чем в Debian. (Это положение должно быть улучшено при введении новой программы-установки debian-installer дистрибутива Sarge.)

Установите Windows. Конфигурация оборудования может быть получена нажатием правой кнопки мыши на иконку "Мой компьютер"с последующим входом в "Менеджер устройств". Запишите всю информацию об используемых ресурсах IRQ, адресах портов ввода/вывода и каналах DMA. Некоторые старые карты могут потребовать своей конфигурации в среде DOS и соответвствующего использования.

### 3.1.5 Миф о LILO

"Lilo ограничен 1024 цилиндрами." Неверно!

Новая версия lilo, используемая после выхода Debian Potato имеет поддержку lba32. Если BIOS Вашей материнской платы поддерживает lba32, lilo сможет работать на цилиндрах выше 1024.

Для этого добавьте строку "lba32" где-нибудь в начале Вашего файла lilo.conf, если у Вас используется конфигурационный файл от старого lilo. Смотрите /usr/share/doc/lilo /Manual.txt.gz.

### 3.1.6 GRUB

Новый системный загрузчик grub из проекта GNU Hurd может быть установлен в систему Debian Woody следующим образом:

```
# apt-get update
# apt-get install grub-doc
# mc /usr/share/doc/grub-doc/html/
... изучите содержимое
# apt-get install grub
# pager /usr/share/doc/grub/README.Debian.gz
... прочтите это:)
```

Для редактирования меню GRUB, правьте файл /boot/grub/menu.lst. Смотрите 'Setting GRUB boot parameters' на стр. 112 чтобы узнать, как установить загрузочные параметры при загрузке системы, так как конфигурирование GRUB немного отличается от конфигурирования lilo.

### 3.1.7 Выбор набора загрузочных дискет

В Potato мне нравился набор дисков IDEPCI для нормальной установки на рабочую станцию. Для Woody мне нравится загрузочный набор bf2.4. Они оба используют пакет boot-floppies для создания загрузочных дискет.

Если у Вас есть сетевая карта PCMCIA, Вам нужно использовать стандартный загрузочный набор дискет (самое большое количество дисков, зато доступны все модули драйверов) и настроить сетевую карту через настройку PCMCIA; не пытайтесь устанавливать сетевую карту через стандартный диалог настройки сети.

Для специфичных систем Вам возможно понадобится создать собственный спасательный диск. Это может быть сделано путем замены образа ядра под именем "linux" на спасательном диске Debian сжатым образом другого ядра, скомилированного где-либо для этой машины. Подробности описаны в файле readme.txt на таком диске. Спасательная дискета использует файловую систему MS-DOS, поэтому Вы можете использовать любую операционную систему, чтобы изучить и отредактировать ее содержимое. Это делает жизнь проще людям, у которых, например, нестандартная сетевая карта.

В Sarge для создания загрузочных дискет будет использоваться программа debian-installer и/или пакет pgi.

### 3.1.8 Установка

Следуйте официальным инструкциям, которые можно найти по адресу http://www.debian.org/releases/stable/installmanual или http://www.debian.org/releases/testing/installmanual (документ в разработке, иногда может быть не доступен).

Если Вы устанавливаете систему, используя программу boot-floppies из дистрибутива testing, то Вам, возможно, понадобится настроить источники системы APT. Для этого переключитесь на вторую виртуальную консоль нажатием Alt-F2 и вручную отредактируйте строки файла /etc/apt/sources.list, заменив "stable" на "testing"

Я придерживаюсь устанавливать lilo в места типа /dev/hda3, в то время как mbr устанавливаю в /dev/hda. Это минимизирует риск перезаписи загрузочной информации.

Вот что я выбираю в процессе установки системы:

- MD5 пароли "yes"
- Теневые пароли "yes"
- Установка "advanced" (через dselect **) и выбираю
  - Исключаю emacs (если он выбран), nvi, tex, telnet, talk(d);
  - Включаю mc, vim, один из редакторов nano-tiny или elvis-tiny. Изучите 'Программа dselect' на стр. 84. Даже если Вы поклонник Emacs, избегайте устанавливать его сейчас и довольствуйтесь nano при установке. Также избегайте устанавливать другие большие пакеты, как то TeX (актуально для Potato) на этой стадии. Смотрите 'Rescue editors' на стр. 207 относительно причин установки nano-tiny или elvis-tiny в этом месте.
- На все конфигурационные вопросы отвечаю "y" (замена текущего) во время диалога установки каждого пакета.
- exim: Выбираю "2" для своей машины, так как я отправляю почту через SMTP-сервер моего провайдера Интернет.

За дополнительной информацией по dselect обращайтесь к 'Программа dselect' на стр. 84.

### 3.1.9 Хосты и IP-адреса для использования в локальных сетях

Перед Вами пример конфигурации ЛВС (сеть класса C: 192.168.1.0/24):

```
Интернет
  |
  +--- Внешний провайдер интернет предоставляет POP-сервис (доступ по fetchmail)
  |
  Точка доступа провайдера предоставляет сервис DHCP и SMTP-сервис (с возможностью отправки почты перез него)
  |                    :
Кабельный модем       (Dialup)
  |                    :
Внешний интерфейс шлюза: eth0 (IP, выданный провайдером по DHCP)
 на базе старенького ноутбука (IBM Thinkpad, 486 DX2 50MHz, 20MB RAM)
```

```
  работает на ядре Linux 2.4 с файловой системой ext3.
  выполняет пакет "ipmasq" (с stronger-патчами, NAT и firewall)
  выполняет пакет "dhcp-client" настроенный на интерфейс eth0 (не принимает во внимание установку DNS)
  выполняет пакет "dhcp" настроенный на интерфейс eth1
  выполняет пакет "exim" в режиме смартхоста (режим 2)
  выполняет пакет "fetchmail" с большим интервалом (fallback)
  выполняет пакет "bind" в качестве кэширующего сервера имен Интернет для запросов из ЛВС и
                  в качестве авторитетного сервера имен домена ЛВС для запросов из ЛВС
  выполняет пакет "ssh" на порту 22 и 8080 (подключение с любого хоста)
  выполняет пакет "squid" в качестве кэширующего сервера архива Debian (для APT)
Внутренний интерфейс шлюза : eth1 (IP = 192.168.1.1, фиксированный)
                    |
      +--- Коммутатор (100base T) ---+
      |                              |
Несколько клиентов с            Несколько DHCP-клиентов в ЛВС
фиксированными IP-адресами          (IP = 192.168.1.128-200, динамические)
(IP = 192.168.1.2-127, фиксированные)
```

Смотрите 'Network configuration' на стр. 177 относительно деталей по настройке сети. Также смотрите 'Building a gateway router' на стр. 201 относительно конфигурирования сервер-шлюза.

### 3.1.10 Учетные записи пользователей

Чтобы свободно перемещаться между машинами, несколько первых учетных записей всегда одинаковы на моих системах.

Я всегда создаю первую пользовательскую учетную запись под именем "admin" (uid=1000). Почта root перенаправляется в почтовый ящик admin. Эта учетная запись помещается в группу adm (смотрите '"Why GNU su does not support the wheel group"' на стр. 138), которая получает неплохую долю привилегий root через команду su, используя PAM или через команду sudo. Подробности смотрите в 'Добавление пользовательской учетной записи' на стр. 49.

### 3.1.11 Создание файловых систем

#### Разбиение жесткого диска

Я предпочитаю использовать различные разделы для разных деревьев каталогов с целью уменьшения объема разрушений при системном сбое. Например:

```
/         == (/ + /boot + /bin + /sbin)
```

```
           == 50MB+
/tmp       == 100MB+
/var       == 100MB+
/home      == 100MB+
/usr       == 700MB+ с системой X Window
/usr/local == 100MB
```

Размер каталога /usr сильно зависит от присутствия в системе приложений X Window и документации к ним. /usr/ может быть около 300MB, если используются только терминальное ПО, или 2GB–3GB, если установлено много приложений Gnome. Когда /usr/ сильно разростается, то перенос каталога /usr/share/ на другой раздел будет наиболее эффективным средством. С новыми большими ядрами Linux 2.4, поставляемыми в виде пакетов, размер корневого каталога / может быть более 200MB.

Для примера приведу текущее состояние моего Интернет-шлюза (вывод команды df -h):

```
Filesystem     Size  Used Avail Use% Mounted on
/dev/hda3      300M  106M  179M  38% /
/dev/hda7      100M   12M   82M  13% /home
/dev/hda8      596M   53M  513M  10% /var
/dev/hda6      100M  834k   94M   1% /var/lib/cvs
/dev/hda9      596M  222M  343M  40% /usr
/dev/hda10     596M  130M  436M  23% /var/cache/apt/archives
/dev/hda11     1.5G  204M  1.2G  14% /var/spool/squid
```

(Большая область зарезервирована для /var/spool/squid/ - кэша прокси-сервера для кэширования скачиваемых пакетов.)

Далее идет вывод команды fdisk -l, предоставляющий структуру разделов диска:

```
# fdisk -l /dev/hda

/dev/hda1          1    41   309928+   6  FAT16 # DOS
/dev/hda2         42    84   325080   83  Linux # (не используется)
/dev/hda3   *     85   126   317520   83  Linux # основной
/dev/hda4        127   629  3802680    5  расширенный раздел
/dev/hda5        127   143   128488+  82  Linux swap
/dev/hda6        144   157   105808+  83  Linux
/dev/hda7        158   171   105808+  83  Linux
/dev/hda8        172   253   619888+  83  Linux
/dev/hda9        254   335   619888+  83  Linux
/dev/hda10       336   417   619888+  83  Linux
/dev/hda11       418   629  1602688+  83  Linux
```

Существует несколько неиспользующихся разделов. Они предназначены для установки второго дистрибутива Linux или для пространства расширения растущих деревьев каталогов.

Монтирование файловых систем

Монтирование ранее созданных файловых систем выполняется согласно файла /etc/fstab:

```
# /etc/fstab: статическая информация о файловых системах.
#
# файловая_система точка_монтирования тип опции_монтирования дамп проход
/dev/hda3 / ext2 defaults,errors=remount-ro 0 1
/dev/hda5 none swap sw 0 0
proc /proc proc defaults 0 0
/dev/fd0 /floppy auto defaults,user,noauto 0 0
/dev/cdrom /cdrom iso9660 defaults,ro,user,noauto 0 0
#
# keep partitions separate
/dev/hda7 /home ext2 defaults 0 2
/dev/hda8 /var ext2 defaults 0 2
/dev/hda6 /var/lib/cvs ext2 defaults 0 2
# параметр noatime повышает скорость чтения файлов
/dev/hda9 /usr ext2 defaults,noatime 0 2
/dev/hda10 /var/cache/apt/archives ext2 defaults 0 2

# очень большой раздел для кэша прокси-сервера
/dev/hda11 /var/spool/squid ext2 rw 0 2

# резервный загрузочный раздел DOS
/dev/hda1 /mnt/dos vfat rw,noauto 0 0
# резервный загрузочный раздел с системой Linux (не готова)
/dev/hda2 /mnt/linux ext2 rw,noauto 0 0
#
# монтирование nfs
mickey:/ /mnt/mickey nfs ro,noauto,intr 0 0
goofy:/ /mnt/goofy nfs ro,noauto,intr 0 0
# minnie:/ /mnt/minnie smbfs ro,soft,intr,credentials={filename} 0 2
```

Для NFS я использую опции noauto,intr, сочетаемые с опцией hard, которая стоит по умолчанию. Благодаря этому становится возможным восстановиться из повысшего процесса по причине умершего соединения посредством нажатия Ctrl-C.

Для соединений к Windows-машинам, подключенным через Samba (smbfs), разумно использовать опции rw,auto,soft,intr. Смотрите 'Настройка Samba' на стр. 41.

Для флоппи-дисководов использование набора опций noauto,rw,sync,user,exec предотвратит повреждение файлов в результате случайного извлечения дискеты до ее размонтирования, но замедлит запись на дискету.

Автоматическое монтирование

Ключевые моменты для автомонтирования:
- Загрузите модуль vfat, чтобы можно было в файле /etc/auto.misc указать -fstype=auto:
  # modprobe vfat # выполняется перед доступом к флоппи-диску
  ...либо, чтобы это было автоматически:
  # cat >>/etc/modules
  vfat
  ^D
  ... и перезагрузите систему.
- Отредактируйте файл /etc/auto.misc следующим образом:
  floppy -fstype=auto,sync,nodev,nosuid,gid=100,umask=000 :/dev/fd0
  ... где gid=100 соответствует группе "users".
- Создайте ссылки cdrom и floppy в каталоге /home/user, указывающие соответственно на /var/autofs/misc/cdrom и /var/autofs/misc/floppy.
- Добавьте user к группе ”users“.

Монтирование NFS

Внешний Linux NFS сервер (goofy) размещается за файерволом (шлюзом). У меня достаточно слабая политика безопасности в моей ЛВС, так как я единственный пользователь. Чтобы разрешить доступ по NFS, на стороне NFS сервера нужно добавить в файл /etc /exports следующее:

```
# /etc/exports: список управления доступом для файловых систем, которые
# можно экспортировать клиентам по NFS. Смотрите exports(5).
/       (rw,no_root_squash)
```

Это нужно для активации NFS сервера помимо установки пакетов NFS-сервера и клиентов.

Для простоты я обычно создаю один раздел 2 GB для экспериментов или для второй инсталляции Linux. По желанию можно сделать так, что обе установки будут разделять один раздел подкачки и раздел /tmp. A multipartition scheme is too involved for these usages. Если требуется только консольная система, 500MB будет вполне достаточно.

### 3.1.12 Рекомендации по оперативной памяти

Требования системы в оперативной памяти:

```
 4MB:  Минимум для работы ядра Linux.
16MB:  Минимум для приемлемой консольной системы.
32MB:  Минимум для простой системы X Window.
64MB:  Минимум для системы X Window с GNOME/KDE.
```

128MB: Для комфортной работы с системой X Window с GNOME/KDE.

256MB (и больше): А почему бы и нет, если Вы можете себе столько позволить? DRAM не такая дорогая.

Использование загрузочной опции mem=4m (или lilo append="mem=4m") покажет, как система будет работать на 4MB оперативной памяти. Данный загрузочный параметр необходим для систем с объемом оперативной памяти более 64MB со старым BIOS.

### 3.1.13 Пространство для подкачки

Я придерживаюсь следующим рекомендациям по распределению пространства для подкачки:

- Каждый раздел подкачки < 128MB (для старых ядер 2.0 ), < 2GB (для более свежих ядер)
- Общий размер пространства для подкачки = либо (1-2 величины объема установленной оперативной памяти) либо (128MB- 2GB) в качестве общей рекомендации
- Распределите этот объем по разным физическим дискам и монтируйте их все с опцией sw,pri=1 в файле /etc/fstab. Это гарантирует, что ядро выполнит RAID-чередование swap-разделов для максимальной производительности подкачки.
- По возможности используйте середину жесткого диска.

Даже если Вам не требуется подкачка, некоторый раздел подкачки все же желателен, например, чтобы оттянуть сбой системы с работающей программой, которая грешит утечками памяти.

## 3.2 Конфигурация командного интерпретатора Bash

Я изменяю стартовые скрипты по всей системе по своему вкусу таким образом:

| | |
|---|---|
| /etc/bash.bashrc | заменяю на свой |
| /etc/profile | Оставляю копию из дистрибутива ( \w -> \W) |
| /etc/skel/.bashrc | заменяю на свой |
| /etc/skel/.profile | заменяю на свой |
| /etc/skel/.bash_profile | заменяю на свой |
| ~/.bashrc | заменяю на свой для всех учетных записей |
| ~/.profile | заменяю на свой для всех учетных записей |
| ~/.bash_profile | заменяю на свой для всех учетных записей |

Смотрите подробности в моих примерных скриптах (http://www.debian.org/doc/manuals/debian-reference/examples//). Так как мне нравиться прозрачная система, я устанавливаю значение umask равное 002 или 022.

Переменная PATH устанавливается через следующие конфигурационные файлы в таком порядке:

```
/etc/login.defs  - перед тем, как оболочка устанавливает PATH
/etc/profile     (может вызывать /etc/bash.bashrc)
~/.bash_profile  (может вызывать  ~/.bashrc)
```

## 3.3 Настройка мыши

### 3.3.1 Мышь PS/2

В случае мыши на PS/2 на материнской плате ATX сделайте, чтобы сигналы от мыши проходили такой путь:

```
mouse -> /dev/psaux -> gpm -> /dev/gpmdata = /dev/mouse -> X
```

Здесь создается символическая ссылка /dev/mouse, указывающая на файл /dev/gpmdata, для того, чтобы нормально работали некоторые утилиты конфигурирования и чтобы было легче выполнять перенастройку. (Например, если Вы потом решите отказаться использовать демон gpm, то после этого просто настройте символическую ссылку /dev/mouse таким образом, чтобы она указывала на файл /dev/psaux.)

Такой путь сигналов от мыши/клавиатуры позволяет отсоединять и инициализировать их, повторно перезапуская gpm после подключения устройств. При этом система X Window будет нормально работать!

Протокол взаимодействия между gpm и X input может быть реализован одним из двух способов: либо как "ms3" (т.е. использовать протокол трехкнопочной мыши Microsoft последовательного порта), либо как сырой "raw"(использовать такой же протокол, как у подключенной мыши), причем данный выбор предопределяет выбор протокола, использующийся в X Window.

Продемонстрируем пример настройки трехкнопочной PS/2 мыши фирмы Logitech (традиционная мышь в Unix).

Если Вам не повезло, и Ваша видеокарта не поддерживается четвертой версией XFree, и Вы вынуждены использовать третью версию XFree (некоторые карты ATI 64), то для настройки в следующих примерах используйте файл /etc/X11/X86Config вместо /etc/X11/X86Config-4.

Подход с использованием протокола ms3

```
/etc/gpm.conf             | /etc/X11/X86Config-4
=========================+=========================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=ms3           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
```

```
append=""                 |  Option     "Device"    "/dev/mouse"
sample_rate=              |  Option     "Protocol" "IntelliMouse"
                          | EndSection
```

В данном случае тип мыши настраивается только в файле gpm.conf, а конфигурация X Window остается неизменной. Смотрите Примерные скрипты (http://www.debian.org/doc/manuals/debian-reference/examples//).

Подход с использованием сырого "raw" протокола

```
/etc/gpm.conf             | /etc/X11/X86Config-4
==========================+==========================================
device=/dev/psaux         | Section "InputDevice"
responsiveness=           |  Identifier "Configured Mouse"
repeat_type=raw           |  Driver     "mouse"
type=autops2              |  Option     "CorePointer"
append=""                 |  Option     "Device"    "/dev/mouse"
sample_rate=              |  Option     "Protocol" "MouseManPlusPS/2"
                          | EndSection
```

В данном случае тип мыши устанавливается как через файл gpm.conf, так и через файл конфигурации X.

Как настроить другую мышь

У программы gpm тип устройства autops2 предполагает автоопределение большинства типов мышей PS/2, существующих сейчас на рынке. К несчастью, это не только работает не всегда, но не было доступно в релизах до Woody. В таких случаях попробуйте в файле gpm.conf установить тип ps2 или imps2 вместо autops2. Чтобы выяснить, о каких мышках знает программа gpm, выполните: gpm -t help. Также смотрите gpm(8).

Если используется двухкнопочная PS/2 мышь, разрешите в X-протоколе эмуляцию третьей кнопки через Emulate3Buttons. Разница в протоколах между двухкнопочной и трехкнопочной мышкой определяется и настраивается автоматически программой gpm после однократного нажатия средней кнопки мыши.

Для X-протокола с 'Подход с использованием сырого "raw" протокола' на стр. 38 или без использования программы gpm используйте:

- IntelliMouse: мышь последовательного порта (gpm-повторитель протокола "ms3")
- PS/2: мышь порта PS/2 (сначала всегда используйте это)
- IMPS/2: любая мышь порта PS/2 (2, 3, мышь с прокруткой, получше)
- MouseManPlusPS/2: Мышь Logitech для порта PS/2
- ...

Дополнительно смотрите информацию на странице Поддержка мыши в XFree86 (http://www.xfree86.org/current/mouse.html).

Типовая мышь Microsoft с прокруткой работает лучше с такой настройкой:

```
/etc/gpm.conf           | /etc/X11/X86Config-4
=========================+===================================
device=/dev/psaux        | Section "InputDevice"
responsiveness=          |  Identifier "Configured Mouse"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "CorePointer"
append=""                |  Option     "Device"   "/dev/mouse"
sample_rate=             |  Option     "Protocol" "IMPS/2"
                         |  Option     "Buttons" "5"
                         |  Option  "ZAxisMapping" "4 5"
                         | EndSection
```

На некоторых ноутбуках Toshiba активирование программы gpm перед PCMCIA в скриптах начальной загрузки (System-V) помогает предотвратить зависания системы. Странно, но факт.

### 3.3.2 Мышь USB

Сначала убедитесь, что следующие функции ядра подключены либо в самом ядре, либо реализованы в виде модулей:

- В секции "Input core support":
  - "Input core support" (CONFIG_INPUT, input.o),
  - "Mouse support" (CONFIG_INPUT_MOUSEDEV, mousedev.o),
- В секции "USB support":
  - "Support for USB" (CONFIG_USB, usbcore.o),
  - "Preliminary USB device filesystem" (CONFIG_USB_DEVICEFS),
  - "UHCI" или "OHCI" (CONFIG_USB_UHCI || CONFIG_USB_UHCI_ALT || CONFIG_USB_OHCI, usb-uhci.o || uhci.o || usb-ohci.o),
  - "USB Human Interface Device (full HID) support" (CONFIG_USB_HID, hid.o), и
  - "HID input layer support" (CONFIG_USB_HIDINPUT)

Здесь имена, приведенные в нижнем регистре, являются именами модулей.

Если вы не используете devfs, то следующим образом создайте файл устройства /dev/input /mice с major 13 и minor 63:

```
# cd /dev
# mkdir input
# mknod input/mice c 13 63
```

Для типовой USB мыши с прокруткой должна использоваться следующая комбинация конфигурационных файлов:

```
/etc/gpm.conf           | /etc/X11/X86Config-4
=========================+===================================
```

```
device=/dev/input/mice   | Section "InputDevice"
responsiveness=          |  Identifier "Generic Mouse"
repeat_type=raw          |  Driver     "mouse"
type=autops2             |  Option     "SendCoreEvents" "true"
append=""                |  Option     "Device"   "/dev/input/mice"
sample_rate=             |  Option     "Protocol" "IMPS/2"
                         |  Option     "Buttons" "5"
                         |  Option  "ZAxisMapping" "4 5"
                         | EndSection
```

За дополнительной информацией смотрите страницу Проект Linux USB (http://www.linux-usb.org/).

### 3.3.3 Сенсорный коврик ноутбука

Хотя сенсорный коврик на ноутбуке по умолчанию эмулирует поведение двухкнопочной мышки PS/2, пакет tpconfig позволяет полностью контроллировать устройством. Например, установка опции OPTIONS="—tapmode=0" в файле /etc/default/tpconfig запретит надоедливое поведение "щелчок легким нажатием". Измените файл /etc/gpm.conf следующим образом, чтобы использовать как сенсорный коврик, так и внешнюю мышь USB для работы в консоли:

```
device=/dev/psaux
responsiveness=
repeat_type=ms3
type=autops2
append="-M -m /dev/input/mice -t autops2"
sample_rate=
```

## 3.4 Конфигурация NFS

Настройка NFS выполняется через файл /etc/exports.

```
# apt-get install nfs-kernel-server
# echo "/ *.domainname-for-lan-hosts(rw,no_root_squash,nohide)" \
      >> /etc/exports
```

Подробности смотрите в моих примерных скриптах (http://www.debian.org/doc/manuals/debian-reference/examples//).

## 3.5 Настройка Samba

Ссылки:
- http://www.samba.org/
- samba-doc пакет

Гораздо проще настраивать Samba для режима "share", когда создаются Windows for Workgroup- подобные расшаренные ресурсы. Однако, установка Samba в режим "user" является предпочтительной.

Конфигурация Samba выполняется или через debconf или в редакторе vi:

```
# dpkg-reconfigure --priority=low samba # в Woody
# vi /etc/samba/smb.conf
```

Подробности смотрите в моих примерных скриптах (http://www.debian.org/doc/manuals/debian-reference/examples//).

Добавление нового пользователя в файл smbpasswd может быть выполнена командой smbpasswd:

```
$ su -c "smbpasswd -a username"
```

Убедитесь, что Вы используете шифрованные пароли для максимальной совместимости.

Установите os level согласно следующей таблице эквивалентности систем (чем больше число, тем выше приоритет системы при выборах обозревателя сети):

```
0:    Samba с проигрышным атрибутом (никогда не станет главным обозревателем)
1:    WfW 3.1, Win95, Win98, Win/Me?
16:   Win NT WS 3.51
17:   Win NT WS 4.0
32:   Win NT SVR 3.51
33:   Win NT SVR 4.0
255:  Samba с максимальным приоритетом
```

Убедитесь, что пользователи являются членами группы, владеющей каталогом, предназначенным для совместного доступа , и что права на каталог включают право выполнения.

## 3.6 Настройка принтера

Традиционный подход использования принтера заключается в установке программ lpr/lpd. Существует также новая система печати - CUPS™ (Common UNIX Printing System). Еще существует PDQ. За дополнительной информацией обращайтесь к Linux Printing HOWTO (http://www.tldp.org/HOWTO/Printing-HOWTO.html).

### 3.6.1 lpr/lpd

Чтобы использовать систему печати lpr/lpd (пакеты lpr, lprng и gnulpr), отредактируйте файл (базовая настройка) /etc/printcap следующим образом (в случае использования принтера PostScript или чисто текстового):

```
lp|alias:\
        :sd=/var/spool/lpd/lp:\
        :mx#0:\
        :sh:\
        :lp=/dev/lp0:
```

Эти строки обозначают следующее:

- Заглавная строка: lp – имя спула, alias = алиас
- mx#0 – размер файла неограничен
- sh – не печатать заголовки разорванных страниц
- lp=/dev/lp0 – устройство локального принтера, или port@host для удаленного принтера

Это хорошая конфигурация, если у Вас подключен PostScript-принтер. Это также будет хорошо работать в случае печати с Windows-машины через Samba, на любом принтере, поддерживающимся Windows (правда, двунаправленная связь с принтером не поддерживается). Вам нужно только выбрать соответствующую конфигурацию принтера на Windows-машине.

Если же у Вас принтер не поддерживает PostScript, Вам нужно установить фильтрующую систему на базе программы gs. Существует масса инструментов для автоматической настройки печати с установкой файла /etc/printcap. Возможна любая из этих комбинаций:

- gnulpr, (lpr-ppd) и printtool— я использую эту.
- lpr и apsfilter
- lpr и magicfilter
- lprng и lprngtool
- lprng и apsfilter
- lprng и magicfilter

Чтобы запустить GUI-инструменты по настройке, такие как printtool, смотрите ‘Getting root in X’ на стр. 155 для получения привелегий root. Принтерные пулы, созданные программой printtool, используют gs и работают как PostScript-принтеры. Поэтому при доступе к ним, необходимо использовать драйверы принтера Postscript. В Windows таким стандартным принтером является "Apple LaserWriter“.

### 3.6.2 CUPS™

Установите систему печати Common UNIX Printing System (сокращенно CUPS™):

```
# apt-get install cupsys foomatic-bin foomatic-db
# apt-get install cupsys-bsd cupsys-driver-gimpprint
```

Затем настройте систему через web-браузер:

$ мой_браузер http://localhost:631

Например, чтобы добавить Ваш принтер на некоторый порт в список доступных принтеров, сделайте:

- нажмите "Printers" на главной странице, а там "Add Printer",
- Введите "root" в качестве имени пользователя и потом его пароль,
- перейдите к добавлению принтера, следуя подсказкам,
- вернитесь к странице "Printers", нажмите "Configure Printer", и
- перейдите к настройке размера бумаги, разрешения и других параметров.

За дополнительной информацией обращайтесь по адресу http://localhost:631/documentation.html и http://www.cups.org/cups-help.html.

## 3.7 Прочие советы по инсталляции системы

### 3.7.1 Установка дополнительных пакетов после первоначальной инсталляции

После первоначальной инсталляции системы Вы получаете небольшую, но уже функционирующую систему Debian. И это самый подходящий момент установить большее количество пакетов.

- Запустите программу tasksel. Изучите 'Установка задачи при помощи программ tasksel или aptitude' на стр. 83.

  В случае необходимости Вы можете выбрать следующее:
  - Для конечного пользователя – X Window System
  - Для разработки – C and C++
  - Для разработки – Python
  - Для разработки – Tcl/Tk
  - Прочее – TeX/LaTeX environment
  - Для всего остального я предпочитаю использовать программу tasksel как руководство, изучая состав компонентов, описанных в <Task Info>, для реализации некой задачи, и устанавливаю пакеты по отдельности через dselect.

- Запуск dselect.

  Первое, что Вы захотите слелать, это выбрать Ваш любымый редактор и программы, которые Вам нужны. Вы можете установить множество вариаций Emacs одновременно. Смотрите 'Программа dselect' на стр. 84 и 'Popular editors' на стр. 207.

  Также Вы можете заменить некоторые пакеты по умолчанию на пакеты с большими возможностями.

  – ...

- ...

Я обычно правлю файл /etc/inittab для простоты завершения работы системы следующим образом:

```
...
# Выполнить при нажатии CTRL-ALT-DEL.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -h now
...
```

### 3.7.2 Модули

Модули для драйверов устройств конфигурируются в процессе первоначальной инсталляции. Программа modconf предоставляет меню-подобный инферфейс для настройки модулей после инсталляции. Эта программа полезна, если некоторые модули отсутствовали в ходе первоначальной инсталляции или после установки нового ядра.

Все модули, которые необходимо загружать при старте системы, должны быть перечислены в файле /etc/modules. Я также использую программы lsmod и depmod для работы с модулями.

Также для ядер 2.4 добавьте несколько строк в файл /etc/modules для работы IP-маскарадинга (FTP и прочие). Смотрите ‘The modularized 2.4 kernel’ на стр. 103, в частности ‘Network function’ на стр. 104.

### 3.7.3 Базовая установка CD-RW

Отредактируйте следующие файлы:

```
/etc/lilo.conf  (добавьте append="hdc=ide-scsi ignore=hdc", (CDRW стоит мастером на втором канале - прим. переводчика)
                запустите lilo для вступления в силу изменений)
/dev/cdrom      (создайте символическую ссылку следующим образом
                # cd /dev; ln -sf scd0 cdrom
)
/etc/modules    (добавьте "ide-scsi" и "sg". И если нужно "sr" после них.)
```

Подробности смотрите в ‘CD writers’ на стр. 140.

### 3.7.4 Большой размер оперативной памяти и автовыключение питания

Отредактируйте файл /etc/lilo.conf следующим образом для установки загрузочных параметров большого объема ОЗУ (для ядер 2.2) и автовыключения питания (для систем с улучшенным управлением питанием APM)

```
append="mem=128M apm=on apm=power-off noapic"
```

Выполните lilo для установки этих значений. Параметр apm=power-off необходим для SMP-ядра, а noapic - чтобы избежать проблем с аппаратурой SMP, содержащей ошибки. Тоже самое может быть сделано путем прямого ввода опций перед загрузкой ядра. Смотрите ‘Other boot tricks with the boot prompt’ на стр. 111.

Если поддержка APM скомпилировна как модуль, что в Debian является по умолчанию для ядер 2.4, то после загрузки запустите команду insmod apm power_off=1 или для правки файла /etc/modules выполните:

```
# echo "apm power_off=1" >>/etc/modules
```

Кроме этого, на новых ядрах включение в ядре поддержки ACPI достигает той же цели и, кажется, является более дружественной к SMP (это требует более новых материнских плат). Также ядро 2.4. на новых материнских платах должно правильно определять большие объемы ОЗУ.

```
CONFIG_PM=y
CONFIG_ACPI=y
...
CONFIG_ACPI_BUSMGR=m
CONFIG_ACPI_SYS=m
```

и добавьте следующие строки в файл /etc/modules в следующем порядке:

```
ospm_busmgr
ospm_system
```

Либо перекомпилируйте ядро со всеми этими опциями, установленными на значение ”y“. В любом случае для ACPI никакие загрузочные параметры не нужны.

### 3.7.5 Странные проблемы доступа на некоторые web-сайты

Недавние ядра Linux по умолчанию разрешают ECN, что может привести к проблемам доступа к некоторым web-сайтам, находящимся за неправильными маршрузаторами. Чтобы проверить статус ECN:

# cat /proc/sys/net/ipv4/tcp_ecn
... или
# sysctl net.ipv4.tcp_ecn

Для выключения ECN используйте:

# echo "0" > /proc/sys/net/ipv4/tcp_ecn
... или
# sysctl -w net.ipv4.tcp_ecn=0

Чтобы запретить TCP ECN при каждой загрузке системы добавьте в файл /etc/sysctl.conf следующее:

net.ipv4.tcp_ecn = 0

### 3.7.6 Конфигурация Dialup PPP

Установите пакет pppconfig, чтобы настроить dialup PPP.

# apt-get install pppconfig
# pppconfig
... следуйте инструкциям для конфигурирования dialup PPP
# adduser user_name dip
... разрешает пользователю user_name доступ к dialup PPP

Включение пользователем (user_name) dialup PPP доступа к интернету :

$ pon Имя_провайдера # старт PPP-доступа к Вашему провайдеру интернет
... наслаждайтесь интернетом
$ poff ISP_name # останавливает PPP-доступ, Имя_провайдера указывать не обязательно

Подробности смотрите в /usr/share/doc/ppp/README.Debian.gz.

Альтернативно можно использовать пакет wvdial для установки dialup PPP-доступа. Обратите внимание на существование хорошо известной ошибки http://bugs.debian.org/82095, иногда предотвращающей работу с dialup PPP обычных пользователей.

Все программы дозвона используют демон pppd, который запускает программы, найденные в /etc/ppp/ip-up.d/ после подключения к провайдеру и все программы, найденные в /etc /ppp/ip-down.d/ после отключения от провайдера. Это можно использовать для скачивания и отправки почты.

### 3.7.7 Другие конфигурационные файлы для настройки из каталога /etc/

Вы можете захотеть добавить файл /etc/cron.deny, отсутствующй в стандартной установке Debian (можно скопировать файл /etc/at.deny).

# Глава 4

# Руководство по Debian

Эта секция поможет новичкам ориентироваться в мире Debian. Если Вы уже хотя бы немного использовали Unix-подобные операционные системы, то, вероятно, знакомы со всем тем, о чем я здесь пишу. Пожалуйста, используйте эту секцию для проверки своих знаний.

## 4.1 Введение

После установки системы Debian на Ваш ПК Вам нужно изучить несколько полезных вещей. Я предлагаю это сделать в форме экспресс-тренировки.

### 4.1.1 Вход в систему как пользователь root

Сразу после перезагрузки системы Вам представляется либо графический экран входа в систему, либо текстовый - в зависимости от изначального выбора пакетов при установке. Для простоты, если Вы видите графический экран для входа, нажмите Ctrl-Alt-F1, чтобы перейти к текстовому экрану входа в систему. [1]

Предположим, что имя Вашей системы - foo, тогда приглашение на вход будет таким:

```
foo login:
```

Наберите root, нажмите клавишу Enter и введите пароль, который Вы указывали в процессе установки. В системе Debian согласно традиций Unix пароль чувствителен к регистру. После этого система выводит приветствие и представляет Вам приглашение командой строки пользователя root, ожидая ввода. [2]

```
foo login: root
```

[1] Клавиши Левый-Ctrl, Левый-Alt, и F1 нажимаются одновременно.

[2] Заметьте, приветствие может быть другим, если Вы редактировали файл /etc/motd.

```
Password:
Last login: Sun Oct 26 19:04:09 2003 on tty3
Linux foo 2.4.22-1-686 #6 Sat Oct 4 14:09:08 EST 2003 i686 GNU/Linux

Most of the programs included with the Debian GNU/Linux system are
freely redistributable; the exact distribution terms for each program
are described in the individual files in /usr/share/doc/*/copyright

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

root@foo:root#
```

Теперь Вы готовы к тому, чтобы заниматься системным администрованием через командую строку root. Учетная запись root также называется суперпользовательской или привелегированной. Используя эту учетную запись, Вы можете:

- Читать, записывать и удалять любые файлы в системе, независимо от установленных на них прав доступа
- Устанавливать владельцов и права доступа на любой файл в системе
- Устанавливать пароль любому непривелегированному пользователю в системе
- Регистрироваться в системе под любой учетной записью без ввода соответствующего пароля

Если Вам нужно дать кому-либо часть полномочий root, не стоит этого делать, разделяя с ним пароль учетной записи root. Используйте для этого программы типа sudo(8).

В Unix считается хорошей привычкой регистрироваться в системе, используя непривелегированную учетную запись, даже если Вы собираетесь выполнять некую административную работу. Используйте команды sudo, super, или su -c для получения дозированных привилегий root, когда это необходимо. Смотрите 'Working more safely – sudo' на стр. 139. [3]

### 4.1.2 Установка минимальной среды окружения для новичка

Я думаю, что изучение компьютерной системы схоже с изучением нового иностранного языка. И хотя учебники достаточно полезны, Вам нужно попрактивать с подручными инструментами. В этом контексте я считаю, что хорошей идей будет установка нескольких дополнительных пакетов типа mc, vim, lynx, doc-linux-text и debian-policy. [4]

```
# apt-get update
 ...
# apt-get install mc vim lynx doc-linux-text debian-policy
 ...
```

[3] Нужно сказать, что я работал под root чаще, чем это требовалось, так как это было проще, да и я был небрежен в этом отношении.

[4] Также будет неплохо установить пакеты gpm, emacs21 и doc-linux-html. Смотрите 'Настройка мыши' на стр. 37 и 'Editors' на стр. 207.

Если эти пакеты уже имеются в системе, то ничего не будет установлено.

### 4.1.3 Добавление пользовательской учетной записи

В процессе установки Вы обычно создаете учетную запись непривилегированного пользователя, который будет получать электронную почту, адресованную root. [5] Если Вы не хотите использовать эту специальную учетную запись для последующего обучения, Вам нужно создать еще одну пользовательскую учетную запись.

Предположим, что Вы захотели, чтобы ее имя было penguin. Тогда следущая команда:

```
root@foo:root# adduser penguin
... ответьте на все вопросы
```

создаст ее. [6] Перед тем, как идти дальше, мы должны сперва изучить еще несколько вещей.

### 4.1.4 Переключение между виртуальными консолями

В системе Debian по умолчанию существует шесть независимых превдотерминалов, т.е. Вы можете использовать текстовый консольный экран VGA Вашего персонального компьютера как 6 переключаемых терминалов VT-100. Переключение между терминалами осуществляется одновременным нажатием клавиши левый-Alt и одной из функциональных F1–F6. Каждый псевдотерминал позволяет независимую регистрацию в системе. Это является проявлением такой важной черты Unix, как многопользовательность.

Если Вы случайно нажали Alt-F7 на системе с работающей системой X Window и видите графический экран, то для возврата к текстовым консолям нажмите Ctrl-Alt-F1. Для тренировки просто попытайтесь попереключаться на другую консоль и возвращаться назад.

### 4.1.5 Как завершить работу системы

Как любая современная операционная система, где файловые операции используют кэширование данных в памяти, система Debian нуждается в соответствующей процедуре завершения работы системы перед тем, как питание может быть безопасно выключено без повреждения целосности файлов. Используйте следующую команду в командой строке root для завершения работы системы:

```
# shutdown -h now
```

[5] Я стараюсь задавать имя этого пользователя как admin, но оно может быть произвольным.

[6] Возможно, Вы захотите добавить этого пользователя penguin в группу adm, чтобы разрешить чтение многих журналов событий (лог-файлов) в каталоге /var/log/. Смотрите также passwd(5), group(5), shadow(5), group(5), vipw(8) и vigr(8). За официальным описанием пользователей и групп смотрите свежую версию документа Пользователи и Группы (/usr/share/doc/base-passwd/users-and-groups.html).

Эта команда используется в нормальном многопользовательском режиме. Если же Вы в однопользовательском режиме, то в командной строке root используйте следующую команду:

# poweroff -i -f

Также Вы можете нажать Ctrl-Alt-Delete для завершения работы системы. [7]

Подождите, пока система не отобразит сообщение "System halted", и только тогда выключайте питание. Если функции APM или ACPI надлежащим образом включены как в BIOS, так и в Linux, то система выключит питание самостоятельно. Подробности смотрите в 'Большой размер оперативной памяти и автовыключение питания' на стр. 45.

### 4.1.6 Время поразвлечься

Теперь Вы готовы без риска поиграть с системой Debian, так как используете учетную запись непривилегированного пользователя penguin. [8]

Давайте войдем в систему как пользователь penguin. Если Вы сейчас работаете как root, нажмите Ctrl-D [9] в командой строке, чтобы закрыть командный интерпретатор root'а и выйти на приглашение к входу в систему. Введите Ваше только что созданное имя пользователя penguin и его пароль. [10] Вам будет показано следующее приглашение ко вводу команд.

penguin@foo:penguin$

С этого момента для простоты даваемые примеры будут использовать упрощенное приглашение ко вводу команд. Я буду использовать:

- # : для обозначения ввода команд от имени root
- $ : для обозначения ввода команд от имени непривилегированного пользователя

Мы начнем изучение системы Debian сначало по-простому через 'Midnight Commander (MC)' на стр. 51, а позже по-настоящему через 'Традиционная для Unix рабочая среда' на стр. 54.

[7] Левая клавиша Ctrl, левая клавиша Alt и клавиша Delete нажимаются вместе, когда Вы находитесь в консоли. В системе с настройками по умолчанию это приведет к перезагрузке. Вам нужно изменить файл /etc/inittab, чтобы использовать опцию -h команды shutdown, как это было описано в разделе 'Установка дополнительных пакетов после первоначальной инсталляции' на стр. 43.

[8] Это потому, что система Debian даже после установки по умолчанию настроена с такими правами доступа к файлам, которые предотвращают повреждение системы непривилегированным пользователем. Конечно, все же могут существовать некоторые дыры, которые могут использованы (для взлома системы - прим. переводчика), так что те, кому эти вопросы небезразличны, должны вместо чтения этой секции изучить документ Руководство по безопасности Debian (http://www.debian.org/doc/manuals/securing-debian-howto/).

[9] Левая клавиша Ctrl и клавиша d нажимаются вместе. При этом нет необходимости нажимать клавишу Shift, хотя эти управляющие символы даются в форме с верхним регистром буквы "D".

[10] Если Вы ввели root вместо penguin и указали соответствующий пароль, то Вы получите доступ к учетной записи root. Эта процедура будет нужна для получения прав root.

## 4.2 Midnight Commander (MC)

Midnight Commander (MC) - это "Швейцарский армейский нож" от проекта GNU, применяемый в консоли Linux и прочих терминальных средах. Он дает новичку опыт работы в консоли через меню-подобный интерфейс, который гораздо проще изучить, чем стандартные команды Unix.

Используйте эту программу для изучения системы Debian. Это самый лучший способ для изучения. Пожалуйста, исследуйте следующие несколько ключевых мест, используя клавиши управления курсором и Enter:
- /etc и его подкаталоги.
- /var/log и его подкаталоги.
- /usr/share/doc и его подкаталоги.
- /sbin и /bin

### 4.2.1 Улучшим MC

Чтобы MC изменял рабочий каталог при выходе из него, Вам нужно изменить файл ~/.bashrc (или /etc/bash.bashrc, вызываемый из .bashrc) образом, описанным на соответствующей man-странице mc(1) касательно опции -P. [11]

### 4.2.2 Запуск MC

```
$ mc
```

MC выполняет все операции с файлами через его меню, требуя от пользователя минимум усилий. Для получения справки просто нажмите F1. Вы можете поиграть с MC путем нажатий клавиш управления курсором и функциональных клавиш. [12]

### 4.2.3 Файловый менеджер в MC

По умолчанию отображаются две панели каталога со списками файлов. Еще один полезный режим - это когда правое окно отображает информацию о режиме доступа к файлу и пр. Далее описаны несколько важных клавишных комбинаций. С демоном gpm можно также использовать мышь. (Обязательно нажмите клавишу Shift, чтобы получить нормальное поведение операций cut и paste в MC.)
- F1: Меню справки
- F3: Встроенный просмотрщик файлов
- F4: Встроенный редактор
- F9: Вывести выпадающее меню
- F10: Выход из Midnight Commander

[11] Если Вы пока не поняли точно, что я имею в виду, можно сделать это позже.

[12] Находясь в терминалах типа kon или kterm для японского языка, которые имеют проблемы с некоторыми графическими символами, поэтому добавьте опцию -a при запуске MC, чтобы разрешить эти проблемы.

- Tab: Переход между панелями
- Insert: Пометить файл для операций с множеством файлов, например, для копирования
- Del: Удалить файл (будьте осторожны — установите MC в безопасный режим удаления)
- Клавиши управления курсором: понятно из названия

### 4.2.4 Особенности командной строки в MC

- Любая команда cd изменяет каталог, отображаемый на выбранной панели.
- Ctrl-Enter или Alt-Enter копирует имя файла в командную строку. Используйте это для команд cp или mv совместно с редактированием командной строки.
- Alt-Tab покажет список выбора командного интерпретатора с возможными именами файлов.
- Можно задать стартовые каталоги для обеих панелей через аргументы MC; например так: mc /etc /root.
- Esc + цифра == Fn (т.е., Esc + '1' = F1, и так далее; Esc + '0' = F10)
- Клавиша Esc == Клавише Alt (= Meta, M-); т.е., ввод Esc + 'c' соответствует Alt-C.

### 4.2.5 Редактор в MC

Встроенный редактор имеет интересную систему "вырезать-вставить". Нажатие F3 помечает начало выделения текста, повторное нажатие помечает конец выделения и высвечивает выбранный текст. Теперь Вы можете передвигать Ваш курсор. И если нажмете F6, то выделенная область будет перенесена в позицию курсора. Если же Вы нажмете F5, то выделенная область будет скопирована в позицию курсора. F2 сохраняет файл, F10 завершает работу редактора. Большинство курсорных клавиш работают так, как Вы интуитивно преполагаете.

Этот редактор может быть запущен непосредственно для работы с файлом:

```
$ mc -e имя_файла_для_правки
$ mcedit имя_файла_для_правки
```

Данный редактор не является мультиоконным, но можно использовать множество линукс-консолей для достижения похожего эффекта. Чтобы копировать между окнами, для переключения между виртуальными консолями используйте клавиши Alt-Fn, а для переноса части файла в другой используйте "Файл->Вставить файл" или "Файл->Копировать в файл"

Встроенный редактор можно заменить любым внешним по Вашему выбору.

Также многие программы используют переменную окружения EDITOR или VISUAL для определения, какой редактор использовать. Если Вам неудобно в vim, установите эти переменные на mcedit, добавив следующие строки в файл ~/.bashrc:

```
...
export EDITOR=mcedit
export VISUAL=mcedit
...
```

По возможности я рекомендую устанавливать эти переменные на vim. Использование редактора vim предпочтительнее потому, что он (а точнее vi) всегда существует на любой Linux/Unix системе. [13]

### 4.2.6 Программа просмотра в MC

Представляет собой очень интеллектуальную программу. Это великолепный инструмент для поиска слов в документах. Я использую эту возможность для файлов в каталоге /usr /share/doc. Это самый быстрый способ просмотреть множество информации по Линукс. Программа просмотра может быть непосредственно запущена таким образом:

```
$ mc -v имя_файла_для_просмотра
```

### 4.2.7 Возможности MC по автоматическому запуску программ

Нажмите клавишу Enter на файле и соответствующая программа будет обрабатывать содержимое файла. Это очень удобная возможность MC.

выполняемый файл: Выполняет команду
файл man, html-страницы: Направляет содержимое в программу просмотра
файл tar.gz, deb: Отображает содержимое файла как подкаталог

Чтобы позволить функционировать этим возможностям просмотра, просматриваемые файлы не должны быть выполняемыми. Изменить их статус можно или через программу chmod или через меню MC.

### 4.2.8 Виртуальная FTP-файловая система MC

MC можно использовать для доступа через Интернет к файлам, используя протокол FTP. Перейдите в меню, нажав F9, затем нажмите 'p' для активации виртуальной файловой системы FTP (в моей версии MC работает клавиша 'F' - прим. переводчика) . Введите URL в форме имя_пользователя:пароль@имя_хоста.имя_домена, после чего будет осуществлена выборка удаленного каталога, который будет представлен как локальный.

Попробуйте в качестве URL указать http.us.debian.org/debian и просмотреть файловый архив Debian. Смотрите 'Архивы Debian' на стр. 5 относительно того, как они (архивы) организованы.

[13] Н самом деле, программы, которые есть везде, - это vi или nvi. Для новичков же я предлагаю vim, так как он предоставляет справку по клавише F1, оставаясь при этом достаточно простым и очень мощным.

## 4.3 Традиционная для Unix рабочая среда

Хотя MC позволяет делать Вам практически все, очень важно изучить, как использовать инстументы командной строки, вызываемые из приглашения командного интерпретатора и познакомиться с рабочей средой, традиционной для Unix. [14]

### 4.3.1 Специальные комбинации клавиш

В традиционной unix-среде существует несколько комбинаций клавиш, которые имеют специальный смысл. [15]

- Ctrl-U: Удалить строку перед курсором.
- Ctrl-H: Удалить символ перед курсором.
- Ctrl-D: Завершить ввод. (при применении в командном интерпретаторе - выход из него)
- Ctrl-C: Завершить выполняющуюся программу.
- Ctrl-Z: Временно остановить программу. (поместить ее в фоновое задание, смотрите 'команда &' на стр. 61)
- Ctrl-S: Остановить вывод на экран. [16]
- Ctrl-Q: Продолжить вывод на экран.

Командный интерпретатор по умолчанию, bash, имеет возможности по хранению истории команд командной строки и tab-completion для повышения интерактивности.

- стрелка вверх: начинает поиск в истории команд.
- Ctrl-R: начинает инкрементальный поиск в истории команд.
- TAB: Завершает ввод имени файла в командной строке.
- Ctrl-V TAB: Ввод символа табуляции TAB в командую строку без expansion.

Запомните еще несколько важных комбинаций клавиш:

- Ctrl-Alt-Del: Перезагрузка/останов системы, смотрите 'Установка дополнительных пакетов после первоначальной инсталляции' на стр. 43.
- Левая клавиша мыши совместно с передвижением мыши: выделение и копирование текста в буфер обмена.
- Нажатие средней кнопки мыши: Вставить текст из буфера обмена в позицию курсора.
- Meta-клавиша (в терминологии Emacs) назначена по традиции на левую клавишу Alt. Некоторые системы могут быть настроены использовать Windows-клавишу в качестве Meta-клавиши.

Чтобы использовать мышь в текстовой консоли Linux, Вам нужно иметь запущенный демон gpm. [17] Смотрите 'Настройка мыши' на стр. 37.

---

[14] В этой части руководства в качестве командного интерпретатора используется bash. О том, какие еще бывают командные интерпретаторы, смотрите в 'Shell' на стр. 221.

[15] На нормальной текстовой консоли Linux только левая клавиша Ctrl и левая клавиша Alt работают так, как ожидается.

[16] Вы можете запретить эту возможность, используя stty(1).

[17] В среде X Window мышь в программе xterm работает так же.

### 4.3.2 Основные команды Unix

Давайте изучим основные команды Unix. [18] Выполните все следующие команды от имени непривелегированного пользователя penguin:

- pwd
  - Отобразить имя текущего/рабочего каталога.
- whoami
  - Отобразить имя текущего пользователя.
- file foo
  - Отобразить тип файла foo.
- type -p имя_команды
  - Отобразить месторасположение файла команды имя_команды.
  - Это же делает команда which имя_команды.
- type имя_команды
  - Отобразить информацию по команде имя_команды.
- apropos ключевое_слово
  - Найти команды, связанные с ключевым_словом.
  - Это же делает команда man -k ключевое_слово.
- whatis имя_команды
  - Отобразить однострочное пояснение по команде имя_команды.
- man -a имя_команды
  - Отобразить краткое пояснение по команде имя_команды. (Unix-стиль)
- info имя_команды
  - Отобразить более полное объяснение по команде имя_команды. (стиль GNU)
- ls
  - Вывести содержимое каталога. (нескрытые файлы и каталоги) [19]
- ls -a
  - Вывести содержимое каталога. (все файлы и каталоги)
- ls -A
  - Вывести содержимое каталога. (почти все файлы и каталоги, т.е. пропускаются файлы "..“ и ".“)
- ls -la
  - Вывести все содержимое каталога с подробной информацией. Смотрите ‘Концепция файловой системы в Debian’ на стр. 68.
- ls -d
  - Выводит все подкаталоги каталоги в текущем каталоге.
- lsof foo
  - Вывести состояние файла foo с точки зрения - открыт он или нет.
- mkdir foo
  - Создать новый каталог foo в текущем каталоге.
- rmdir foo

[18] Здесь я использую имя "Unix“ в самом общем смысле. Любой клон Unix обычно предлагает эквивалентные команды. Система Debian - не исключение. Не расстраивайтесь, если некоторые команды не работают так, как Вы хотите. И не обязательно выполнять данные примеры в указанном порядке.

[19] В Unix есть традиция скрывать имена файлов, которые начинаются с ".“. Это, как правило, файлы, которые содержат конфигурационную информацию и пользовательские настройки.

  - Удалить каталог foo в текущем каталоге.
- cd foo
  - Перейти в каталог foo, находящийся в текущем каталоге или описанный в переменной CDPATH. Смотрите описание команды cd в builtins(7).
- cd /
  - Перейти в корневой каталог.
- cd
  - Перейти в домашний каталог текущего пользователя.
- cd /foo
  - Перейти в каталог по полному пути /foo.
- cd ..
  - Перейти в родительский каталог.
- cd ~foo
  - Перейти в домашний каталог пользователя foo.
- cd -
  - Перейти в предыдущий каталог.
- </etc/motd pager
  - Отобразить содержимое файла /etc/motd, используя программу постраничного вывода по умолчанию. Смотрите 'команда < foo' на стр. 62. [20]
- touch junkfile
  - Создать пустой файл junkfile.
- cp foo bar
  - Копировать существующий файл foo в новый файл bar.
- rm junkfile
  - Удалить файл junkfile.
- mv foo bar
  - Переименовать существующий файл foo в новый файл bar.
- mv foo bar/baz
  - Переместить существующий файл foo в новое месторасположение под новым именем bar/baz. Каталог bar/ должен существовать.
- chmod 600 foo
  - Сделать существующий файл foo недоступным для чтения и записи для всех пользователей, кроме владельца, и запретить выполнение для всех пользователей.
- chmod 644 foo
  - Разрешить чтение и запретить запись в существующий файл foo всем пользователям, кроме владельца, выполнение запрещено всем.
- chmod 755 foo
  - Разрешить чтение и запретить запись в существующий файл foo всем пользователям, кроме владельца, выполнение файла разрешено всем пользователям.
- top
  - Показать информацию о процессах на весь экран. Используйте "q" для выхода.

[20] Функцию постраничного вывода по умолчанию в системе Debian выполняет программа more, которая не может листать текст назад. Установив пакет less командой apt-get install less, less станет программой постраничного вывода по умолчанию и Вы сможете листать текст назад, используя клавишу управления курсором.

- ps aux | pager
  - Показать информацию о всех выполняющихся процессах в стиле вывода системы BSD. Смотрите 'команда1 | команда2' на стр. 62.
- ps -ef | pager
  - Показать информацию о всех выполняющихся процессах в стиле вывода системы Unix System-V.
- ps aux | grep -e "[e]xim"
  - Показать все выполняющиеся процессы exim. Регулярные выражения для поиска можно изучить на man-странице grep(1), получить которую можно, набрав man grep.
- ps axf | pager
  - Показать информацию о всех выполняющихся процессах, используя вывод в стиле "ASCII art".
- kill 1234
  - Убить процесс по его идентификатору процесса: 1234. Смотрите 'Kill a process' на стр. 116.
- grep -e "образец" *.html
  - Найти "образец" во всех файлах, заканчивающихся на .html в текущем каталоге и показать их всех.
- gzip foo
  - Сжать файл foo и создать файл foo.gz, используя кодирование Lempel-Ziv (LZ77).
- gunzip foo.gz
  - Разжать файл foo.gz для создания файла foo.
- bzip2 foo
  - Сжать файл foo и создать файл foo.bz2, используя алгоритм "Burrows-Wheeler block sorting text compression algorithm" и Huffman-кодирование. (Обеспечивает лучшее сжатие, чем gzip)
- bunzip2 foo.bz2
  - Разжать файл foo.bz2 для создания файла foo.
- tar -xvvf foo.tar
  - Извлечь файлы из архива foo.tar.
- tar -xvvzf foo.tar.gz
  - Извлечь файлы из архива foo.tar.gz, сжатого gzip.
- tar -xvvf --bzip2 foo.tar.bz2
  - Извлечь файлы из архива foo.tar.bz2. [21]
- tar -cvvf foo.tar bar/
  - Архивировать содержимое каталога bar/ в архивный файл foo.tar.
- tar -cvvzf foo.tar.gz bar/
  - Архивировать содержимое каталога bar/ в сжатый архивный файл foo.tar.gz.
- tar -cvvf --bzip2 foo.tar.bz2 bar/
  - Архивировать содержимое каталога bar/ в сжатый архивный файл foo.tar.bz2. [22]
- zcat README.gz | pager
  - Отобразить содержимое сжатого файла README.gz, используя программу по-

[21] Здесь используется опция --bzip2 вместо новой короткой опции -j, чтобы команда работала со старыми версиями программы tar в Potato.

[22] Опция --bzip2 здесь также используется для совместимости.

страничного вывода по умолчанию.

- zcat README.gz > foo
  - Создать файл foo с разжатым содержимым файла README.gz.
- zcat README.gz >> foo
  - Добавить разжатое содержимое файла README.gz в конец файла foo. (Если файл не существует, то сначала создать его.)
- find . -name шаблон
  - Искать имена файлов, соответствующие шаблону шаблон. (медленно)
- locate -d . шаблон
  - Искать имена файлов, соответствующие шаблону шаблон. (Работает быстрее благодаря использованию регулярно генерируемой базы данных)

Пожалуйста, походите по каталогам и изучите систему, используя ранее описанные команды, для тренировки. Если у Вас есть вопросы по любой из консольных команд, будьте добры прочесть соответствующую man-страницу. Например, данные команды хороши для старта:

```
$ man man
$ man bash
$ man ls
```

Также неплохо запустить vim и нажать клавишу F1. Вам нужно прочесть не менее 35 строк. Затем пройдите онлайновый курс обучения, передвинув курсор к |tutor| и нажав Ctrl-]. Смотрите 'Editors' на стр. 207 для дополнительного изучения редакторов.

Пожалуйста заметьте, что многие Unix-команды, включая из GNU и BSD, будут показывать краткую справочную информацию, если Вы вызовете их одним из следующих способов (или, в некоторых случаях, без аргументов вовсе):

```
$ имя_команды --help
$ имя_команды -h
```

Для самоподготовки также посмотрите примеры из 'Debian tips' на стр. 109.

### 4.3.3 Выполнение команд

Теперь Вы имеете представление о том, как использовать систему Debian. Давайте внимательно взглянем на механизм выполнения команд в системе Debian. [23]

### 4.3.4 Простая команда

Простая команда - это последовательность, состоящая из:

1 необязательных присвоений переменных

[23] Здесь я для новичков несколько упрощаю действительность. Для точного описания смотрите bash(1).

2 имени команды
3 необязательных аргументов
4 необязательных перенаправлений (> , >> , < , << и пр.)
5 необязательного оператор контроля (&& , || ; <newline> , ; , & , ( , ) )

Для более сложных команд с кавычками и заменами смотрите ‘Command-line processing’ на стр. 225.

### 4.3.5 Выполнение команды и переменные окружения

Рассмотрим следующую типовую последовательность команд: [24]

```
$ date
Sun Oct 26 08:17:20 CET 2003
$ LC_ALL=fr_FR date
dim oct 26 08:17:39 CET 2003
```

Здесь выполняется программа date. Переменная окружения LC_ALL:

- для первой команды не установлена (равна значению по умолчанию в системе; тоже самое, что и C)
- для второй команды установлена в fr_FR (французская локаль)

Обычно при выполнении команд определение переменных окружения отсутствует. Для выше приведенного примера Вы можете сделать и так:

```
$ LC_ALL=fr_FR
$ date
dim oct 26 08:17:39 CET 2003
```

Как Вы видете, вывод команды зависит от переменной окружения. Если же Вы хотите, чтобы переменная окружения наследовалась дочерними процессами (например, когда Вы запускаете скрипт на языке командного интерпретатора), Вам нужно "экспортировать“ ее таким образом:

```
$ export LC_ALL
```

### 4.3.6 Путь поиска команды

Когда Вы даете команду командному интерпретатору, он ищет ее в списке каталогов, содержащихся в переменной окружения PATH. Значение этой переменной окружения называется путем поиска командного интерпретатора.

[24] Чтобы получить данный результат, Вам нужно установить французскую локаль, подробности смотрите в ‘Locales’ на стр. 168. В данный момент это не существенно для нашего руководства, просто это сделано, чтобы показать возможные эффекты.

В системе Debian по умолчанию переменная окружения PATH у пользовательской учетной записи не включает каталог /sbin. Таким образом, если Вы хотите запускать любые команды вроде ifconfig из каталога /sbin (без указания полного пути к команде - прим. переводчика), Вам нужно дополнить значение переменной PATH каталогом /sbin. Эта переменная обычно устанавливается в стартовом файле ~/.bash_profile, подробнее 'Конфигурация командного интерпретатора Bash' на стр. 36.

### 4.3.7 Опции командной строки

Некоторые команды принимают аргументы. Аргументы, начинающиеся с - или —, называются опциями и управляют поведением команды.

```
$ date
Mon Oct 27 23:02:09 CET 2003
$ date -R
Mon, 27 Oct 2003 23:02:40 +0100
```

Здесь аргумент командной строки -R изменяет поведение команды date для вывода строки даты, соответствующей документу RFC-2822.

### 4.3.8 Шаблоны имени файла (wildcards) командного интерпретатора

Часто Вам нужно, чтобы команда работала с группой файлов без набора всех их имен. Для описания группы файлов используются шаблоны имени файла командного интерпретатора:

- *
  - Соответствует любой группе из нуля и более символов.
  - Не соответствует имени файла с ".“ в начале.
- ?
  - Соответствует ровно одному символу.
- [...]
  - Соответствует ровно одному символу из тех, что заключены в скобки.
- [a-z]
  - Соответствует ровно одному символу из тех, что между a и z.
- [^...]
  - Соответствует ровно одному символу, если он не среди тех, что перечислены в скобках (символ "^“ - спецсимвол и здесь на соответствие не проверяется).

Для примера попробуйте следующие команды и поразмышляйте над полученными результатами:

```
$ mkdir junk; cd junk; $ touch 1.txt 2.txt 3.c 4.h .5.txt
$ echo *.txt
1.txt 2.txt
$ echo *
```

```
1.txt 2.txt 3.c 4.h
$ echo *.[hc]
3.c 4.h
$ echo .*
. .. .5.txt
$ echo .[^.]*
.5.txt
$ echo [^1-3]*
4.h
$ cd ..; rmdir junk
```

### 4.3.9 Код возврата команды

Каждая команда возвращает свой статус завершения как код возврата.

- код возврата равен нулю, если команда завершилась успешно.
- код возврата отличен от нуля, если команда завершилась с ошибкой.

Этот код возврата сразу после завершения команды может быть получен через переменную командного интерпретатора $?.

Пожалуйста заметьте, что когда код возврата используется в логическом контексте командного интерпретатора, успешное завершение обрабатывается как логическая ИСТИНА. Это некоторым образом нелогично, так как успешное завершение соответствует значению нуль.

Смотрите также ‘Shell conditionals’ на стр. 224.

### 4.3.10 Типовые последовательности команд

Постарайтесь запомнить следующие распространенные способы запуска команд. А после этого посмотрите части ‘Shell parameters’ на стр. 223, ‘Shell redirection’ на стр. 223, ‘Shell conditionals’ на стр. 224 и ‘Command-line processing’ на стр. 225.

команда &

команда выполняется в дочернем командном интерпретаторе в фоновом режиме. Задания в фоновом режиме позволяют пользователям запустить множество программ через один единственный командный интерпретатор.

Управление фоновыми процессаами осуществляется через встроенные команды: jobs, fg, bg и stop. Пожалуйста изучите man-страницу bash(1) в части описаний ”SIGNALS“, ”JOB CONTROL“ и ”SHELL BUILTIN COMMANDS“. [25]

[25] Система Debian является мультизадачной операционной системой.

команда1 | команда2

Стандартный поток вывода команды команда1 направляется в стандартный поток ввода команды команда2. Обе команды могут выполняться одновременно. Этот механизм называется программным каналом.

команда1 ; команда2

Команды команда1 и команда2 выполняются последовательно.

команда1 && команда2

Выполняется команда команда1. Если она завершается успешно, за ней последовательно выполняется команда команда2. Код возврата будет успешным, если обе команды команда1 и команда команда2 завершились успешно.

команда1 || команда2

Выполнятеся команда команда1. Если она завершается с ошибкой, последовательно запускается команда команда2. Код возврата будет успешным, если хотя бы одна из команд завершилась успешно.

команда > foo

Перенаправить стандартный поток вывода команды команда в файл foo (с перезаписью файла, если он существует)

команда >> foo

Перенаправить стандартный поток вывода команды команда в файл foo. (данные добавляются в конец файла)

команда > foo 2>&1

Перенаправить потоки стандартного вывода и стандартного вывода ошибок команды команда в файл foo.

команда < foo

Содержимое файла foo направляется в стандартный поток ввода команды команда. Попробуйте следующее:

```
$ </etc/motd pager
 ... (приветствие)
$ pager </etc/motd
 ... (приветствие)
$ pager /etc/motd
 ... (приветствие)
$ cat /etc/motd | pager
 ... (приветствие)
```

Хотя все 4 конструкции покажут одно и то же, последний пример запускает дополнительную команду cat и тем самым без причины расходует ресурсы.

### 4.3.11 Псевдоним команды

Вы можете установить псевдоним для часто используемой команды. Например:

```
$ alias la='ls -la'
```

Теперь la работает как сокращенная форма ls -la, которая выводит список всех файлов в полном формате (лучший вариант был предложен на кулере: alias ls='logout' - прим. переводчика ;-) )

Вы можете определить точный путь или подлинность команды, используя команду type. Например:

```
$ type ls
ls is hashed (/bin/ls)       информация о ls хэширована (/bin/ls)
$ type la
la is aliased to `ls -la'    la - псевдоним `ls -la'
$ type echo
echo is a shell builtin      echo - встроенная команда командного интерпретатора
$ type file
file is /usr/bin/file        команда file = /usr/bin/file
```

Здесь недавно выполнялся поиск команды ls, а команда file искалась в первый раз, поэтому команда ls помечена как "hashed", т.е. командный интерпретатор имеет внутреннюю запись для быстрого ответа на запрос о месторасположении файла ls.

## 4.4 Традиционная для Unix обработка текста

Существует несколько стандартных инструментов по обработке текста, которые очень часто используются на unix-подобных системах.

- Неиспользующие регулярные выражения:

  - head выводит начало файлов.
  - tail выводит конец файлов.
  - sort сортирует строки текстовых файлов.
  - uniq удаляет строки-дубликаты из сортированного файла.
  - tr выполняет замену или удаление символов.
  - diff сравнивает построчно файлы.
- Использующие базовые регулярные выражения (БРВ):
  - grep определяет соответствие текста с шаблоном.
  - ed - примитивный строковый редактор.
  - sed - потоковый редактор.
  - vi - экранный редактор.
  - emacs - экранный редактор.
- Использующие расширенные регулярные выражения (РРВ):
  - egrep определяет соответствие текста с шаблоном.
  - awk выполняем простую обработку текста. Смотрите ‘Awk’ на стр. 225.
  - perl выполняет всевозможную обработку текста. Смотрите ‘Perl’ на стр. 227.

Смотрите ‘Regular-expression substitution’ на стр. 121, ‘Script snippets for piping commands’ на стр. 123 и ‘Perl short script madness’ на стр. 125 на предмет примеров скриптов.

### 4.4.1 Регулярные выражения

Регулярные выражения используются во многих инструментах обработки текста. Они очень похожи на шаблоны имен файлов командного интерпретатора (смотрите ‘Шаблоны имени файла (wildcards) командного интерпретатора’ на стр. 60), но они не только сложнее, но и мощнее.

Регулярное выражение описывает шаблон соответствия и составляется из символов текста и метасимволов. Метасимвол - это простой символ со специальным смыслом. Существует два основных стиля регулярных выражений, БРВ и РРВ, в зависимости от текстовых инструментов, описанных в ‘Традиционная для Unix обработка текста’ на стр. 63.

Для РРВ метасимволы включают "\ . [ ] ^ $ * + ? ( ) { } |". Регулярное выражение означает:

- c
  - соответствие неметасимволу "c".
- \c
  - соответствие символу, определяемому escape-последовательностью "c", или, если "c" не escape-последовательность, то просто символу "c".
- .
  - соответствие любому символу, включая символ новой строки.
- ^
  - соответствие началу строки.
- $
  - соответствие концу строки.
- \<
  - соответствие началу слова.
- \>

  - соответствие концу слова.
- [abc...]
  - соответствие любому символу из списка "abc...".
- [^abc...]
  - отсутствие соответствия любому символу из списка "abc...".
- r*
  - соответствие регулярному выражению "r", повторенному ноль или большее количество раз.
- r+
  - соответствие регулярному выражению "r", повторенному один или большее количество раз.
- r?
  - соответствие регулярному выражению "r", повторенному ноль или один раз.
- r1|r2
  - соответствие регулярному выражению "r1" либо регулярному выражению "r2".
- (r1|r2)
  - соответствие регулярному выражению "r1" либо "r2"; конструкция обрабатывается как регулярное выражение, заключенное в скобки.

В БРВ метасимволы "+ ? ( ) { } |" теряют свое особое значение, взамен используйте те же символы с обратным слешем впереди "\+ \? \( \) \{ \} \|". Поэтому конструкция с скобками (r1|r2), записанная с использованием БРВ, должна быть в виде \(r1|r2\). Однако программа emacs, использующая БРВ, обрабатывает символы "+ ?" как метасимволы. То есть нет необходимости предворять их обратным слешем. Смотрите 'Выражения подстановки' на стр. 65, чтобы узнать, как используются конструкции со скобками.

Например, программа grep может быть использована для выполнения поиска текста с использованием регулярных выражений:

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
                    GNU GENERAL PUBLIC LICENSE
                    GNU GENERAL PUBLIC LICENSE
  Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

### 4.4.2 Выражения подстановки

В выражениях подстановки следующие символы имеют особое значение:

- &
  - означает, какое регулярное выражение совпало. (в программе emacs используйте конструкцию \&)
- \n
  - означает совпавшее регулярное выражение, заключенное в скобки, по номеру n

Для выражений подстановки языка Perl используется конструкция $n вместо \n и символ & не имеет особого значения.

Например:

```
$ echo zzz1abc2efg3hij4 | \
  sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
  sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
  perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
  perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=&=
```

Здесь обратите дополнительное внимание на стиль регулярных выражений, заключенных в скобки, и на то, как совпавшие строки используются в процессе замены текста у разных инструментов.

Эти регулярные выражения могут быть использованы для перемещения курсора и также для действий по замене текста в редакторах.

Пожалуйста, прочтите все связанные man-страницы для изучения этих команд.

## 4.5 Традиционная файловая система Unix

В системах GNU/Linux и других Unix-подобных операционных системах файлы размещаются в каталогах. [26] Все файлы и каталоги образуют одно большое дерево, иерархическую файловую структуру, корень которой /.

Эти файлы и каталоги могут располагаться на нескольких устройствах. Команда mount(8) занимается подключением файловой системы, найденной на некотором устройстве в единое дерево файлов. И наоборот, команда unmount(8) отсоединит файловую систему устройства из дерева файлов.

### 4.5.1 Основы файлов Unix

Вот несколько основых положений:

- Имена файлов являются зависимыми от регистра. То есть, файл MYFILE и файл MyFile являются различными файлами.

[26] Каталоги на некоторых системах называются папками.

- На корневой каталог (root) обычно ссылаются по имени /. Не путайте этот "root" с пользователем root. Смотрите 'Вход в систему как пользователь root' на стр. 47.
- Каждый каталог имеет имя, которое может состоят из любых букв или символов, кроме символа /. [27] Корневой каталог является исключением, его имя - / (произносится как "слеш" или "корневой каталог") и не может быть переименовано.
- На каждый файл или каталог можно сослаться по полному имени файла, абсолютному имени файла или через путь, указав последовательность каталогов, через которые нужно пройти, чтобы достичь файла. Все три термина являются синонимами. Все абсолютные имена начинаются с каталога / и существует символ / между каждым каталогом или именем файла. Первый символ / - это имя каталога, а все последующие являются просто разделителями, чтобы отделить части имени файла друг от друга. Чтобы не запутаться, посмотрите на следующий пример:

  /usr/share/keytables/us.map.gz

  Это полное имя файла, иногда некоторые называют его путем. Однако, на файл можно сослаться просто по его имени us.map.gz. [28]
- Корневой каталог имеет множество ответвлений (подкаталогов - прим. переводчика), такие как /etc или /usr. Эти подкаталоги в свою очередь содержат еще подкаталоги, такие как /etc/init.d или /usr/local соответственно. Все это вместе называется деревом каталогов. Вы можете считать, что абсолютное имя файла - это как маршрут из начала дерева (/) к концу некоей ветви этого дерева (файлу). Вы также могли слышать, что о дереве каталогов говорят как о семейном дереве: так подкаталоги имеют родителей, а путь показывает полное происхождение файла. Существуют также относительные пути, которые начинаются с места, отличного от корневого каталога. Вам нужно запомнить, что каталог ../ ссылается на родительский каталог.
- Не существует каталога, который бы соответствовал физическому устройству, например, Вашему жесткому диску. В этом отличие Unix от операционных систем CP/M, DOS и Windows, где все пути начинаются с имени устройства, например, C:\. Смотрите 'Концепция файловой системы в Debian' на стр. 68.

Хорошо описанные примеры по иерархии файлов есть в документе Стандарт на иерархию файловой системы (/usr/share/doc/debian-policy/fhs/fhs.txt.gz). Вам же для начала нужно запомнить следующие факты:

- /
  - Просто символ / обозначает корневой каталог.
- /etc
  - Здесь размещаются конфигурационные файлы системы.
- /var/log
  - Здесь размещаются системные лог-файлы (журналы событий - прим. переводчика).
- /home

[27] Хотя Вы и можете использовать почти любые буквы или символы в имени файла, на практике это плохая идея. Лучше избегать использовать символы, которые часто имеют специальное значение в командой строке, включая пробелы, символы табуляции, перевода строки и другие спецсимволы: { } ( ) [ ] ' ' " \ / > < | ; ! # & ^ * % @ $ . Если Вы хотите отделить слова в имени, лучше используйте точку, дефис или подчеркивание. Вы можете также начинать каждое слово с большой буквы: ВотТак.

[28] Существует также еще одно использование слова путь. Смотрите 'Путь поиска команды' на стр. 59. В общем, значение слова обычно становится понятным из контекста, в котором оно используется.

– Здесь размещаются домашние каталоги все непривелегированных пользователей.

### 4.5.2 Концепция файловой системы в Debian

По традициям Unix система Debian имеет файловую систему, ниже которой находятся физические данные жестких дисков и других устройств хранения информации, а также в унифицированном виде реализованы механизмы взаимодействия с оборудованием, таким как консольные экраны, удаленные последовательные консоли.

Каждый файл, каталог, именованный канал или физическое устройство в системе Debian имеет структуру данных, называющуюся индексным дескриптором (inode), который описывает связанные атрибуты, такие как владелец файла, группа, которая владеет файлом, время последнего доступа к файлу и пр. Точное описание структуры inode в системе Debian GNU/Linux смотрите в документе /usr/include/linux/fs.h.

Это унифицированное представление физических устройств является очень мощной особенностью, так как это позволяет нам использовать одну и ту же команду для выполнения однотипных действий на множестве достаточно разнотипных устройств.

Все Ваши файлы могут размещаться на одном диске, или на 20 дисках: причем некоторые из них подключены к другим компьютерам где-то в сети. И хотя Вы не можете просто так взглянуть на дерево каталогов, почти все команды работают точно также вне зависимости от того, на каком физическом устройстве реально размещаются файлы.

### 4.5.3 Права доступа к файлам и каталогам

Права доступа к файлу и каталогу определяются индивидуально для каждой из трех следующих категорий пользователей:
- для пользователя, который владеет этим файлом (u),
- для пользователей из группы, владеющей данным файлом (g) и
- для всех остальных пользователей(o).

Для файла соответствующие права доступа позволяют соответственно:
- право чтения (r): изучить содержимое файла
- право записи (w): изменять файл
- право выполнения (x): выполнять файл как команду.

Для каталога права доступа позволяют соответственно:
- право чтения (r): получать список содержимого каталога
- право записи (w): добавлять или удалять файлы в каталог
- право выполнения (x): получать доступ к файлам.

Здесь право выполнения для каталога означает не только возможность чтения файлов в каталоге, но и возможность просмотра их атрибутов, таких как размер и время модификации.

Чтобы отобразить информацию о правах доступа (и многую другую) на файлы и каталоги используется программа ls. Смотрите ls(1). Когда программа ls запускается с опцией -l, она показывает следующую информацию в указанном порядке:

- тип файла (первый символ)
  - -: обычный файл
  - d: каталог
  - l: символическая ссылка
  - c: файл символьного устройства
  - b: файл блочного устройства
  - p: именованный канал
  - s: сокет
- Права доступа к файлу (следующие девять символов, объединенных в три группы прав для владельца, группы и прочих пользователей по три символа в каждой группе в порядке право-на-чтение право-на-запись право-на-выполнение)
- количество ссылок на файл
- имя пользователя, владеющего файлом
- имя группы, владеющей файлом
- размер файла в символах (байтах)
- дата и время файла (mtime)
- имя файла.

Чтобы изменить владельца файла, используется программа chown, запущенная от имени учетной записи root. Чтобы изменить группу файла, используется программа chgrp, запущенная от имени учетной записи владельца файла или root. Для изменения прав доступа к файлу или каталогу используется программа chmod, запущенная от имени пользователя-владельца файла/каталога или от имени пользователя root. Основной синтаксис работы с файлом-примером foo следующий:

```
# chown новый_владелец foo
# chgrp новая_группа foo
# chmod  [ugoa][+-=][rwx][,...] foo
```

Подробности смотрите в соответствующих man-страницах chown(1), chgrp(1) и chmod(1).

Например, чтобы для некоторого дерева каталогов задать владельцем пользователя foo и сделать группу bar, владеющей этим деревом, выполните следующие команды от имени пользователя root:

```
# cd /некий/каталог/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Существует три бита специальных прав:
- set user ID (в выводе ls выводится как s или S в позиции права выполнения для владельца),
- set group ID (в выводе ls выводится как s или S в позиции права выполнения для группы),
- sticky bit (в выводе ls выводится как t или T в позиции права выполнения для прочих).

В выводе команде ls -l бит специального права выводится заглавной буквой (S или T), если бит выполнения, который он скрыл собой, не установлен, и наоборот.

Установка бита set user ID на выполняемый файл позволяет пользователю выполнить этот файл от имени владельца файла (например, root) Подобным образом, установка бита set group ID на выполняемый файл позволяет выполнить данный файл от имени группы, владеющей этим файлом, например, от имени группы root). Так как эти установки могут создать серъезную дыру в безопасности системы, устанавливайте эти биты очень внимательно.

Установка бита set group ID на каталог включает схему создания файлов как системах BSD, где все файлы созданные в данном каталоге будут принадлежать группе group, владеющей этим каталогом.

Установка бита sticky bit на каталог предотвращает удаление файлов в каталоге пользователями, ими не владеющими. Для защиты содержимого файла в каталоге, в который разрешена запись всем, например, /tmp, или в каталоге, в который разрешена запись некоторой группе, нужно не только снять право записи в файл, но и установить бит sticky bit на каталог. В противном случае любой пользователь, имеющий право записи в этот каталог, может удалить этот файл и создать новый с тем же именем (и с любым содержимым - прим. переводчика)

Вот несколько интересных примеров прав на файлы:

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/pppd
crw-rw----    1 root    dip       108,   0 Jan 18 13:32 /dev/ppp
-rw-r--r--    1 root    root          1051 Jan 26 08:29 /etc/passwd
-rw-r-----    1 root    shadow         746 Jan 26 08:29 /etc/shadow
-rwsr-xr--    1 root    dip         234504 Nov 24 03:58 /usr/sbin/pppd
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt    4 root    root          4096 Feb  9 16:35 /tmp
drwxrwsr-x   10 root    staff         4096 Jan 18 13:31 /usr/local
drwxrwsr-x    3 root    src           4096 Jan 19 08:36 /usr/src
drwxrwsr-x    2 root    mail          4096 Feb  2 22:19 /var/mail
drwxrwxrwt    3 root    root          4096 Jan 25 02:48 /var/tmp
```

В командах chmod(1) существует альтернативный числовой способ описания прав на файлы. Этот числовой способ использует трех-четырехзначные восьмеричные (основание системы счисления = 8) числа. Каждая цифра соответствует:

- Первая необязательная цифра: сумма атрибута set user ID (вес = 4), атрибута set group ID (вес = 2) и sticky bit (вес = 1)
- Вторая цифра: сумма атрибута чтение (вес = 4), атрибута запись (вес = 2) и атрибута выполнения (вес = 1), относящиеся к списку прав пользователя
- Третья цифра: то же самое для группы
- Четвертая цифра: то же самое для прочих пользователей

Это выглядит сложновато, но на самом деле все очень просто. Если вы посмотрите на первые несколько столбцов (2-10) вывода команды ls -l и проинтерпретируете их как двоичное (с основанием = 2) представление прав на файл ("-" означает "0", а "r", "w", "x" означает "1"), то полученное число, переведенное в восьмеричную систему счисления будет являться восьмеричным представлением прав на файл. [29] Попробуйте для примера:

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
$ ls -l foo bar
-rw-r--r--    1 penguin  penguin  0 Nov  3 23:30  foo
-rw-r--r--    1 penguin  penguin  0 Nov  3 23:30  bar
```

Маска прав на файл по умолчанию может быть установлена встроенной командой командного интерпретатора umask. Смотрите страницу builtins(7).

### 4.5.4 ВременнЫе метки (Timestamps)

У файла в системе GNU/Linux существует три типа временных метки:

- mtime: время модификации (ls -l),
- ctime: время изменения состояния (ls -lc) и
- atime: время последнего доступа (ls -lu).

Заметьте, что ctime не является временем создания файла.

- Перезапись файла изменит каждую из временных меток mtime, ctime и atime файла.
- Изменение прав доступа или владельца файла изменит временные метки ctime и atime.
- Чтение файла изменит временную метку atime.

Обратите внимание, что даже простое чтение файла в системе Debian обычно приводит к операции записи с файлом для обновления временной метки atime в структуре inode. Монтирование файловой системы с опцией noatime позволит системе не делать эту операцию и приведет к повышению скорости чтения. Смотрите страницу mount(8).

Используйте команду touch(1)для изменения временных меток существующих файлов.

### 4.5.5 Ссылки

Существует два способа связать файл foo с другим именем файла bar.

- жесткая ссылка - это имя-дубликат для существующего файла (ln foo bar),
- символичесая ссылка или "symlink" - это специальный файл, который указывает на другой файл по его имени (ln -s foo bar).

Смотрите следующий пример на предмет изменения счетчика ссылок и тонкую разницу в результате команды rm

```
$ echo "Исходное содержимое" > foo
$ ls -l foo
```

[29] Конечно, это метод работает только для права, представленного тремя цифрами.

```
-rw-r--r--   1 osamu   osamu        4 Feb  9 22:26 foo
$ ln foo bar    # жесткая ссылка
$ ln -s foo baz  # символическая ссылка
$ ls -l foo bar baz
-rw-r--r--   2 osamu   osamu        4 Feb  9 22:26 bar
lrwxrwxrwx   1 osamu   osamu        3 Feb  9 22:28 baz -> foo
-rw-r--r--   2 osamu   osamu        4 Feb  9 22:26 foo
$ rm foo
$ echo "Новое содержимое" > foo
$ cat bar
Исходное содержимое
$ cat baz
Новое содержимое
```

Символическая ссылка всегда имеет номинальные права доступа файла “rwxrwxrwx”, как это было показано в предыдущем примере, и эффективные права доступа соответствующие правам доступа файлу, на который ссылка указывает.

Каталог . является ссылкой на сам каталог, в котором он находится, то есть счетчик ссылок любого нового каталога начинается с 2. Каталог .. ссылается на родительский каталог, поэтому счетчик ссылок каталога увеличивается с добавлением новых подкаталогов.

### 4.5.6 Именованные каналы (FIFOs)

Именованный канал - это файл, которые действует как труба. Вы помещаете что-то в данный файл, и оно выходит с другого конца. Это называется FIFO или First-In-First-Out: то, что первым помещено в канал, первым из него и выходит с другой стороны.

Если Вы пишите в именованный канал, то процесс, который это делает, не завершается до тех пор, пока записанная информация не будет прочтена из канала. Если Вы читаете из именнованного канала, то процесс, который это делает, будет работать до тех пор, пока в канале есть информация для чтения. Размер файла канала всегда нулевой - он не хранит данные, он просто связывает два процесса, как это делает операция | командного интерпретатора. Однако, так как этот канал имеет имя, то нет необходимости запускать эти процессы (читающий и пишущий - прим. переводчика) в одной и тоже командной строке (в отличие от | - прим. переводчика) и от имени одного и того же пользователя.

Вы можете изучить, как работает канал, на следующем примере:

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # запустим в фоновом режиме
[1] 5952
$ ls -l mypipe
prw-r--r--   1 penguin penguin  0 2003-11-06 23:18 mypipe
$ cat mypipe
hello
```

```
[1]+  Done                    echo hello >mypipe
$ ls mypipe
prw-r--r--    1 penguin penguin  0 2003-11-06 23:20 mypipe
$ rm mypipe
```

### 4.5.7 Сокеты

Сокеты очень похожи на именованные каналы (FIFO) и позволяют процессам обмениваться информацией. В случае с сокетом эти процессы не обязаны выполняться на одной и тоже машине или быть потомками одного и тоже родительского процесса. Это конечная точка в в межпроцессном взаимодействии. Обмен информации может происходить через сеть между различными компьютерами.

### 4.5.8 Файлы устройств

Файлы устройств ссылаются на физические или виртуальные устройства в Вашей системе, такие как жесткий диск, видеокарта, экран или клавиатура. Пример виртуальнго устройства - консоль, представленная файлом устройства /dev/console.

Существует два типа устройств:

- символьные устройства
  - Доступ к ним по одному символу за раз, то есть самый маленький элемент данных, который может быть записан в устройство или прочитан с устройства - символ (байт).
- блочные устройства
  - Доступ к ним осуществляется посредством бОльших элементов, называемых блоками, которые могут содержать множество символов. Ваш жесткий диск является блочным устройством.

Вы можете читать или записывать в файл устройства, хотя сам файл содержит двоичные данные, не очень понятные человеку. Запись данных непосредственно в такие файлы иногда полезны для решения проблем подключений оборудования. Например, Вы можете вывести текстовый файл в принтерный файл устройства /dev/lp0 или послать команды модему в соответствующий файл устройства последовательного порта /dev/ttyS0. Но делайте это аккуратно, так как это может привести к серъезным последствиям. Будьте осторожны.

#### /dev/null и прочее.

Файл /dev/null - это специальный файл устройства, который обрасывает все то, что Вы в него записываете. Если Вам что-то не нужно, направьте это в /dev/null. Это в сущности бездонная яма. Если же Вы читаете файл /dev/null, то сразу получите символ конца файла (EOF).

Файл /dev/zero очень похож, но только если Вы читате его, то получите символ \0 (это не тоже самое, что ASCII-код цифры ноль). Смотрите 'Dummy files' на стр. 130.

Номер устройства

Номер устройства отображается при выполнении команды ls:

```
$ ls -l /dev/hda /dev/ttyS0 /dev/zero
brw-rw----    1 root     disk       3,   0 Mar 14  2002 /dev/hda
crw-rw----    1 root     dialout    4,  64 Nov 15 09:51 /dev/ttyS0
crw-rw-rw-    1 root     root       1,   5 Aug 31 03:03 /dev/zero
```

Здесь

- Файл /dev/hda имеет старший номер устройства 3 и младший номер устройства 0. Файл доступен для чтения/записи пользователем из группы disk.
- Файл /dev/ttyS0 имеет старший номер устройства 4 и младший номер устройства 64. И он доступен для чтения/записи пользователем из группы dialout и
- Файл /dev/zero имеет страший номер устройства 1 и младший номер устройства 5. Доступен для чтения/записи всеми пользователями.

В системах постарше процесс установки создавал файлы устройств при помощи команды /sbin/MAKEDEV. Смотрите MAKEDEV(8).

В более новых системах файловая система в /dev автоматически наполняется при помощи файловой системы устройств, как это делается, например, в /proc.

### 4.5.9 Файловая система /proc

Файловая система /proc - это псевдо-файловая система, содержащая информацию о системе и работающих процессах.

Часто пугаются того, что один файл в этой файловой системе /proc/kcore очень большого размера. Он содержит копию (более или менее) оперативной памяти Вашего компьютера и используется для отладки ядра. Поскольку фактически он нигде не существует, то и не беспокойтесь о его размере.

Смотрите раздел 'Tuning the kernel through the proc filesystem' на стр. 107 и страницу proc(5).

## 4.6 Система X Window System

Смотрите 'X' на стр. 145.

### 4.6.1 Запуск системы the X Window System

Система X Window System может быть запущена автоматически при помощи специального графического демона входа навроде xdm или через ввод команды в консоли:

```
$ exec startx
```

### 4.6.2 Меню в системе X Window System

Так как среда X может работать со многими оконными менеджерами, их пользовательские интерфейсы могут отличаться. Пожалуйста запомните, что нажатие правой кнопки мыши на главном (корневом) окне приводит к появлению меню. Почти всегда.

- Чтобы получить командную строку, запустите Xterm из меню:
  - "XShells" –> "XTerm".
- Для графического просмотра web-страниц запустите Mozilla из меню:
  - "Apps" –> "Net" –> "Mozilla Navigator".
- Для графического просмотра PDF-файлов запустите Xpdf из меню:
  - "Apps" –> "Viewers" –> "Xpdf".

Если Вы не нашли нужный элемент в меню, установите требуемые пакеты. Смотрите раздел 'Основы управления пакетами в Debian' на стр. 82.

### 4.6.3 Клавиатурные комбинации клавиш в системе X Window System

Запомните следующие важные при работе в системе X Window System комбинации клавиш:

- Ctrl-Alt-F1 по F6: Переключиться в соответствующий превдотерминал (из системы X Window, DOSEMU и т.д.)
- Alt-F7: Вернуться назад в X Window
- Ctrl-Alt-минус: Изменить разрешение экрана в системе X Window ("серый" минус)
- Ctrl-Alt-плюс: Изменить разрешение экрана в системе X Window ("серый" плюс)
- Ctrl-Alt-Backspace: Завершить программу X Server
- Alt-X, Alt-C, Alt-V: Привычные в Windows/Mac клавиатурные комбинации Вырезать, Копировать, Вставить, нажимаемые с клавишей Ctrl, заменяются на комбинации с клавишей Alt для использования в некоторых программах, например, Netscape Composer.

## 4.7 Для дальнейшего изучения

На этот момент я рекомендую Вам прочесть ключевые руководства из раздела Проекта документации Linux: Руководства (http://www.tldp.org/guides.html):

- "Руководство системного администратора Linux",
  - Охватывает все аспекты поддержания работающей системы, работу с учетными записями пользователей, резервное копирование, настройку системы.
  - пакет: sysadmin-guide
  - файл: /usr/share/doc/sysadmin-guide/html/index.html
  - web: http://www.tldp.org/LDP/sag/index.html
- "Руководство администратора сети в Linux, вторая редакция",
  - Это единственный справочник по администрированию сети в Linux-среде.
  - пакет: (not available)
  - файл: (notapplicable)
  - web: http://www.tldp.org/LDP/nag2/index.html

- “Книга рецептов по Linux”,
  - Содержит более 1,500 полезных рецептов и советов для очень занятого пользователя компьютера.
  - пакет: linuxcookbook (only in Woody)
  - файл: /usr/share/doc/linuxcookbook/cookbook_toc.html
  - web: http://www.tldp.org/LDP/linuxcookbook/html/index.html

Дополнительную информацию смотрите в разделе ‘Support for Debian’ на стр. 245.

# Глава 5

# Обновление до дистрибутива testing

Официальный документ по обновлению размещается по адресам http://www.debian.org/releases/stable/releasenotes и http://www.debian.org/releases/testing/releasenotes (в разработке).

Процесс обновления системы до дистрибутива testing/unstable следующий:

- если Ваша система - Potato, то нужно обновить систему APT до woody-версии, так как версия системы APT в Potato не имеет всех возможностей, описанных на man-странице apt_preferences(5).
- измените Ваши файлы /etc/apt/sources.list и /etc/apt/preferences таким образом, чтобы они содержали ссылки на секцию "testing" репозитария, который Вы используете. Если Вы хотите, можно также добавить ссылки и на секцию "unstable".
- обновите Ваши списки пакетов и установите все пакеты, которые можно обновить.

## 5.1 Перевод системы APT к ее Woody-версии

Если Вы все еще работаете в Potato, то возможно следующее обновление по сети системы APT 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
```

## 5.2 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. Закоментируйте 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. 82.

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

## 5.3 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. 81 и к разделу 'Разрешение проблем APT upgrade' на стр. 89, если возникли трудности.

### 5.3.1 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. 84.

Используйте программу dselect. Она всегда работает :)

### 5.3.2 Устаревший способ обновления через apt-get

Использование apt-get, описанное ниже, широко распространено, но не рекомендуется для обновления системы. Если Вам нужно обновиться без использования dselect в системах новее Woody, рассмотрите программу aptitude и прочие.

Если система не имеет много пакетов или архив Debian не сильно изменился, следующего может быть достаточно (иногда):

```
# apt-get update # всегда делается перед обновлением
 ... для обновления системы на основе зависимостей "depends":
# apt-get upgrade                    # затем всегда делаем это
 ... для обновления в целом системы на основе информации о "depends"-зависимостях:
# apt-get -u dist-upgrade
... или обновиться и придерживаться текущего выбора в dselect (новый и лучший способ):
# apt-get -u dselect-upgrade         # используем результат выбора в dselect
```

Так как этот способ обновления использует программу apt-get, то обработка зависимостей recommends и suggests ограничена. Смотрите ‘Зависимости пакетов’ на стр. 16.

# Глава 6

# Управление пакетами в Debian

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

Чтобы снизить нагрузку на репозитарии Debian по сети, установите локальный HTTP прокси-сервер squid для кэширования пакетов, скачиваемых системой APT, и настройте переменную окружения http_proxy или задайте значение параметру http в файле /etc/apt /apt.conf (для того, чтобы APT 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 метода, описанного в 'chroot' на стр. 130 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).

## 6.1 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 :-).

### 6.1.1 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 репозитариями пакетов, интерфейс командной строки к системе APT)
tasksel – инсталлятор задач (набора пакетов)
aptitude – инсталлятор пакетов (пакеты & задачи, интерфейс к APT на базе ncurses)
deity – альтернатива ncurses-интерфейса к APT
synaptic, gsynaptic – графические утилиты для работы с APT

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

APT 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 имееют 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. Смотрите ‘Зависимости пакетов’ на стр. 16.

### 6.1.2 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
...

## 6.2 Основы управления пакетами в Debian

Вы можете установить набор пакетов, называемых задачей, установить пакеты по одному или обновить систему, используя инструменты управления пакетами, описанные ниже . Также обращайтесь к ‘Советы по установке системы Debian’ на стр. 27, ‘Обновление до дистрибутива testing’ на стр. 77 и ‘Rescue editors’ на стр. 207.

### 6.2.1 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.

### 6.2.2 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. 77 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
```

[1] I 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.

### 6.2.3 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 лучшим был их пропуск при начальной установке системы путем ручного снятия отметки соответствущих 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.

### 6.2.4 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 клавишами управления |
| u | Обновить информацию об доступных пакетах в репозитарии |
| 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.

### 6.2.5 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.

### 6.2.6 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 APT - нет перевода!!!!' на стр. 83 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.

### 6.2.7 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.

### 6.2.8 Обзор файла /etc/apt/preferences

В примере файла /etc/apt/preferences, представленного в разделе 'Установка системы APT - нет перевода!!!!' на стр. 83, 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 Packages.gz из архива Debian, у которого в файле Release есть строка "Archive: stable".

В последней строчке запись "Pin-Priority: 500" указвает 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 диапозона 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 /apt.conf следующим образом:

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

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

## 6.3 Команды выживания в системе Debian

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

### 6.3.1 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
```

Большое количество ресурсов по поддержке приведены по адресу 'Support for Debian' на стр. 245.

### 6.3.2 Разрешение проблем APT upgrade

Проблемы с зависимостями пакетов могут случаться при обновлениях в unstable/testing, описанных в 'Обновление системы Debian' на стр. 79. 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 возникновения ошибки
... или
# apt-get dist-upgrade -f # продолжение обновления дистрибутива (dist-upgrade) даже после 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 and 'The equivs package' на стр. 99.

### 6.3.3 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. (Смотрите ‘Каталог pool’ на стр. 9.)

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
```

### 6.3.4 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 идей 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
```

### 6.3.5 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. 90. ([FIXME]: This procedure needs more experiments to verify.)

### 6.3.6 Установка пакета в систему, которая не загружается

Загрузите Linux со спасательного диска/дискеты Debian или альтернативного раздела мультизагрузочной Linux-системы. See ‘Booting the system’ на стр. 109. 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).

### 6.3.7 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.

## 6.4 Нирвана в командах Debian

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

### 6.4.1 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.
```

### 6.4.2 Информация о пакете

Поиск и вывод информации из архива пакетов. Убедитесь, что система APT 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 mc 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*
```

### 6.4.3 Автоматическая установка при помощи APT

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

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

Это эквивалентно запуску команды apt-get -q -y 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 повтороно конфигурировать любой конкретный пакет путем ‘Повторное конфигурирование установленных пакетов’ на стр. 94.

### 6.4.4 Повторное конфигурирование установленных пакетов

Используйте следующие команды для повторного конфигурирования любого уже установленного пакета.

```
# dpkg-reconfigure --priority=medium пакет [...]
# dpkg-reconfigure --all   # повторно конфигурирвоать 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 настроечными 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
...
```

### 6.4.5 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 ...
```

### 6.4.6 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$"
```

### 6.4.7 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
```

### 6.4.8 Удаление файлов кэшированных пакетов

В процессе установки через APT 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
```

### 6.4.9 Запись/копирование конфигурации системы

Для создания локальной копии списка состояния выбранных пакетов сделайте:

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

Использование "*" приводит к помещению в файл мои_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 <мои_выбранные_пакеты
# apt-get -u dselect-upgrade    # или dselect install
```

### 6.4.10 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 суффиком "-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, дл компиляции пакета pine, который поставляется только в виде исходного текста, выполните:

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

### 6.4.11 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 -1 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.

### 6.4.12 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.

### 6.4.13 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).

### 6.4.14 Optimized sources.list

In short, fancy efforts to create an optimized sources.list did not produce a significant improvement for me from a location in the USA. I manually chose a nearby site using apt-setup.

apt-spy creates sources.list automatically, based on latency and bandwidth. netselect-apt creates a more complete sources.list, but uses an inferior method of choosing the best mirror (ping time comparison).

```
# apt-get install apt-spy
# cd /etc/apt ; mv sources.list sources.list.org
# apt-spy -d testing -l sources.apt
```

## 6.5 Other Debian peculiarities

### 6.5.1 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 (see 'Сохранение текущей конфигурации' на стр. 13).

```
# 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.

### 6.5.2 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.
```

### 6.5.3 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. For details, see ‘Custom X sessions’ на стр. 150.

/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.

### 6.5.4 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.

### 6.5.5 Отключение неиспользующихся сервисов

Дистрибутив Debian серъезно 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 подробнее в ‘Restricting logins with PAM’ на стр. 137. Очень полезная команда pidof (смотрите pidof(8)).

С недавних пор в Debian система X11 не позволяет удаленные TCP/IP подключения. Смотрите ‘Using X over TCP/IP’ на стр. 153. Проброс трафика системы X Window в тунеле SSH также запрещена. Смотрите ‘Connecting to a remote X server – ssh’ на стр. 153.

# Глава 7

# The Linux kernel under Debian

Debian has its own method of recompiling the kernel and related modules. See also 'Debian и ядро' на стр. 24.

## 7.1 Kernel recompile

The use of gcc, binutils, and modutils from Debian unstable may help when compiling the latest Linux kernel. See /usr/share/doc/kernel-package/README.gz, especially the bottom of this, for the official information.

Since it is a moving target, kernel compilation is a difficult subject that may confuse even the most admired developer:

Manoj Srivastava wrote: —initrd requires a Debian-only cramfs patch.

Herbert Xu wrote: No it does not, all you have to do to use a filesystem other than CRAMFS is to set MKIMAGE in /etc/mkinitrd/mkinitrd.conf.

Be careful and always rely on the /usr/share/doc/kernel-package/README.gz by Manoj and Kent. Make sure to obtain the latest unstable version of the kernel-package package if you are to compile the latest version of the kernel.

initrd is not needed for a kernel compiled only for one machine. I use it since I want my kernel to be almost the same as the one provided by the kernel-image packages. If you use initrd, make sure to read mkinitrd(8) and mkinitrd.conf(5). See also http://bugs.debian.org/149236.

### 7.1.1 Debian standard method

Watch out for bug reports on kernel-package, gcc, binutils and modutils. Use new versions of them as needed.

Compiling a custom kernel from source under a Debian system requires special care. Use the new —append_to_version with make-kpkg to build multiple kernel-images.

```
# apt-get install debhelper modutils kernel-package libncurses5-dev
# apt-get install kernel-source-2.4.18    # use latest version
# apt-get install fakeroot
# vi /etc/kernel-pkg.conf                 # input my name and email
$ cd /usr/src                             # build directory
$ tar --bzip2 -xvf kernel-source-2.4.18.tar.bz2
$ cd kernel-source-2.4.18                 # if this is your kernel source
$ cp /boot/config-2.4.18-386 .config      # get current config as default
$ make menuconfig                         # customize as one wishes
$ make-kpkg clean                         # must run (per: man make-kpkg)
$ fakeroot make-kpkg --append_to_version -486 --initrd \
        --revision=rev.01 kernel_image \
        modules_image # modules_image is for pcmcia-cs* etc.
$ cd ..
# dpkg -i kernel-image*.deb pcmcia-cs*.deb # install
```

make-kpkg kernel_image actually does make oldconfig and make dep. Do not use –initrd if initrd is not used.

If one wants to use modules from pcmcia-cs or no pcmcia, one should select "General setup –>" to "PCMCIA/CardBus support –>" in make menuconfig and set the configuration to "< > PCMCIA/CardBus support" (i.e., uncheck the box).

On an SMP machine, set CONCURRENCY_LEVEL according to kernel-pkg.conf(5).

### 7.1.2 Classic method

Get pristine sources from:

- Linux: http://www.kernel.org/
- pcmcia-cs: http://pcmcia-cs.sourceforge.net/

or use equivalent sources in Debian and do the following:

```
# cd /usr/src
# tar xfvz linux-whatever.tar.gz
# rm -rf linux
# ln -s linux-whatever linux
# tar xfvz pcmcia-cs-whatever.tar.gz
# ln -s pcmcia-cs-whatever pcmcia
# cd linux
# make menuconfig
... configure stuff ...
# make dep
# make bzImage
... edits for lilo / grub ...
... move /usr/src/linux/arch/i386/boot/bzImage to boot ...
... /sbin/lilo or whatever you do for grub
```

```
# make modules; make modules_install
# cd ../pcmcia
# make config
# make all
# make install
... add needed module names to /etc/modules
# shutdown -r now
... boot to new kernel ...
```

### 7.1.3 Kernel headers

Most "normal" programs don't need kernel headers and in fact may break if you use them directly; instead they should be compiled against the headers with which glibc was built, which are the versions in /usr/include/linux and /usr/include/asm of the Debian system.

So do not put symlinks to the directories in /usr/src/linux from /usr/include/linux and /usr /include/asm, as suggested by some outdated documents.

If you need particular kernel headers for some kernel-specific application programs, alter the makefile(s) so that their include path points to dir-of-particular-kernel-headers/include/linux and dir-of-particular-kernel-headers/include/asm.

## 7.2 The modularized 2.4 kernel

The new Debian 2.4 kernels provided by kernel-image-2.4.NN are very modularized. You have to make sure those modules are activated to make the kernel function as you intend.

Although I have many examples for /etc/modules in the following section as a quick fix, I hear that the correct way to fix these module-related issues is to provide an alias for the device in a file in /etc/modutils/ since there are enough aliases available with current kernels. Some modules may be auto activated by hardware detection programs such as discover. See also 'Hardware detection for X' на стр. 146.

See Documentation/*.txt in the Linux source for the precise information.

### 7.2.1 PCMCIA

/etc/modules may need to contain the following for some old PCMCIA to function:

```
# ISA PnP driver
isa-pnp
# New Low level PCMCIA driver
# yenta_socket # does not seem to be needed in my case
```

The rest is taken care of by PCMCIA scripts (from the pcmcia-cs package), depmod and kmod. I think I needed isa-pnp because my laptop is an old ISA-PCMCIA. Recent laptops with CardBus/PCMCIA may not require this.

Voice of the genius Miquel van Smoorenburg <miquels@cistron.nl>:

"I simply removed the entire pcmcia stuff from the laptop here at work, including the cardmgr etc and just installed a 2.4 kernel with cardbus support, and the new hotplug package from woody.

As long as you only have 32-bit cards you don't need the pcmcia package; 2.4 has card services built in. And the standard tulip driver should work fine with your dlink card.

—Mike.“

See Linux PCMCIA HOWTO (http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html) and 'Network configuration and PCMCIA' на стр. 198.

### 7.2.2 SCSI

[NOT TESTED] /etc/modules needs to contain the following for SCSI to function:

```
# SCSI core
scsi_mod
# SCSI generic driver
sg
# SCSI disk
sd_mod
# All other needed HW modules
...
```

depmod may take care of some of the above modules.

### 7.2.3 Network function

/etc/modules needs to contain the following for extra network function:

```
# net/ipv-4
ip_gre
ipip

# net/ipv-4/netfilter
# iptable (in order)
ip_tables
ip_conntrack
ip_conntrack_ftp
```

```
iptable_nat
iptable_filter
iptable_mangle
#
ip_nat_ftp
ip_queue
#
ipt_LOG
ipt_MARK
ipt_MASQUERADE
ipt_MIRROR
ipt_REDIRECT
ipt_REJECT
ipt_TCPMSS
ipt_TOS
ipt_limit
ipt_mac
ipt_mark
ipt_multiport
ipt_owner
ipt_state
ipt_tcpmss
ipt_tos
ipt_unclean
#
#ipchains
#ipfwadm
```

The preceding may not be optimized. depmod may take care of some of the above modules.

### 7.2.4 EXT3 filesystem ( > 2.4.17)

Enabling a journaling filesystem with the EXT3 FS involves the following steps using a Debian precompiled kernel-image ( > 2.4.17) package:

```
# cd /etc; mv fstab fstab.old
# sed 's/ext2/ext3,ext2/g' <fstab.old >fstab
# vi /etc/fstab
... set root filesystem type to "auto" instead of "ext3,ext2"
# cd /etc/mkinitrd
# echo jbd >>modules
# echo ext3 >>modules
# echo ext2 >>modules
# cd /
# apt-get update; apt-get install kernel-image-2.4.17-686-smp
```

```
... install latest kernel and set up boot (lilo is run here)
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... For all EXT2 FS's converted to EXT3
# shutdown -r now
```

Now EXT3 journaling is enabled. Using ext3,ext2 as the fstab "type" entry ensures safe fallback to EXT2 if the kernel does not support EXT3 for non-root partitions.

If you have previously installed a 2.4 kernel and do not wish to reinstall, perform the above steps up to the apt-get commands, then:

```
# mkinitrd -o /boot/initrd.img-2.4.17-686-smp /lib/modules/2.4.17-686-smp
# lilo
# tune2fs -j -i 0 /dev/hda1
# tune2fs -j -i 0 /dev/hda2
... for all EXT2 FS's converted to EXT3
# shutdown -r now
```

Now EXT3 journaling is enabled.

If /etc/mkinitrd/modules was not set when mkinitrd was run and you would like to add some modules at boot time:

```
... at initrd prompt to gain shell (5 sec.), type RETURN
# insmod jbd
# insmod ext3 # modprobe ext3 may take care of everything
# insmod ext2
# ^D
... continue booting
```

At the system boot screen (dmesg), "cramfs: wrong magic" may appear but this is known to be harmless. This issue has been resolved in Sarge (2002/10). See http://bugs.debian.org/135537 and the EXT3 File System mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html) or /usr/share/doc/HOWTO/en-txt/mini/extra/ext3-mini-HOWTO.gz for more information.

Some systems are reported to experience severe kernel lockup if EXT3 is enabled but I had no problem (as of 2.4.17).

### 7.2.5 Realtek RTL-8139 support in 2.4

For whatever reason, the RTL-8139 support module is no longer called rtl8139, it's now called 8139too. Just edit your /etc/modules to reflect this change when upgrading a 2.2 kernel to a 2.4 kernel.

### 7.2.6 Parallel port support

For kernel-image-2.4.*, parallel port support is provided as a module. Enable it by:

```
# modprobe lp
# echo lp >> /etc/modules
```

See Documentation/parport.txt in the Linux source.

## 7.3 Tuning the kernel through the proc filesystem

The behavior of the Linux kernel can be changed on the fly using the proc filesystem.

For basic information on changing kernel parameters through the /proc filesystem, read Documentation/sysctl/* in the Linux source.

See some examples of kernel parameter manipulations in /etc/init.d/networking and 'Странные проблемы доступа на некоторые web-сайты' на стр. 45.

See sysctl.conf(5) for how to set up the boot time kernel configuration through /proc filesystem with /etc/init.d/procps.sh script usually run from /etc/rcS.d/S30procps.sh.

### 7.3.1 Too many open files

The Linux kernel may complain "Too many open files". This is due to the small default value (8096) for file-max. To fix this problem, run the following command as root:

```
# echo "65536"  > /proc/sys/fs/file-max  # for 2.2 and 2.4 kernel
# echo "131072" > /proc/sys/fs/inode-max # for 2.2 kernel only
```

or put the following into /etc/sysctl.conf for the permanent change:

```
file-max=65536    # for 2.2 and 2.4 kernel
inode-max=131072 # for 2.2 kernel only
```

### 7.3.2 Disk flush intervals

You can change disk flush intervals through the proc filesystem. The following will shorten its interval from the default five seconds to one second.

```
# echo "40 0 0 0 100 30000 60 0 0"  > /proc/sys/vm/bdflush
```

This may negatively impact file I/O performance a little bit. But this secures file contents except for the last one second which is shorter than the default five seconds. This is true even for journaling filesystems.

### 7.3.3 Sluggish old low memory machines

For some old low memory systems, it may still be useful to enable over-commit of memory through the proc filesystem:

```
# echo 1 > /proc/sys/vm/overcommit_memory
```

# Глава 8

# Debian tips

## 8.1 Booting the system

See the LDP BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html) for detailed information on the boot prompt.

### 8.1.1 "I forgot the root password!" (1)

It is possible to boot a system and log on to the root account without knowing the root password as long as one has access to the console keyboard. (This assumes there are no password requests from the BIOS or from a boot loader such as lilo that would prevent one from booting the system.)

This is a procedure which requires no external boot disks and no change in BIOS boot settings. Here, "Linux" is the label for booting the Linux kernel in the default Debian install.

At the lilo boot screen, as soon as boot: appears (you must press a shift key at this point on some systems to prevent automatic booting), enter:

```
boot: Linux init=/bin/sh
```

This causes the system to boot the kernel and run /bin/sh instead of its standard init. Now you have gained root privileges and a root shell. Since / is currently mounted read-only and many disk partitions have not been mounted yet, you must do the following to have a reasonably functioning system.

```
init-2.03# mount -n -o remount,rw /
init-2.03# mount -avt nonfs,noproc,nosmbfs
init-2.03# cd /etc
init-2.03# vi passwd
init-2.03# vi shadow
```

(If the second data field in /etc/passwd is "x" for every username, your system uses shadow passwords, and you must edit /etc/shadow.) To disable the root password, edit the second data field in the password file so that it is empty. Now the system can be rebooted and you can log on as root without a password. When booting into runlevel 1, Debian (at least after Potato) requires a password, which some older distributions did not.

It is a good idea to have a minimal editor in /bin/ in case /usr/ is not accessible (see 'Rescue editors' на стр. 207).

Also consider installing the sash package. When the system becomes unbootable, execute:

```
boot: Linux init=/bin/sash
```

sash serves as an interactive substitute for sh even when /bin/sh is unusable. It's statically linked, and includes many standard utilities as built-ins (type "help" at the prompt for a reference list).

### 8.1.2 "I forgot the root password!" (2)

Boot from any emergency boot/root disk set. If /dev/hda3 is the original root partition, the following will let one edit the password file just as easily as the above.

```
# mkdir fixit
# mount /dev/hda3 fixit
# cd fixit/etc
# vi shadow
# vi passwd
```

The advantage of this approach over the previous method is one does not need to know the lilo password (if any). But to use it one must be able to access the BIOS setup to allow the system to boot from floppy disk or CD, if that is not already set.

### 8.1.3 Cannot boot the system

No problem, even if you didn't bother to make a boot disk during install. If lilo is broken, grab the boot disk from the Debian installation set and boot your system from it. At the boot prompt, assuming the root partition of your Linux installation is on /dev/hda12 and you want runlevel 3, enter:

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

Then you are booted into an almost fully functional system using the kernel on the floppy. (There may be minor glitches due to lack of kernel features or modules.)

See also 'Установка пакета в систему, которая не загружается' на стр. 91 if you have a broken system.

If you need a custom boot floppy, follow readme.txt on the rescue disk.

### 8.1.4 "Let me disable X on boot!"

Chasing unstable/sid is fun, but buggy xdm, gdm, kdm, and wdm started during the boot process can bite you bad.

First get the root shell by entering the following at the boot prompt:

```
boot: Linux vga=normal s
```

Here, Linux is the label for the kernel image you are booting; "vga=normal" will make sure lilo runs in normal VGA screen, and "s" (or "S") is the parameter passed to init to invoke single-user mode. Enter the root password at the prompt.

There are few ways to disable all the X starting daemons:

- run update-rc.d ?dm stop 99 1 2 3 4 5 6 .
- insert "exit 0" at the start of all /etc/init.d/?dm files.
- rename all /etc/rc2.d/S99?dm files to /etc/rc2.d/K99?dm.
- remove all /etc/rc2.d/S99?dm files.
- run :>/etc/X11/default-display-manager

Here, number in rc2.d must correspond to the runlevel specified in the /etc/inittab. Also ?dm means that you need to run the command multiple times by substituting it with all of the xdm, gdm, kdm, and wdm.

Only the first one in the list is "the one true way" in Debian. The last one is easy but only works on Debian and requires you to set the display manager again later using dpkg-reconfigure. Others are generic methods to disable daemons.

You can still start X with the startx command from any console shell.

### 8.1.5 Other boot tricks with the boot prompt

The system can be booted into a particular runlevel and configuration using the lilo boot prompt. Details are given in the BootPrompt-HOWTO (http://www.tldp.org/HOWTO/BootPrompt-HOWTO.html) (LDP).

If you want to boot the system into runlevel 4, use the following input at the lilo boot prompt.

```
boot: Linux 4
```

If you want to boot the system into normally functioning single-user mode and you know the root password, one of the following examples at the lilo boot prompt will work.

```
boot: Linux S
boot: Linux 1
boot: Linux -s
```

If you want to boot the system with less memory than system actually has (say 48MB for a system with 64MB), use this input at the lilo boot prompt:

```
boot: Linux mem=48M
```

Make sure not to specify more than the actual memory size here, otherwise the kernel will crash. If one has more than 64MB of memory, e.g. 128MB, unless one executes mem=128M at the boot prompt or includes a similar append line in /etc/lilo.conf, old kernels and/or a motherboard with an old BIOS will not use memory beyond 64MB.

### 8.1.6 Setting GRUB boot parameters

GRUB is a new boot manager from the GNU Hurd project and is much more flexible than Lilo but has slightly different handling of boot parameters.

```
grub> find /vmlinuz
grub> root (hd0,0)
grub> kernel /vmlinuz root=/dev/hda1
grub> initrd /initrd
grub> boot
```

Here, you must be aware of the Hurd device names:

| the Hurd/GRUB | Linux | MSDOS/Windows |
|---|---|---|
| (fd0) | /dev/fd0 | A: |
| (hd0,0) | /dev/hda1 | C: (usually) |
| (hd0,3) | /dev/hda4 | F: (usually) |
| (hd1,3) | /dev/hdb4 | ? |

See /usr/share/doc/grub/README.Debian.gz and /usr/share/doc/grub-doc/html/ for details.

## 8.2 Recording activities

### 8.2.1 Recording shell activities

System administration involves much more elaborate tasks in a Unix environment than in an ordinary personal computer environment. Make sure to know the most basic means of configuration in case you need to recover from system trouble. X11-based GUI configuration tools look nice and convenient but are often unsuitable in these emergency situations.

In this context, recording shell activities is a good practice, especially as root.

Emacs: Use M-x shell to start recording into a buffer, and use C-x C-w to write the buffer to a file.

Shell: Use the screen command with "^A H" as described in 'Console switching with screen' на стр. 127; or use the script command.

```
$ script
Script started, file is typescript
 ... do whatever ...
 Ctrl-D
$ col -bx <typescript >savefile
$ vi savefile
```

The following can be used instead of script:

```
$ bash -i 2>&1 | tee typescript
```

### 8.2.2 Recording X activities

If you need to record the graphic image of an X application, including an xterm display, use gimp (GUI). It can capture each window or the whole screen. Alternatives are xwd (xbase-clients), import (imagemagick), and scrot (scrot).

## 8.3 Copy and archive a whole subdirectory

### 8.3.1 Basic commands for copying a whole subdirectory

If you need to rearrange file structure, move content including file links by:

```
Standard method:
# cp -a /source/directory /dest/directory # requires GNU cp
# (cd /source/directory && tar cf - . ) | \
       (cd /dest/directory && tar xvfp - )
If a hard link is involved, a pedantic method is needed:
# cd /path/to/old/directory
# find . -depth -print0 | afio -p -xv -0a /mount/point/of/new/directory
If remote:
# (cd /source/directory && tar cf - . ) | \
       ssh user@host.dom (cd /dest/directory && tar xvfp - )
```

```
If there are no linked files:
# scp -pr user1@host1.dom:/source/directory \
        user2@host2.dom:/dest/directory
```

Here, scp <==> rcp and ssh <==> rsh.

The following comparative information on copying a whole subdirectory was presented by Manoj Srivastava <srivasta@debian.org> to debian-user@lists.debian.org.

### 8.3.2 cp

Traditionally, cp was not really a candidate for this task since it did not dereference symbolic links, or preserve hard links either. Another thing to consider was sparse files (files with holes).

GNU cp has overcome these limitations; however, on a non-GNU system, cp could still have problems. Also, you can't generate small, portable archives using cp.

```
% cp -a . newdir
```

### 8.3.3 tar

Tar overcame some of the problems that cp had with symbolic links. However, although cpio handles special files, traditional tar doesn't.

tar's way of handling multiple hard links to a file places only one copy of the link on the tape, but the name attached to that copy is the only one you can use to retrieve the file; cpio's way puts one copy for every link, but you can retrieve it using any of the names.

The tar command changed its option for .bz2 files between Potato and Woody, so use —bzip2 in scripts instead of its short form -I (Potato) or -j (Woody).

### 8.3.4 pax

The new, POSIX (IEEE Std 1003.2-1992, pages 380–388 (section 4.48) and pages 936–940 (section E.4.48)), all-singing, all-dancing, Portable Archive Interchange utility. pax will read, write, and list the members of an archive file, and will copy directory hierarchies. pax operation is independent of the specific archive format, and supports a wide variety of different archive formats.

pax implementations are still new and wet behind the ears.

```
# apt-get install pax
$ pax -rw -p e . newdir
 or
$ find . -depth  | pax -rw -p e  newdir
```

### 8.3.5 cpio

cpio copies files into or out of a cpio or tar archive. The archive can be another file on the disk, a magnetic tape, or a pipe.

```
$ find . -depth -print0 | cpio --null --sparse -pvd new-dir
```

### 8.3.6 afio

afio is a better way of dealing with cpio-format archives. It is generally faster than cpio, provides more diverse magnetic tape options and deals somewhat gracefully with input data corruption. It supports multivolume archives during interactive operation. afio can make compressed archives that are much safer than compressed tar or cpio archives. afio is best used as an "archive engine" in a backup script.

```
$ find . -depth -print0 | afio -px -0a new-dir
```

All my backups onto tape use afio.

## 8.4 Differential backup and data synchronization

Differential backup and data synchronization can be implemented with several methods:

- rcs: backup and history, text-only
- rdiff-backup: backup and history. symlink OK.
- pdumpfs: backup and history within a filesystem. symlink OK
- rsync: 1-way synchronization
- unison: 2-way synchronization
- cvs: multi-way synchronization with server backup and history, text-only, mature. See 'Concurrent Versions System (CVS)' на стр. 213.
- arch: multi-way synchronization with server backup and history, no such thing as a "working directory".
- subversion: multi-way synchronization with server backup and history, Apache.

Combination of one of these with the archiving method described in 'Copy and archive a whole subdirectory' на стр. 113 and the automated regular job described in 'Schedule activity (cron, at)' на стр. 127 will make a nice backup system.

I will explain three easy-to-use utilities.

### 8.4.1 Differential backup with rdiff

rdiff-backup offers nice and simple backup with differential history for any types of files, including symlinks. To back up most of ~/ to /mnt/backup:

```
$ rdiff-backup --include ~/tmp/keep --exclude ~/tmp  ~/ /mnt/backup
```

To restore three-day-old data from this archive to ~/old:

```
$ rdiff-backup -r 3D /mnt/backup ~/old
```

See rdiff-backup(1).

### 8.4.2 Daily backup with pdumpfs

pdumpfs is a simple daily backup system similar to Plan9's dumpfs which preserves every daily snapshot. You can access the past snapshots at any time for retrieving a certain day's file. Let's backup your home directory with pdumpfs and cron!

pdumpfs constructs the snapshot YYYY/MM/DD in the destination directory. All source files are copied to the snapshot directory when pdumpfs is run for the first time. On and after the second time, pdumpfs copies only updated or newly created files and stores unchanged files as hard links to the files of the previous day's snapshot in order to save disk space.

```
$ pdumpfs src-dir dest-dir [dest-basename]
```

See pdumpfs(8).

### 8.4.3 Regular differential backup with RCS

Changetrack will record changes to the text-based configuration files in RCS archives regularly. See changetrack(1).

```
# apt-get install changetrack
# vi changetrack.conf
```

## 8.5 System freeze recovery

### 8.5.1 Kill a process

Run top to see what process is acting funny. Press 'P' to sort by CPU usage, 'M' to sort by memory, and 'k' to kill a process. Alternatively, BSD-style ps aux | less or System-V-style ps

-efH | less may be used. The System-V-style syntax displays parent process IDs (PPID) which can be used for killing zombie (defunct) children.

Use kill to kill (or send a signal to) a process by process ID, killall to do the same by process command name. Frequently used signals:

```
 1: HUP,  restart daemon
15: TERM, normal kill
 9: KILL, kill hard
```

### 8.5.2 Alt-SysRq

Insurance against system malfunction is provided by the kernel compile option "Magic SysRq key". Pressing Alt-SysRq on an i386, followed by one of the keys r 0 k e i s u b, does the magic.

Un'r'aw restores the keyboard after things like X crashes. Changing the console loglevel to '0' reduces error messages. sa'k' (system attention key) kills all processes on the current virtual console. t'e'rminate kills all processes on the current terminal except init. k'i'll kills all processes except init.

'S'ync, 'u'mount, and re'b'oot are for getting out of really bad situations.

Debian default installation kernels are not compiled with this option at the time this document is written. Recompile the kernel to activate this function. Detailed information is in /usr/share/doc/kernel-doc-version/Documentation/sysrq.txt.gz or /usr/src/kernel-version /Documentation/sysrq.txt.gz.

## 8.6 Nifty little commands to remember

### 8.6.1 Pager

less is the default pager (file content browser). Hit 'h' for help. It can do much more than more. less can be supercharged by executing eval $(lesspipe) or eval $(lessfile) in the shell startup script. See more in /usr/share/doc/lessf/LESSOPEN. The -R option allows raw character output and enables ANSI color escape sequences. See less(1).

w3m may be a useful alternative pager for some code systems (EUC).

### 8.6.2 Free memory

free and top give good information on memory resources. Do not worry about the size of "used" in the "Mem:" line, but read the one under it (38792 in the example below).

```
$ free -k # for 256MB machine
             total       used       free     shared    buffers cached
```

```
Mem:      257136    230456    26680     45736    116136 75528
-/+ buffers/cache:   38792    218344
Swap:     264996         0    264996
```

The exact amount of physical memory can be confirmed by grep '^Memory' /var/log/dmesg, which in this case gives "Memory: 256984k/262144k available (1652k kernel code, 412k reserved, 2944k data, 152k init)".

```
Total         = 262144k = 256M (1k=1024, 1M=1024k)
Free to dmesg = 256984k = Total - kernel - reserved - data - init
Free to shell = 257136k = Total - kernel - reserved - data
```

About 5MB is not usable by the system because the kernel uses it.

### 8.6.3 Set time (BIOS)

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

This will set system and hardware time to MM/DD hh:mm, CCYY. Times are displayed in local time but hardware time uses UTC.

### 8.6.4 Set time (NTP)

Reference: Managing Accurate Date and Time HOWTO (http://www.tldp.org/HOWTO/TimePrecision-HOWTO/index.html).

#### Set time with permanent Internet connection

Set system clock to the correct time automatically via a remote server:

```
# ntpdate server
```

This is good to have in /etc/cron.daily/ if your system has a permanent Internet connection.

#### Set time with sporadic Internet connection

Use the chrony package.

### 8.6.5 How to control console features such as the screensaver

For disabling the screensaver, use following commands.

In the Linux console:

```
# setterm -powersave off
```

Start the kon2 (kanji) console with:

```
# kon -SaveTime 0
```

While running X:

```
# xset s off
 or
# xset -dpms
 or
# xscreensaver-command -prefs
```

Read the corresponding manpages for controlling other console features. See also stty(1) for changing and printing terminal line settings.

### 8.6.6 Search administrative database

Glibc offers getent(1) for searching entries from administrative databases, i.e., passwd, group, hosts, services, protocols, or networks.

```
getent database [key ...]
```

### 8.6.7 Disable sound (beep)

One can always unplug the PC speaker. ;-) For the Bash shell:

```
echo "set bell-style none">> ~/.inputrc
```

### 8.6.8 Error messages on the console screen

In order to quiet on-screen error messages, the first place to check is /etc/init.d/klogd. Set KLOGD="-c 3“ in this script and run /etc/init.d/klogd restart. An alternative method is to run dmesg -n3.

Here error levels mean:

- 0: KERN_EMERG, system is unusable
- 1: KERN_ALERT, action must be taken immediately
- 2: KERN_CRIT, critical conditions
- 3: KERN_ERR, error conditions
- 4: KERN_WARNING, warning conditions
- 5: KERN_NOTICE, normal but significant condition
- 6: KERN_INFO, informational
- 7: KERN_DEBUG, debug-level messages

If one particular useless error message bothers you a lot, consider making a trivial kernel patch like shutup-abit-bp6 (available in the examples subdirectory (http://www.debian.org/doc/manuals/debian-reference/examples/)).

Another place to look may be /etc/syslog.conf; check to see whether any messages are logged to a console device.

### 8.6.9 Set console to the correct type

Console screens in Unix-like systems are usually accessed using (n)curses library routines. These give the user a terminal-independent method of updating character screens with reasonable optimization. See ncurses(3X) and terminfo(5).

On a Debian system, there are quite a lot of predefined entries:

```
$ toe | less                    # all entries
$ toe /etc/terminfo/ | less     # user reconfigurable entries
```

Export your selection as environment variable TERM.

If the terminfo entry for xterm doesn't work with a non-Debian xterm, change your terminal type from "xterm" to one of the feature-limited versions such as "xterm-r6" when you log in to a Debian system remotely. See /usr/share/doc/libncurses5/FAQ for more. "dumb" is the lowest common denominator for terminfo.

### 8.6.10 Get the console back to a sane state

When the screen goes berserk after cat some-binary-file (you may not be able to see the command echoed as you type):

```
$ reset
```

### 8.6.11 Convert a text file from DOS to Unix style

Convert a DOS text file (end-of-line = ^M^J) to a Unix text file (end-of-line = ^J).

```
# apt-get install sysutils
$ dos2unix dosfile
```

### 8.6.12 Regular-expression substitution

Replace all instances of FROM_REGEX with TO_TEXT in all of the files FILES . . . :

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' FILES ...
```

-i is for "in-place editing", -p is for "implicit loop over FILES . . . ". If the substitution is complex, you can make recovery from errors easier by using the parameter -i.bak instead of -i; this will keep each original file, adding .bak as a file extension.

### 8.6.13 Edit a file in place using a script

The following script will remove lines 5–10 and lines 16–20 in place.

```
#!/bin/bash
ed $1 <<EOF
16,20d
5,10d
w
q
EOF
```

Here, ed commands are the same as vi command-mode commands. Editing from the back of file makes it easy for scripting.

### 8.6.14 Extract differences and merge updates for source files

Following one of these procedures will extract differences between two source files and create unified diff files file.patch0 or file.patch1 depending on the file location:

```
$ diff -u file.old file.new1 > file.patch0
$ diff -u old/file new1/file > file.patch1
```

The diff file (alternatively called patch file) is used to send a program update. The receiving party will apply this update to another file by:

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

If you have three versions of source code, you can merge them more effectively using diff3:

```
$ diff3 -m file.mine file.old file.yours > file
```

### 8.6.15 Convert a large file into small files

```
$ split -b 650m file   # split file into 650MB chunks
$ cat x* >largefile    # merge files into 1 large file
```

### 8.6.16 Extract data from text file table

Let's consider a text file called DPL in which all previous Debian project leader's names and their initiation days are listed in a space-separated format.

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   January 1998
Wichert Akkerman  January 1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr March   2003
```

Awk is frequently used to extract data from these types of files.

```
$ awk '{ print $3 }' <DPL                   # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL          # DPL called Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Shells such as Bash can be also used to parse this kind of file:

```
$ while read first last month year; do
    echo $month
  done <DPL
... same output as the first Awk example
```

Here, read built-in command uses the characters in $IFS (internal field separators) to split lines into words.

If you change IFS to ":", you can parse /etc/passwd with shell nicely:

```
$ oldIFS="$IFS"   # save old value
$ IFS=":"
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "osamu" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
osamu's ID is 1001
$ IFS="$oldIFS"   # restore old value
```

(If Awk is used to do the equivalent, use FS=":" to set the field separator.)

IFS is also used by the shell to split results of parameter expansion, command substitution, and arithmetic expansion. These do not occur within double or single quoted words. The default value of IFS is <space>, <tab> and <newline> combined.

Be careful about using this shell IFS tricks. Strange things may happen, when shell interprets some parts of the script as its input.

```
$ IFS=":,"                       # use ":" and "," as IFS
$ echo IFS=$IFS,   IFS="$IFS"     # echo is a Bash built-in
IFS=  , IFS=:,
$ date -R                        # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                # sub shell --> input to main shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                      # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

### 8.6.17 Script snippets for piping commands

The following scripts will do nice things as a part of a pipe.

```
find /usr | egrep -v "/usr/var|/usr/tmp|/usr/local"
                    # find all files in /usr excluding some files
xargs -n 1 command   # run command for all items from stdin
xargs -n 1 echo |    # split white-space-separated items into lines
xargs echo      |    # merge all lines into a line
grep -e pattern|     # extract lines containing pattern
cut -d: -f3 -|
        # extract third field separated by : (passwd file etc.)
awk '{ print $3 }' | # extract third field separated by whitespaces
awk -F'\t' '{ print $3 }' |
        # extract third field separated by tab
```

```
col -bx |           # remove backspace and expand tabs to spaces
expand -|           # expand tabs
sort -u|            # sort and remove duplicates

tr '\n' ' '|        # concatenate lines into one line
tr '\r' ''|         # remove CR
tr 'A-Z' 'a-z'|     # convert uppercase to lowercase
sed 's/^/# /'|      # make each line a comment
sed 's/\.ext//g'|   # remove .ext
sed  -n -e 2p|      # print the second line
head -n 2 -|        # print the first 2 lines
tail -n 2 -|        # print the last 2 lines
```

### 8.6.18 Script snippets for looping over each file

The following ways of looping over each file matching *.ext ensures proper handling of funny file names such as ones with spaces and performs equivalent process:

- Shell loop:

```
for x in *.ext; do
  if test -f "$x"; then
    command "$x"
  fi
done
```

- find and xargs combination:

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | \
 xargs -0 -n 1 command
```

- find with -exec option with a command:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec command '{}' \;
```

- find with -exec option with a short shell script:

```
find . -type f -maxdepth 1 -name '*.ext' \
 -exec sh -c "command '{}' && echo 'successful'" \;
```

### 8.6.19 Perl short script madness

Although any Awk scripts can be automatically rewritten in Perl using a2p(1), one-liner Awk scripts are best converted to one-liner perl scripts manually. For example

```
awk '($2=="1957") { print $3 }' |
```

is equivalent to any one of the following lines:

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

Since all the whitespace in the arguments to perl in the line above can be removed, and taking advantage of the automatic conversions between numbers and strings in Perl:

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

See perlrun(1) for the command-line options. For more crazy Perl scripts, http://perlgolf.sourceforge.net may be interesting.

### 8.6.20 Get text or a mailing list archive from a web page

The following will read a web page into a text file. Very useful when copying configurations off the Web.

```
$ lynx -dump http://www.remote-site.com/help-info.html >textfile
```

links and w3m can be used here, too, with slight differences in rendering.

If this is a mailing list archive, use munpack to obtain mime contents from text.

### 8.6.21 Pretty print a web page

The following will print a web page into a PostScript file/printer.

```
$ apt-get install html2ps
$ html2ps URL | lpr
```

See 'lpr/lpd' на стр. 42. Also check a2ps and mpage packages for creating PostScript files.

### 8.6.22 Pretty print a manual page

The following will print a manual page into a PostScript file/printer.

```
$ man -Tps some-manpage | lpr
$ man -Tps some-manpage | mpage -2 | lpr
```

### 8.6.23 Merge two PostScript or PDF files

You can merge two PostScript or PDF files.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite \
  -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite \
  -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

### 8.6.24 Time a command

Display time used by a process.

```
# time some-command >/dev/null
real    0m0.035s       # time on wall clock (elapsed real time)
user    0m0.000s       # time in user mode
sys     0m0.020s       # time in kernel mode
```

### 8.6.25 nice command

Use nice (from the GNU shellutils package) to set a command's nice value when starting. renice (bsdutils) or top can renice a process. A nice value of 19 represents the slowest (lowest priority) process; negative values are "not-nice", with -20 being a very fast (high priority) process. Only the superuser can set negative nice values.

```
# nice  -19 top                                      # very nice
# nice --20 cdrecord -v -eject speed=2 dev=0,0 disk.img # very fast
```

Sometimes an extreme nice value does more harm than good to the system. Use this command carefully.

### 8.6.26 Schedule activity (cron, at)

Use cron and at to schedule tasks under Linux. See at(1), crontab(5), crontab(8).

Run the command crontab -e to create or edit a crontab file to set up regularly scheduled events. Example of a crontab file:

```
# use /bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to 'paul', no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
# run at 00:05, every day
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1  * *   $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Run the at command to schedule a one-time job:

```
$ echo 'command -args'| at 3:40 monday
```

### 8.6.27 Console switching with screen

The screen program allows you to run multiple virtual terminals, each with its own interactive shell, on a single physical terminal or terminal emulation window. Even if you use Linux virtual consoles or multiple xterm windows, it is worth exploring screen for its rich feature set, which includes

- scrollback history,
- copy-and-paste,
- output logging,
- digraph entry, and
- the ability to detach an entire screen session from your terminal and reattach it later.

#### Remote access scenario

If you frequently log on to a Linux machine from a remote terminal or using a VT100 terminal program, screen will make your life much easier with the detach feature.

1 You are logged in via a dialup connection, and are running a complex screen session with editors and other programs open in several windows.

2 Suddenly you need to leave your terminal, but you don't want to lose your work by hanging up.
3 Simply type ^A d to detach the session, then log out. (Or, even quicker, type ^A DD to have screen detach and log you out itself.)
4 When you log on again later, enter the command screen -r, and screen will magically reattach all the windows you had open.

Typical screen commands

Once you start screen, all keyboard input is sent to your current window except for the command keystroke, by default ^A. All screen commands are entered by typing ^A plus a single key [plus any parameters]. Useful commands:

```
^A ?     show a help screen (display key bindings)
^A c     create a new window and switch to it
^A n     go to next window
^A p     go to previous window
^A 0     go to window number 0
^A w     show a list of windows
^A a     send a Ctrl-A to current window as keyboard input
^A h     write a hardcopy of current window to file
^A H     begin/end logging current window to file
^A ^X    lock the terminal (password protected)
^A d     detach screen session from the terminal
^A DD    detach screen session and log out
```

This is only a small subset of screen's commands and features. If there's something you want screen to be able to do, chances are it can! See screen(1) for details.

Backspace and/or Ctrl-H in screen session

If you find that backspace and/or Ctrl-H do not work properly when you are running screen, edit /etc/screenrc, find the line reading

```
bindkey -k kb stuff "\177"
```

and comment it out (i.e., add "#" as the first character).

Equivalent program to screen for X

Check out xmove. See xmove(1).

### 8.6.28 Network testing basics

Install netkit-ping, traceroute, dnsutils, ipchains (for 2.2 kernel), iptables (for 2.4 kernel), and net-tools packages and:

```
$ ping yahoo.com            # check Internet connection
$ traceroute yahoo.com      # trace IP packets
$ ifconfig                  # check host config
$ route -n                  # check routing config
$ dig [@dns-server.com] host.dom [{a|mx|any}] |less
      # check host.dom DNS records  by dns-server.com
      # for a {mx|any} record
$ ipchains  -L -n |less     # check packet filter (2.2 kernel)
$ iptables -L -n |less      # check packet filter (2.4 kernel)
$ netstat -a                # find all open ports
$ netstat -l --inet         # find listening ports
$ netstat -ln --tcp         # find listening TCP ports (numeric)
```

### 8.6.29 Flush mail from local spool

To flush mail from the local spool:

```
# exim -q    # flush waiting mail
# exim -qf   # flush all mail
# exim -qff  # flush even frozen mail
```

-qff may be better as an option in the /etc/ppp/ip-up.d/exim script.

### 8.6.30 Remove frozen mail from local spool

To remove frozen mail from the local spool with a delivery error message:

```
# exim -Mg `mailq | grep frozen | awk '{ print $3 }'`
```

### 8.6.31 Redeliver mbox contents

You need to manually deliver mails to the sorted mailboxes in your home directory from /var /mail/username if your home directory became full and procmail failed. After making disk space in the home directory, run:

```
# /etc/init.d/exim stop
# formail -s procmail </var/mail/username
# /etc/init.d/exim start
```

### 8.6.32 Clear file contents

In order to clear the contents of a file such as a logfile, do not use rm to delete the file and then create a new empty file, because the file may still be accessed in the interval between commands. The following is the safe way to clear the contents of the file.

```
$ :>file-to-be-cleared
```

### 8.6.33 Dummy files

The following commands will create dummy or empty files:

```
$ dd if=/dev/zero    of=filename bs=1k count=5 # 5KB of zero content
$ dd if=/dev/urandom of=filename bs=1M count=7 # 7MB of random content
$ touch filename # create 0B file (if file exists, updates mtime)
```

For example, the following commands executed from the shell of the Debian boot floppy will erase all the content of the hard disk /dev/hda completely for most practical uses.

```
# dd if=/dev/urandom of=/dev/hda ; dd if=/dev/zero of=/dev/hda
```

### 8.6.34 chroot

The chroot program, chroot(8), enables us to run different instances of the GNU/Linux environment on a single system simultaneously without rebooting.

One may also run a resource hungry program such as apt-get or dselect under the chroot of a fast host machine while NFS-mounting a slow satellite machine to the host as r/w and the chroot point being the mount point of the satellite machine.

#### Run a different Debian flavor with chroot

A chroot Debian environment can easily be created by the debootstrap command in Woody. For example, to create a Sid chroot on /sid-root while having fast Internet access:

```
main # cd / ; mkdir /sid-root
main # debootstrap sid /sid-root http://ftp.debian.org/debian/
... watch it download the whole system
main # echo "proc-sid /sid-root/proc proc none 0 0" >> /etc/fstab
main # mount proc-sid /sid-root/proc -t proc
main # cp /etc/hosts /sid-root/etc/hosts
main # chroot /sid-root /bin/bash
chroot # cd /dev; /sbin/MAKEDEV generic ; cd -
```

```
chroot # apt-setup # set-up /etc/apt/sources.list
chroot # vi /etc/apt/sources.list # point the source to unstable
chroot # dselect  # you may use aptitude, install mc and vim :-)
```

At this point you should have a fully working Debian system, where you can play around without fear of affecting your main Debian installation.

This debootstrap trick can also be used to install Debian to a system without using a Debian install disk, but instead one for another GNU/Linux distribution. See http://www.debian.org/releases/stable/i386/apcs04.

### Setting up login for chroot

Typing chroot /sid-root /bin/bash is easy, but it retains all sorts of environment variables that you may not want, and has other issues. A much better approach is to run another login process on a separate virtual terminal where you can log in to the chroot directly.

Since on default Debian systems tty1 to tty6 run Linux consoles and tty7 runs the X Window System, let's set up tty8 for a chrooted console as an example. After creating a chroot system as described in 'Run a different Debian flavor with chroot' на стр. 130, type from the root shell of the main system:

```
main # echo "8:23:respawn:/usr/sbin/chroot /sid-root "\
      "/sbin/getty 38400 tty8"  >> /etc/inittab
main # init q    # reload init
```

### Setting up X for chroot

You want to run the latest X and GNOME safely in your chroot? That's entirely possible! The following example will make GDM run on virtual terminal vt9.

First install a chroot system using the method described in 'Run a different Debian flavor with chroot' на стр. 130. From the root of the main system, copy key configuration files to the chroot system.

```
main # cp /etc/X11/XF86Config-4 /sid-root/etc/X11/XF86Config-4
main # chroot /sid-root # or use chroot console
chroot # cd /dev; /sbin/MAKEDEV generic ; cd -
chroot # apt-get install gdm gnome x-window-system
chroot # vi /etc/gdm/gdm.conf # do s/vt7/vt9/ in [servers] section
chroot # /etc/init.d/gdm start
```

Here, /etc/gdm/gdm.conf was edited to change the first virtual console from vt7 to vt9.

Now you can easily switch back and forth between full X environments in your chroot and your main system just by switching between Linux virtual terminals; e.g. by using Ctrl-Alt-F7 and Ctrl-Alt-F9. Have fun!

[FIXME] Add a comment and link to the init script of the chrooted gdm.

Run other distributions with chroot

A chroot environment for another Linux distribution can easily be created. You install a system into separate partitions using the installer of the other distribution. If its root partition is in /dev/hda9:

```
main # cd / ; mkdir /other-dist
main # mount -t ext3 /dev/hda9 /other-dist
main # chroot /other-dist /bin/bash
```

Then proceed as in 'Run a different Debian flavor with chroot' на стр. 130, 'Setting up login for chroot' на стр. 131, and 'Setting up X for chroot' на стр. 131.

Build a package with chroot

There is a more specialized chroot package, pbuilder, which constructs a chroot system and builds a package inside the chroot. It is an ideal system to use to check that a package's build-dependencies are correct, and to be sure that unnecessary and wrong build dependencies will not exist in the resulting package.

### 8.6.35 How to check hard links

You can check whether two files are the same file with two hard links by:

```
$ ls -li file1 file2
```

### 8.6.36 mount hard disk image file

If file.img contains an image of hard disk contents and the original hard disk had a disk configuration which gives xxxx = (bytes/sector) * (sectors/cylinder), then the following will mount it to /mnt:

```
# mount -o loop,offset=xxxx file.img /mnt
```

Note that most hard disks have 512 bytes/sector.

### 8.6.37 Samba

Basics of getting files from Windows:

```
# mount -t smbfs -o username=myname,uid=my_uid,gid=my_gid \
      //server/share /mnt/smb  # mount Windows files to Linux
# smbmount //server/share /mnt/smb \
      -o "username=myname,uid=my_uid,gid=my_gid"
# smbclient -L 192.168.1.2 # list the shares on a computer
```

Samba neighbors can be checked from Linux using:

```
# smbclient -N -L ip_address_of_your_PC | less
# nmblookup -T "*"
```

### 8.6.38 Utilities for foreign filesystems

Many foreign filesystems have Linux kernel support, and can thus be accessed simply by mounting the devices containing the filesystems. For certain filesystems, there are also a few specialized tools to access the filesystems without mounting the devices. This is accomplished with user-space programs so that kernel filesystem support is not needed.

- mtools: for MSDOS filesystem (MS-DOS, Windows)
- cpmtools: for CP/M filesystem
- hfsutils: for HFS filesystem (native Macintosh)
- hfsplus: for HFS+ filesystem (modern Macintosh)

In order to create and check an MS-DOS FAT filesystem, dosfstools is useful.

## 8.7 Typical mistakes to be noted

Here are few examples of dangerous actions. The negative impacts will be enhanced if you are using privileged account: root.

### 8.7.1 rm -rf .*

In ”rm -rf .*“, ”.*“ expands to include ”.“ and ”..“, and if you happen to have privileges to write to the parent directory then you’ll end up removing all directories next to your current directory as well.

- ”rm -rf .“ : removes everything under current directory and current directory itself.
- ”rm -rf *“ : removes every non-dot files and non-dot directories under current directory
- ”rm -rf .[^.]*“ : removes every dot files and dot-directories under current directory.
- ”rm -rf .*“ : removes everything under parent directory and parent directory itself.

### 8.7.2 rm /etc/passwd

Loss of some important files such as /etc/passwd through your stupidity is tough. The Debian system makes regular backups of them in /var/backups. When you restore these files, you may manually have to set the proper permissions.

```
# cp /var/backups/passwd /etc/passwd
# chmod 644 /etc/passwd
```

See also ‘Восстановление информации о выбранных пакетах’ на стр. 90.

# Глава 9

# Tuning a Debian system

This chapter describes only the basics of system configuration through a command-line interface. Before reading this chapter you should read 'Советы по установке системы Debian' на стр. 27.

If you are concerned about security then you should read the Securing Debian Manual (http://www.debian.org/doc/manuals/securing-debian-howto/) which comes in the harden-doc package.

## 9.1 System initialization

Debian uses the System V init script system. See 'Программа init' на стр. 22 for an introduction.

### 9.1.1 Customizing init scripts

The easiest way to control the behavior of an init script is by changing environment variable assignments in the file named like the init script in the /etc/default directory. [1] For example, /etc/default/hotplug can be used to control how /etc/init.d/hotplug works. The /etc/default/rcS file can be used to customize boot-time defaults for motd, sulogin, etc.

If you cannot get the behavior you want by changing such variables then you can modify the init scripts themselves: they are all configuration files.

### 9.1.2 Customizing system logging

System log mode can be configured using /etc/syslog.conf. Check the colorize package for a program to colorize system logfiles. See also syslogd(8) and syslog.conf(5).

[1] Files in /etc/default/ contain environment variable assignments only. Each file is sourced by the init script to which it corresponds in such a way that these assignments override any default variable settings in the init script itself. The choice of directory name is peculiar (http://lists.debian.org/debian-devel/2003/debian-devel-200308/msg02114.html) to Debian. It is roughly the equivalent of the /etc/sysconfig directory found in RedHat and other distributions.

### 9.1.3 Optimizing hardware

There are a few hardware optimization configurations that Debian leaves to the sysadmin to take care of.

- hdparm
  - Hard disk access optimization. Very effective.
  - Dangerous. You must read hdparm(8) first.
  - hdparm -tT /dev/hda to test disk access speed.
  - hdparm -q -c3 -d1 -u1 -m16 /dev/hda to speed up a modern IDE system. (It may be dangerous.)
- setcd
  - Compact disc drive access optimization.
  - setcd -x 2 to slow down to 2x speed.
  - See setcd(1).
- setserial
  - Collection of tools for serial port management.
- scsitools
  - Collection of tools for SCSI hardware management.
- memtest86
  - Collection of tools for memory hardware management.
- hwtools
  - Collection of tools for low-level hardware management.
    * irqtune: changes the IRQ priority of devices to allow devices that require high priority and fast service (e.g. serial ports, modems) to have it. 3x speedup of serial/modem throughput is possible.
    * scanport: scans I/O space from 0x100 to 0x3ff looking for installed ISA devices.
    * inb: a quick little hack that reads an I/O port and dumps the value in hex and binary.
- schedutils
  - Linux scheduler utilities.
  - taskset, irqset, lsrt, and rt are included.
  - Together with nice and renice (not included), they allow full control of process scheduling parameters.

Mounting a filesystem with the noatime option is also very effective in speeding up read access to the file. See fstab(5) and mount(8).

Some hardware can be tuned directly by the Linux kernel itself through the proc filesystem. See 'Tuning the kernel through the proc filesystem' на стр. 107.

There are many hardware-specific configuration utilities in Debian. Many of them address needs specific to the laptop PC. Here are some interesting packages available in Debian:

- tpconfig - A program to configure touchpad devices
- apmd - Utilities for Advanced Power Management (APM)
- acpi - displays information on ACPI devices
- acpid - Utilities for using ACPI
- lphdisk - prepares hibernation partition for Phoenix NoteBIOS
- sleepd - puts a laptop to sleep during inactivity
- noflushd - allow idle hard disks to spin down
- big-cursor - larger mouse cursors for X
- acme - Enables the "multimedia buttons" found on laptops
- tpctl - IBM ThinkPad hardware configuration tools
- mwavem - Mwave/ACP modem support
- toshset - Access much of the Toshiba laptop hardware interface
- toshutils - Toshiba laptop utilities
- sjog - A program to use the "Jog Dial" on Sony Vaio Laptops
- spicctrl - Sony Vaio controller program to set LCD backlight brightness

Here, ACPI is a newer framework for the power management system than APM.

Some of these packages require special kernel modules. They are already included in the latest kernel source in many cases. In case of trouble, you may need to apply the latest patch to the kernel yourself.

## 9.2 Restricting access

### 9.2.1 Restricting logins with PAM

PAM (Pluggable Authentication Modules) allow you to control how users log in.

```
/etc/pam.d/*             # PAM control files
/etc/pam.d/login         # PAM control file for login
/etc/security/*          # PAM module parameters
/etc/securetty           # this controls root login by console (login)
/etc/login.defs          # this controls login behaviors (login)
```

Change the contents of /etc/pam.d/login as follows, if you want insecure but passwordless console terminals at your own risk.

```
#auth       required   pam_unix.so nullok
auth       required   pam_permit.so
```

Similar tricks can be applied for xdm, gdm, ..., for passwordless console X.

On the other hand, install cracklib2 and set /etc/pam.d/passwd as follows, if you want to enforce a good password policy.

```
password required       pam_cracklib.so retry=3 minlen=6 difok=3
```

A one-time login password for account activation may also help. For this, use the passwd command with the -e option. See passwd(1).

The maximum number of processes can be set with ulimit -u 1000 in a Bash shell or with settings in /etc/security/limits.conf from PAM. Other parameters such as core can be set similarly. The initial value of PATH can be set by /etc/login.defs before the shell startup script.

The documentation for PAM is packaged in the libpam-doc package. The Linux-PAM System Administrator's Guide covers configuring PAM, what modules are available, etc. The documentation also includes The Linux-PAM Application Developers' Guide and The Linux-PAM Module Writers' Guide.

### 9.2.2 "Why GNU su does not support the wheel group"

This is the famous phrase at the bottom of the old info su page by Richard M. Stallman. Not to worry: the current su in Debian uses PAM, so that one can restrict the ability to use su to any group using pam_wheel.so in /etc/pam.d/su. The following will set the adm group in a Debian system as an equivalent of the BSD wheel group and allow su without a password for its members.

```
# anti-RMS configuration in /etc/pam.d/su
auth       required   pam_wheel.so group=adm

# Wheel members to be able to su without a password
auth       sufficient pam_wheel.so trust group=adm
```

### 9.2.3 Purposes of standard groups

A few interesting groups:

- root group is the default wheel group for su if pam_wheel.so is used without the group= argument.
- adm group can read logfiles.
- cdrom group can be used locally to give a set of users access to a CD-ROM drive.
- floppy group can be used locally to give a set of users access to a floppy drive.
- audio group can be used locally to give a set of users access to an audio device.
- src group owns source code, including files in /usr/src. It can be used locally to give a user the ability to manage system source code.
- staff membership is useful for helpdesk types or junior sysadmins, giving them the ability to do things in /usr/local and to create directories in /home.

For a complete list, see the "FAQ" section in the Securing Debian Manual (http://www.debian.org/doc/manuals/securing-debian-howto/), which can also be found as the harden-doc package in Woody. Also the new base-passwd (>3.4.6) contains an authoritative list: /usr/share/doc/base-passwd/users-and-groups.html.

### 9.2.4 Working more safely – sudo

My usage of sudo is mostly a protection from my own stupidity. Personally, I consider using sudo a better alternative to always using the system as root.

Install sudo and activate it by setting options in /etc/sudoers(http://www.debian.org/doc/manuals/debian-reference/examples/). Also check out the sudo group feature in /usr/share/doc/sudo/OPTIONS.

The sample configuration provides "staff" group members access to any commands run as root under sudo and also gives "src" members access to selected commands run as root under sudo.

The advantage of sudo is that it only requires an ordinary user's password to log in, and activity is monitored. This is a nice way to give some authority to a junior administrator. For example:

```
$ sudo chown -R myself:mygrp .
```

Of course if you know the root password (as most home users do), any command can be run under root from a user account:

```
$ su -c "shutdown -h now"
Password:
```

(I know I should tighten the admin account's sudo privileges. But since this is my home server, I have not bothered yet.)

For a different program that allows ordinary users to run commands with root privileges, see the super package.

### 9.2.5 Restricting access to services

The Internet super-server, inetd, is started at boot time by /etc/rc2.d/S20inetd (for RUNLEVEL=2), which is a symlink to /etc/init.d/inetd. Essentially, inetd allows one running daemon to invoke several others, reducing load on the system.

Whenever a request for service arrives, its protocol and service are identified by looking them up in the databases in /etc/protocols and /etc/services. inetd then looks up a normal Internet service in the /etc/inetd.conf database, or a Sun-RPC based service in /etc/rpc.conf.

For system security, make sure to disable unused services in /etc/inetd.conf. Sun-RPC services need to be active for NFS and other RPC-based programs.

Sometimes, inetd does not start the intended server directly but starts the tcpd TCP/IP daemon wrapper program with the intended server name as its argument in /etc/inetd.conf. In this case, tcpd runs the appropriate server program after logging the request and doing some additional checks using /etc/hosts.deny and /etc/hosts.allow.

If you have problems with remote access in a recent Debian system, comment out "ALL: PARANOID" in /etc/hosts.deny if it exists.

For details, see inetd(8), inetd.conf(5), protocols(5), services(5), tcpd(8), hosts_access(5), and hosts_options(5).

For more information on Sun-RPC, see rpcinfo(8), portmap(8), and /usr/share/doc/portmap /portmapper.txt.gz.

### 9.2.6 Centralizing authentication – LDAP

Use Lightweight Directory Access Protocol (LDAP). References:

- OpenLDAP (http://www.openldap.org/)
- OpenLDAP Admin Guide in the openldap-guide package
- LDP: LDAP Linux HOWTO (http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html)
- LDP: LDAP Implementation HOWTO (http://www.tldp.org/HOWTO/LDAP-Implementation-HOWTO/index.html)
- OpenLDAP, extensive use reports (http://portal.aphroland.org/~aphro/ldap-docs/ldap.html)
- Open LDAP with Courier IMAP and Postfix (http://alinux.washcoll.edu/docs/plc/postfix-courier-howto.html)

## 9.3 CD writers

CD-writers with ATAPI/IDE interfaces have recently become a very popular option. It is a nice medium for system backup and archiving for the home user needing < 640MB capacity. For the most authoritative information, see the LDP CD-Writing-HOWTO (http://www.tldp.org/HOWTO/CD-Writing-HOWTO.html).

### 9.3.1 Introduction

First, any disruption of data sent to the CD-writer will cause irrecoverable damage to the CD. Get a CD-writer with as large a buffer as possible. If money is no object, do not bother with ATAPI/IDE, just get a SCSI version. If you have a choice of IDE interface to be connected, use the one on the PCI-bus (i.e., on the motherboard) rather than one on the ISA-bus (an SB16 card, etc.).

When a CD-writer is connected to IDE, it has to be driven by the IDE-SCSI driver instead of an ordinary IDE CD driver for Linux 2.2 and 2.4 kernels. Also, the SCSI generic driver needs to be activated. There are two possible approaches to doing this, assuming a kernel distributed with modern distributions (as of March 2001).

### 9.3.2 Approach 1: modules + lilo

Add the following line to /etc/lilo.conf if you are using a stock Debian kernel. If multiple options are used, list them separated by spaces:

```
append="hdx=ide-scsi ignore=hdx"
```

Here the location of the CD-writer, which is accessed through the ide-scsi driver, is indicated by hdx, where x represents one of the following:

| | |
|---|---|
| hda | for a master on the first IDE port |
| hdb | for a slave on the first IDE port |
| hdc | for a master on the second IDE port |
| hdd | for a slave on the second IDE port |
| hde ... hdh | for a drive on an external IDE port or ATA66/100 IDE port |

Type the following commands as root to activate after finishing all the configuration:

```
# lilo
# shutdown -h now
```

### 9.3.3 Approach 2: recompile the kernel

Debian uses make-kpkg to create a kernel. Use the new —append_to_version with make-kpkg to build multiple kernel images. See 'The Linux kernel under Debian' на стр. 101.

Use the following setup through make menuconfig:

- bzImage
- Exclude the IDE CD driver (not a must, but simpler to do this)
- Compile in ide-scsi and sg, or make them modules

### 9.3.4 Post-configuration steps

Kernel support for the CD-writer can be activated during booting by the following:

```
# echo ide-scsi >>/etc/modules
# echo sg       >>/etc/modules
# cd /dev; ln -sf scd0 cdrom
```

Manual activation can be done by:

```
# modprobe ide-scsi
# modprobe sg
```

After reboot, you can check installation by:

```
$ dmesg|less
# apt-get install cdrecord
# cdrecord -scanbus
```

[Per Warren Dodge] Sometimes there may be conflicts between ide-scsi and ide-cd if there are both CD-ROM and CD-R/RW on the system. Try adding the following line to your /etc/modutils /aliases, running update-modules, and rebooting.

```
pre-install     ide-scsi     modprobe ide-cd
```

This causes the IDE driver to load before ide-scsi. The IDE driver ide-cd takes control of the ATAPI CD-ROM—anything that it hasn't been told to ignore. That leaves just the ignored devices for ide-scsi to control.

### 9.3.5 CD-image file (bootable)

To create a CD-image of files under target-directory/ as cd-image.raw (bootable, Joliet TRANS.TBL-enabled format; if not bootable, take out -b and -c options), insert a boot floppy in the first floppy drive and

```
# dd if=/dev/fd0 target-directory/boot.img
# mkisofs -r -V volume_id -b boot.img -c bootcatalog -J -T \
      -o cd-image.raw target_directory/
```

One funny hack is to make a bootable DOS CD-ROM. If an ordinary DOS boot floppy disk image is in the above boot.img, the CD-ROM will boot as if a DOS floppy were in the first floppy drive (A:). Doing this with freeDOS may be more interesting.

This CD-image file can be inspected by mounting it on the loop device.

```
# mount -t iso9660 -o ro,loop cd-image.raw /cdrom
# cd /cdrom
# mc
# umount /cdrom
```

### 9.3.6 Write to the CD-writer (R, RW):

First test with (assuming double speed)

```
# nice --10 cdrecord -dummy speed=2 dev=0,0 disk.img
```

Then if OK, write to CD-R with

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Or write to a CD-RW disk with

```
# nice --10 cdrecord -v -eject blank=fast speed=2 dev=0,0 disk.img
```

Some CD-RW drives work better with

```
# nice --10 cdrecord -v blank=all speed=2 dev=0,0 disk.img
```

followed by

```
# nice --10 cdrecord -v -eject speed=2 dev=0,0 disk.img
```

Two steps are needed to prevent SCSI timeouts during blanking from interfering with the burning step. The argument value to nice may require some adjustments.

### 9.3.7 Make an image file of a CD

Some CD-Rs and commercial CDs have junk sectors at the end that make copying by dd impossible (the Windows 98 CD is one of them). The cdrecord package comes with the readcd command. Use this to copy any CD contents to an image file. If it is a data disk, mount it and run df to see its actual size. Divide the number shown in blocks (1 block = 1024 bytes) by 2 to get the number of actual CD sectors (1 sector = 2048 bytes). Run readcd with options and use this disk image to burn the CD-R/RW.

```
# readcd dev=target,lun,scsibusno # select function 11
```

Here, set all three parameters to 0 for most cases. Usually the number of sectors given by readcd is excessive! Use the above number from an actual mount for better results.

It should be noted that the use of dd has a few problems if used on CD-ROM. The first run of the dd command may cause an error message and may yield a shorter disk image with a lost tail-end. The second run of dd command may yield a larger disk image with garbage data attached at the end on some systems if the data size is not specified. Only the second run of the dd command with the correct data size specified, and without ejecting the CD after an error message, seems to avoid these problems. If for example the image size displayed by df is 46301184 blocks, use the following command twice to get the right image (this is my empirical information):

```
# dd if=/dev/cdrom of=cd.img bs=2048 count=$((46301184/2))
```

### 9.3.8 Debian CD images

To obtain the latest information on Debian CDs, visit the Debian CD site (http://www.debian.org/CD/).

If you have a fast Internet connection, think about installing over the network using:

- a few floppy images (http://www.debian.org/distrib/floppyinst).
- a minimal bootable CD image (http://www.debian.org/CD/netinst/).

If you do not have a fast Internet connection, think about purchasing CDs from a CD vendor (http://www.debian.org/CD/vendors/).

Please do not waste bandwidth by downloading standard CD images unless you are a CD image tester (even with the new jigdo method).

One noteworthy CD image is KNOPPIX - Live Linux Filesystem On CD (http://www.knopper.net/knoppix/index-en.html). This CD will boot a functioning Debian system without installing itself to the hard disk.

### 9.3.9 Back up the system to CD-R

To copy key configuration files and data files to CD-R, use the example backup script backup (http://www.debian.org/doc/manuals/debian-reference/examples/). Also see 'Copy and archive a whole subdirectory' на стр. 113 and 'Differential backup and data synchronization' на стр. 115.

### 9.3.10 Copy a music CD to CD-R

Not tested by me:

```
# apt-get install cdrecord cdparanoia
# cdparanoia -s -B
# cdrecord dev=0,0,0 speed=2 -v -dao -eject defpregap=1 -audio *.wav
```

or,

```
# apt-get install cdrdao #disk at once
# cdrdao read-cd --device /dev/cdrom --paranoia-mode 3 my_cd # read cd
# cdrdao write --device /dev/cdrom --speed 8 my_cd     # write a new CD
```

cdrdao does a real copy (no gaps, etc...).

## 9.4 X

The X Window System is provided by XFree86 (http://www.xfree86.org/). There are two major versions of X server available on the Debian system: XFree86 Version 3.3 (XF3) and XFree86 Version 4.x series (XF4) both based on X11R6 specifications by X.Org (http://www.x.org/).

For the basics of X, refer to X(7), the LDP XWindow-User-HOWTO (http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html), and the Remote X Apps mini-HOWTO (http://www.tldp.org/HOWTO/mini/Remote-X-Apps.html). For a Debian-specific user guide, read /usr/share/doc/xfree86-common/FAQ.gz provided in the xfree86-common package. This contains an interesting and authoritative review of the key binding issues by Branden Robinson.

'The X server' на стр. 146 a program on a local host that displays an X window and/or desktop on a user's monitor (CRT, LCD) and accepts keyboard and mouse input.

'X clients' на стр. 148 a program on a (local or remote) host that runs X-compatible application software.

This reverses the ordinary use of "server" and "client" in other contexts.

There are several ways of getting the "X server" (display side) to accept remote connections from an "X client" (application side):

- xhost method
  - the host list mechanism (very insecure).
  - non-encrypted protocol (prone to eavesdropping attack).
  - Do not use this, if possible.
  - See 'Connecting to a remote X server – xhost' на стр. 153 and xhost(1x).
- xauth method
  - the MIT magic cookie mechanism (insecure but better than xhost).
  - non-encrypted protocol (prone to eavesdropping attack).
  - use this only for local connection since it is less CPU-intensive than ssh -X.
  - See 'Getting root in X' на стр. 155 and xauth(1x).
- xdm, wdm, gdm, kdm, ... methods
  - the MIT magic cookie mechanism (insecure as xauth).
  - See xdm(1x) and Xsecurity(7) for the basics of X display access control.
  - See wdm(1x), gdm(8), and kdm.options(5) for more information, if these are installed.
  - See 'System-V init and runlevels' на стр. 100 for how to disable xdm to gain a Linux console upon boot without purging the xdm package.
- ssh -X method
  - port forwarding mechanism through secure shell (secure).
  - encrypted protocol (a waste of resources if used locally).
  - use this for remote connections.
  - See 'Connecting to a remote X server – ssh' на стр. 153.

All remote connection methods, except ssh, require TCP/IP connection enabled on the X server. See 'Using X over TCP/IP' на стр. 153.

### 9.4.1 X packages

There are a few (meta)packages provided to ease installation of the X system in Woody.

x-window-system-core This metapackage provides the essential components for a stand-alone workstation running the X Window System. It provides the X libraries, an X server (xserver-xfree86), a set of fonts, and a group of basic X clients and utilities.

x-window-system This metapackage provides substantially all the components of the X Window System as developed by the XFree86 Project, as well as a set of historically popular accessory programs. (Notably, it depends on x-window-system-core, twm, and xdm, i.e., no need to install x-window-system-core if you install this.)

xserver-common-v3 Files and utilities common to XFree86 3.x X servers (XF3)

xserver-* Supplemental XF3 server packages to support hardware not supported by the new XF4 server (xserver-xfree86) for whatever reason. Some old ATI mach64 cards are not supported in XF4, other cards hang badly in the Woody version of XF4, etc. (For available packages, use apt-cache search xserver-|less. All of these XF3 servers depend on xserver-common-v3.)

For most cases, x-window-system is the package to install. (If you want console login, be sure to disable xdm as described in '"Let me disable X on boot!"' на стр. 111.)

### 9.4.2 Hardware detection for X

To enable hardware detection during the X configuration stage, install the following packages prior to installing the X system.:

- discover – hardware identification system.
- mdetect – mouse device autodetection tool.
- read-edid – hardware information-gathering tool for VESA PnP monitors.

### 9.4.3 The X server

See XFree86(1x) for X server information.

Invoke X server from a local console:

```
$ startx -- :<display> vtXX
e.g.:
$ startx -- :1 vt8 -bpp 16
... start on vt8 connected to localhost:1 with 16 bpp mode
```

Arguments given after — are for the X server.

Note, when using a ~/.xserverrc script to customize the X server startup process, be sure to exec the real X server. Failing to do this can make the X server slow to start and exit. For example:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Configuring the X server (version 4)

To (re-)configure an XF4 server,

```
# dpkg-reconfigure --priority=low xserver-common
# dpkg-reconfigure --priority=low xserver-xfree86
```

will generate /etc/X11/XF86Config-4 file and configure X using the script dexconf.

Configuring the X server (version 3)

To (re-)configure an XF3 server, for example, for ATI mach64,

```
# dpkg-reconfigure --priority=low xserver-common-v3
# dpkg-reconfigure --priority=low xserver-mach64
```

will generate /etc/X11/XF86Config file and configure X using the script xf86config-v3.

Configuring the X server manually

For Woody, to add user customizations to /etc/X11/XF86Config-4 file, do not edit the configuration file between the text:

```
### BEGIN DEBCONF SECTION
[snip]
### END DEBCONF SECTION
```

Instead, add the customizations before the text. For example, to use a custom video device, add something resembling the following text to the top of the file:

```
Section "Device"
  Identifier        "Custom Device"
  Driver            "ati"
  Option            "NoAccel"
```

```
EndSection

Section "Screen"
  Identifier  "Custom Screen"
  Device      "Custom Device"
  Monitor     "Generic Monitor"
  DefaultDepth 24
  Subsection "Display"
    Depth       8
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       16
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
  Subsection "Display"
    Depth       24
    Modes       "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubsection
EndSection

Section "ServerLayout"
    Identifier        "Custom"
    Screen            "Custom Screen"
    InputDevice       "Generic Keyboard" "CoreKeyboard"
    InputDevice       "Configured Mouse" "CorePointer"
EndSection
```

For Sarge (testing at the time of writing), if you wish to retain user customizations to /etc/X11 /XF86Config file through upgrade, run the following commands as root:

```
# cp /etc/X11/XF86Config-4 /etc/X11/XF86Config-4.custom
# md5sum /etc/X11/XF86Config-4 > /var/lib/xfree86/XF86Config-4.md5sum
# dpkg-reconfigure xserver-xfree86
```

In order to achieve font de-uglification, you need to edit /etc/X11/XF86Config-4 as described in 'TrueType fonts in X' на стр. 157.

Please also check the other parts of your X configuration. Bad monitor settings can be even more of a headache than bad fonts, so make sure your refresh rate is as high as your monitor can handle (85 Hz is great, 75 Hz is OK, 60 Hz is painful.).

### 9.4.4 X clients

Most X client programs can be started with a command like this:

```
client $ xterm -geometry 80x24+30+200 -fn 6x10 -display hostname:0 &
```

Here, the optional command-line arguments mean:

- -geometry WIDTHxHEIGHT+XOFF+YOFF: the initial size and location of the window.
- -fn FONTNAME: the font to use for displaying text. FONTNAME can be:
  - a14: Normal size font
  - a24: Large size font
  - ... (check available fonts with xlsfont.)
- -display displayname: the name of the X server to use. displayname can be:
  - hostname:D.S means screen S on display D of host hostname; the X server for this display is listening to TCP port 6000+D.
  - host/unix:D.S means screen S on display D of host host; the X server for this display is listening to UNIX domain socket /tmp/.X11-unix/XD (so it's only reachable from host).
  - :D.S is equivalent to host/unix:D.S, where host is the local hostname.

The default displayname for the X client program (application side) can be set by the DISPLAY environment variable. For example, prior to running an X client program, executing one of the following commands achieves this:

```
$ export DISPLAY=:0
        # The default, local machine using the first X screen
$ export DISPLAY=hostname.fulldomain.name:0.2
$ export DISPLAY=localhost:0
```

Its startup can be customized by ~/.xinitrc. For example:

```
xrdb -load $HOME/.Xresources
xsetroot -solid gray &
xclock -g 50x50-0+0 -bw 0 &
xload -g 50x50-50+0 -bw 0 &
xterm -g 80x24+0+0 &
xterm -g 80x24+0-0 &
twm
```

As described in 'Custom X sessions' на стр. 150, this overrides everything normal execution of Xsession does when started from startx. Use ~/.xsession instead and use this approach only as the last resort. See xsetroot(1x), xset(1x) and 'X resources' на стр. 154.

### 9.4.5 X sessions

An X session (X server + X client) can be started by:

- startx: wrapper script command for xinit to start an X server and client from a Linux character console. If ~/.xinitrc does not exist, /etc/X11/Xsession is executed through /etc /X11/xinit/xinitrc.
- xdm, gdm, kdm, or wdm: X display manager daemons to start the X server and client, and to control login via a GUI screen. /etc/X11/Xsession is directly executed.

The console can be made available as in ‘”Let me disable X on boot!“’ на стр. 111.

#### Custom X sessions

The default startup script /etc/X11/Xsession is effectively a combination of /etc/X11/Xsession.d /50xfree86-common_determine-startup and /etc/X11/Xsession.d/99xfree86-common_start.

Execution of /etc/X11/Xsession is somewhat affected by /etc/X11/Xsession.options and is essentially an execution of a program which was first found in the following order with the exec command:

1. ~/.xsession or ~/.Xsession, if it is defined.
2. /usr/bin/x-session-manager, if it is defined.
3. /usr/bin/x-window-manager, if it is defined.
4. /usr/bin/x-terminal-emulator, if it is defined.

The exact meaning of these commands is determined by the Debian alternative system described in ‘Alternative commands’ на стр. 99. For example:

```
# update-alternatives --config x-session-manager
... or
# update-alternatives --config x-window-manager
```

In order to make any X window manager a default while keeping GNOME and KDE session managers installed, replace /etc/X11/Xsession.d/50xfree86-common_determine-startup with the one attached in the second bug report at http://bugs.debian.org/168347 (I hope this will be included soon) and edit /etc/X11/Xsession.options as follows to disallow the X session manager:

```
# /etc/X11/Xsession.options
#
# configuration options for /etc/X11/Xsession
# See Xsession.options(5) for an explanation of the available options.
# Default enabled
allow-failsafe
allow-user-resources
allow-user-xsession
use-ssh-agent
# Default disabled (enable them by uncommenting)
do-not-use-x-session-manager
#do-not-use-x-window-manager
```

Without the above mentioned modification to the system, gnome-session and kdebase are the packages containing these X session managers. Removing them allows X window manager to be a default. (Yack, any better idea?)

On a system where /etc/X11/Xsession.options contains a line allow-user-xsession without preceding characters, any user who defines ~/.xsession or ~/.Xsession will be able to customize the action of /etc/X11/Xsession.

The last command in the ~/.xsession file should use form of exec some-window/session-manager to start your favorite X window/session manager.

A good example of an ~/.xsession script is given at /usr/share/doc/xfree86-common/examples /xsession.gz.

I use this to set the window manager, screen access, and language support for each user account. See 'Starting an X session for a user' на стр. 151, 'Getting root in X' на стр. 155, and 'Example for a multilingual X window system' на стр. 171.

If you wish to have several X client programs started automatically, see 'X clients' на стр. 148 examples and invoke them from ~/.xsession instead of ~/.xinitrc.

User-specific additional X resources can be stored in ~/.Xresources. See 'X resources' на стр. 154.

User-customized keymaps and pointer button mappings in X can also be specified in the user's start up script. See 'Keymaps and pointer button mappings in X' на стр. 155.

### Starting an X session for a user

Following the principle described at 'Custom X sessions' на стр. 150, a user-specific X session/window manager can be activated by installing the package indicated and setting the contents at the end of ~/.xsession file as follows. (I like blackbox/fluxbox for its simple style and fast speed.):

- default X session manager
  - See 'Alternative commands' на стр. 99
  - exec /usr/bin/x-session-manager

- default X window manager
  - See 'Alternative commands' на стр. 99
  - exec /usr/bin/x-window-manager

- GNOME session manager (loaded)
  - Install package: gnome-session
  - exec /usr/bin/gnome-session

- KDE session manager (loaded)
  - Install package: kdebase (or kdebase3 for KDE3)
  - exec /usr/bin/kde2

- Blackbox window manager (lightweight, slick)
  - Install package: blackbox
  - exec /usr/bin/blackbox
- Fluxbox window manager (lightweight, new blackbox)
  - Install package: fluxbox
  - exec /usr/bin/fluxbox
- Xfce window manager (Mac OS-X, SUN CDE–like)
  - Install package: xfce
  - exec /usr/bin/xfwm
- IceWM window manager (lightweight, GNOME alternative)
  - Install package: icewm
  - exec /usr/bin/X11/icewm
- FVWM2 virtual window manager (lightweight, Win95-like)
  - Install package: fvwm
  - exec /usr/bin/fvwm2
- Windowmaker window manager (somewhat NexT-like)
  - Install package: wmaker
  - exec /usr/bin/wmaker
- Enlightenment window manager (loaded)
  - Install package: enlightenment
  - exec /usr/bin/enlightenment

See Window Managers for X (http://www.xwinman.org).

Setting up KDE and GNOME

In order to setup full KDE or GNOME environment, the following metapackages are useful:

- KDE: install the kde package
- GNOME: install the gnome package

Installing these packages with tools which handle recommends, such as dselect and aptitude, provides you with richer choices of software than just installing these with apt-get.

If you want console login, be sure to disable X display managers, such as kdm, gdm, and wdm, which may be pulled in by the dependencies, as described in ‘"Let me disable X on boot!"’ на стр. 111.

If you want to have GNOME as the system default over KDE, make sure to configure x-session-manager as in ‘Alternative commands’ на стр. 99.

### 9.4.6 Using X over TCP/IP

Because a remote TCP/IP socket connection without encryption is prone to an eavesdropping attack, the default setting for X in recent Debian versions disables the TCP/IP socket. Consider using ssh for a remote X connection (see 'Connecting to a remote X server – ssh' на стр. 153).

The method described here is not encouraged unless one is in a very secure environment behind a good firewall system with only trusted users present. Use the following command to verify your current X server setting for the TCP/IP socket:

```
# find /etc/X11 -type f -print0 | xargs -0 grep nolisten
/etc/X11/xinit/xserverrc:exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Remove -nolisten to restore TCP/IP listening on the X server.

### 9.4.7 Connecting to a remote X server – xhost

xhost allows access based on hostnames. This is very insecure. The following will disable host checking and allow connections from anywhere if a TCP/IP socket connection is allowed (see 'Using X over TCP/IP' на стр. 153):

```
$ xhost +
```

You can re-enable host checking with:

```
$ xhost -
```

xhost does not distinguish between different users on the remote host. Also, hostnames (addresses actually) can be spoofed.

This method must be avoided even with more restrictive host criteria if you're on an untrusted network (for instance with dial-up PPP access to the Internet). See xhost(1x).

### 9.4.8 Connecting to a remote X server – ssh

The use of ssh enables a secure connection from a local X server to a remote application server.

- Set X11Forwarding and AllowTcpForwarding entries to yes in /etc/ssh/sshd_config of the remote host, if you want to avoid corresponding command-line options.
- Start the X server on the local host.
- Open an xterm in the local host.

- Run ssh to establish a connection with the remote site.

```
localname @ localhost $ ssh -q -X -l loginname remotehost.domain
Password:
.....
```

- Run X application commands on the remote site.

```
loginname @ remotehost $ gimp &
```

This method allows the display of the remote X client output as if it were locally connected through a local UNIX domain socket.

### 9.4.9 The X terminal emulator – xterm

Learn everything about xterm at http://dickey.his.com/xterm/xterm.faq.html.

### 9.4.10 X resources

Many older X programs, such as xterm, use the X resource database to configure their appearance. The file ~/.Xresources is used to store user resource specifications. This file is automatically merged into the default X resources upon login. The system-wide defaults of X resources are stored in /etc/X11/Xresources/* and application defaults of them are stored in /etc/X11/app-defaults/*. Use these settings as the starting points.

Here are some helpful settings to add to your ~/.Xresources file:

```
! Set the font to a more readable 9x15
XTerm*font: 9x15

! Display a scrollbar
XTerm*scrollBar: true

! Set the size of the buffer to 1000 lines
XTerm*saveLines: 1000

! Large kterm screen
KTerm*VT100*fontList: -*-fixed-medium-r-normal--24-*,\
 -*-gothic-medium-r-normal--24-*,\
 -*-mincho-medium-r-normal--24-*
```

To make these settings take effect immediately, merge them into the database using the command:

```
xrdb -merge ~/.Xresources
```

See xrdb(1x).

### 9.4.11 Keymaps and pointer button mappings in X

The xmodmap program is used to edit and display the keyboard modifier map and keymap table that are used by client applications to convert event keycodes into keysyms in X.

```
$ xmodmap -pm
 ... display the current modifier map
$ xmodmap -pk | pager
 ... display the current keymap table
$ xmodmap -e "pointer = 3 2 1" # set mouse for the left hand.
$ xmodmap ~/.xmodmaprc # set keyboad as in ~/.xmodmaprc
```

It is usually run from the user's session startup script, ~/.xsession.

To get the keycode, run xev in X and press keys. To get the meaning of keysym, look into the MACRO definition in /usr/include/X11/keysymdef.h file. All the #define statements in this file are named as XK_ prepended to the keysym names.

See xmodmap(1x).

### 9.4.12 Getting root in X

If a GUI program needs to be run with root privilege, use the following procedures to display program output on a user's X server. Never attempt to start an X server directly from the root account in order to avoid possible security risks.

Start the X server as a normal user and open an xterm console. Then:

```
$ XAUTHORITY=$HOME/.Xauthority
$ export XAUTHORITY
$ su root
Password:*****
# printtool &
```

When using this trick to su to a non-root user, make sure ~/.Xauthority is group readable by this non-root user.

To automate this command sequence, create a file ~/.xsession from the user's account, containing the following lines:

```
# This makes X work when I su to the root account.
if [ -z "$XAUTHORITY" ]; then
        XAUTHORITY=$HOME/.Xauthority
        export XAUTHORITY
fi
unset XSTARTUP
```

```
# If a particular window/session manager is desired, uncomment
# the following and edit it to fit your needs.
#XSTARTUP=/usr/bin/blackbox
# This starts x-window/session-manager program
if [ -z "$XSTARTUP" ]; then
  if [ -x /usr/bin/x-session-manager ]; then
    XSTARTUP=x-session-manager
  elif [ -x /usr/bin/x-window-manager ]; then
    XSTARTUP=x-window-manager
  elif [ -x /usr/bin/x-terminal-emulator ]; then
    XSTARTUP=x-terminal-emulator
  fi
fi
# execute auto selected X window/session manager
exec $XSTARTUP
```

Then run su (not su -) in an xterm window of the user. Now GUI programs started from this xterm can display output on this user's X window while running with root privilege. This trick works as long as the default /etc/X11/Xsession is executed. If a user set up his customization using ~/.xinit or ~/.xsession, the above mentioned environment variable XAUTHORITY needs to be set similarly in those scripts.

Alternatively, sudo can be used to automate the command sequence:

```
$ sudo xterm
... or
$ sudo -H -s
```

Here /root/.bashrc should contain:

```
if [ $SUDO_USER ]; then
    sudo -H -u $SUDO_USER xauth extract - $DISPLAY | xauth merge -
fi
```

This works fine even with the home directory of the user on an NFS mount, because root does not read the .Xauthority file.

There are also several specialized packages for this purpose: kdesu, gksu, gksudo, gnome-sudo, and xsu. Some other methods can be used to achieve similar results: creating a symlink from /root/.Xauthority to the user's corresponding one; use of the script sux (http://fgouget.free.fr/sux/sux-readme.shtml); or putting "xauth merge ~USER_RUNNING_X/.Xauthority" in the root initialization script.

See more on the debian-devel mailing list (http://lists.debian.org/debian-devel/2002/debian-devel-200207/msg00259.html).

### 9.4.13 TrueType fonts in X

The standard xfs in XFree86-4 works fine with TrueType fonts. You have to install a third-party font server such as xfs-xtt, if you are using XFree86-3.

You just need to make sure that whatever applications you want to use the TrueType fonts are linked against libXft or libfreetype (you probably don't even have to worry about this if you're using pre-compiled .debs).

First set up font support infrastructure:

- Install x-ttcidfont-conf and defoma packages. This automates generation of the fonts.scale and fonts.dir files.

  ```
  # apt-get install x-ttcidfont-conf
  ```

- Edit /etc/X11/XF86Config-4 in the Section "Files" as:

  ```
  Section "Files"
      FontPath "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
      FontPath "/usr/share/fonts/truetype"
      FontPath "/usr/lib/X11/fonts/CID"
      FontPath "/usr/lib/X11/fonts/Speedo"
      FontPath "/usr/lib/X11/fonts/misc"
      FontPath "/usr/lib/X11/fonts/cyrillic"
      FontPath "/usr/lib/X11/fonts/100dpi:unscaled"
      FontPath "/usr/lib/X11/fonts/75dpi:unscaled"
      FontPath "/usr/lib/X11/fonts/Type1"
  EndSection
  ```

  The first line will setup XFree86 to use any TrueType fonts you install from Debian packages. Type1 font entry is moved down since XFree86 does a rather poor job of rendering Type1 fonts. The :unscaled trick for bitmap fonts should not be needed for new XF4 anymore but I included it here just be sure. In order to preserve manual changes of /etc /X11/XF86Config-4 follow instructions in 'Configuring the X server manually' на стр. 147.

Then install DSFG font packages:

- Western TrueType fonts:
  - ttf-bitstream-vera: A set of high-quality TrueType fonts created by Bitstream, Inc. [2]
  - ttf-freefont: A set of free high-quality TrueType fonts covering the UCS character set.
  - ttf-thryomanes: A TrueType Unicode font covering Latin, Greek, Cyrillic and IPA.
- Asian fonts:
  - tfm-arphic-bsmi00lp: Chinese Arphic "AR PL Mingti2L Big5" TrueType font TeX font metric data
  - tfm-arphic-bkai00mp: Chinese Arphic "AR PL KaitiM Big5" TrueType font TeX font metric data
  - tfm-arphic-gbsn00lp: Chinese Arphic "AR PL SungtiL GB" TrueType font TeX font metric data
  - tfm-arphic-gkai00mp: Chinese Arphic "AR PL KaitiM GB" TrueType font TeX font metric data
  - ttf-baekmuk: Korean Baekmuk series TrueType fonts

[2] Though this is not available in Woody, you can install this from Sarge.

  - hbf-jfs56: Chinese Jianti Fangsong 56x56 bitmap font (GB2312) for CJK
  - hbf-cns40-b5: Chinese Fanti Song 40x40 bitmap font (Big5) for CJK
  - hbf-kanji48: Japanese Kanji 48x48 bitmap font (JIS X-0208) for CJK

Since Free fonts are sometimes limited, installing or sharing some commercial TrueType fonts is an option for a Debian users. In order to make this process easy for the user, some convenience packages have been created:

- ttf-commercial
- msttcorefonts (>1.1.0) [3]

You'll have a really good selection of TruType fonts at the expense of contaminating your Free system with non-Free fonts.

All these font packages in Debian should work without any efforts and appear available to all X programs that use the regular "core" font system. This includes things like Xterm, Emacs and most other non-KDE and non-GNOME applications.

Now, run xfontsel and select any TrueType fonts in the fndry menu, you should be able to see many ungrayed out entries in the "fmly" menu.

For KDE2.2 and GNOME1.4 (with libgdkxft0, which is a hack to get GTK 1.2 to do anti-aliased font rendering), you need to setup Xft1, as well. Xft1 is highly deprecated, and is basically only used by GNOME1.4 and KDE2.2. Edit /etc/X11/XftConfig and add a line like

```
dir "/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType"
```

before the other dir lines. [4]

For GNOME2 and KDE3 (post Sarge release), you need to setup fontconfig which Xft2 uses to find fonts. [5] You shouldn't need to install anything extra for this, since all the packages using fontconfig will Depend on it (indirectly) already.

First, look in /etc/fonts/fonts.conf. There should be a line like the one below. If not, open up /etc/fonts/local.conf and add this

```
<dir>/var/lib/defoma/x-ttcidfont-conf.d/dirs/TrueType</dir>
```

just after the <fontconfig> line.

Fontconfig should pick these up immediately, and "fc-list" should list your new fonts. Another neat feature of fontconfig is that you can just drop fonts in ~/.fonts/ and all your fontconfigified programs will have access to them immediately.

If you manually install a new set of TrueType fonts while in X without using Debian package, run

[3] The package in Woody does not work as of 8/2002 due to a change in Microsoft's website. Use Sarge version even in Woody instead.

[4] I don't have any xft1 stuff on my machine anymore, so I'm not sure if you need to restart X or not before this change will take effect. I seem to remember that "xftcache" would update the Xft1 cache, but it'd be good if someone could confirm that for me.

[5] Fontconfig does not exist in Woody.

```
# xset fp rehash
```

to get XFree86 to look at the contents of that directory again and to pickup new ones.

### 9.4.14 Web browsers in X

There are a few web browser packages with graphical display capabilities as of the Woody release:
- mozilla The Mozilla browser (new)
- galeon Mozilla-based browser with a Gnome UI (new)
- konqueror KDE browser
- dillo GTK browser
- amaya-gtk W3C reference browser
- amaya-lesstif W3C reference browser
- netscape-... (many, old)
- communicator-... (many, old)
- ...

The version of mozilla must match the version that galeon requires. Although they differ in UI, these two programs share the Gecko HTML rendering engine.

Plug-ins for browsers such as mozilla and galeon can be enabled by installing "*.so" manually in the plug-in directory and restarting the browsers.

Plug-in resources:
- Java plug-in: install binary "J2SE" from http://java.sun.com.
- Flash plug-in: install binary "Macromedia Flash Player 5" from http://www.macromedia.com/software/flashplayer/.
- freewrl: VRML browser and Netscape plug-in
- ...

## 9.5 SSH

SSH (Secure SHell) is the secure way to connect over the Internet. A free version of SSH called OpenSSH is available as the ssh package in Debian.

### 9.5.1 Basics of SSH

First install the OpenSSH server and client.

```
# apt-get update && apt-get install ssh
```

/etc/ssh/sshd_not_to_be_run must not be present if one wishes to run the OpenSSH server.

SSH has two authentication protocols:

- SSH protocol version 1:
  - Potato version only supports this protocol.
  - available authentication methods:
    * RSAAuthentication: RSA identity key based user authentication
    * RhostsAuthentication: .rhosts based host authentication (insecure, disabled)
    * RhostsRSAAuthentication: .rhosts authentication combined with RSA host key (disabled)
    * ChallengeResponseAuthentication: RSA challenge-response authentication
    * PasswordAuthentication: password based authentication

- SSH protocol version 2:
  - post-Woody versions use this as the primary protocol.
  - available authentication methods:
    * PubkeyAuthentication: public key based user authentication
    * HostbasedAuthentication: .rhosts or /etc/hosts.equiv authentication combined with public key client host authentication (disabled)
    * ChallengeResponseAuthentication: challenge-response authentication
    * PasswordAuthentication: password based authentication

Be careful about these differences if you are migrating to Woody or using a non-Debian system.

See /usr/share/doc/ssh/README.Debian.gz, ssh(1), sshd(8), ssh-agent(1), and ssh-keygen(1) for details.

Following are the key configuration files:

- /etc/ssh/ssh_config: SSH client defaults. See ssh(1). Notable entries are:
  - Host: Restricts the following declarations (up to the next Host keyword) to be only for those hosts that match one of the patterns given after the keyword.
  - Protocol: Specifies the SSH protocol versions. The default is "2,1".
  - PreferredAuthentications: Specifies the SSH2 client authentication method. The default is "hostbased,publickey,keyboard-interactive,password".
  - PasswordAuthentication: If you want to log in with a password, you have to make sure this is not set no.
  - ForwardX11: The default is disabled. This can be overridden by the command-line option "-X".

- /etc/ssh/sshd_config: SSH server defaults. See sshd(8). Notable entries are:
  - ListenAddress: Specifies the local addresses sshd should listen on. Multiple options are permitted.
  - AllowTcpForwarding: The default is disabled.
  - X11Forwarding: The default is disabled.

- $HOME/.ssh/authorized_keys: the lists of the default public keys that clients use to connect to this account on this host. See ssh-keygen(1).

- $HOME/.ssh/identity: See ssh-add(1) and ssh-agent(1).

The following will start an ssh connection from a client.

```
$ ssh username@hostname.domain.ext
$ ssh -1 username@hostname.domain.ext # Force SSH version 1
$ ssh -1 -o RSAAuthentication=no -l username foo.host
    # force password on SSH1
$ ssh -o PreferredAuthentications=password -l username foo.host
    # force password on SSH2
```

For the user, ssh functions as a smarter and more secure telnet (will not bomb with ^]).

### 9.5.2 Port forwarding for SMTP/POP3 tunneling

To establish a pipe to connect to port 25 of remote-server from port 4025 of localhost, and to port 110 of remote-server from port 4110 of localhost through ssh, execute on the local machine:

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 \
         username@remote-server
```

This is a secure way to make connections to SMTP/POP3 servers over the Internet. Set the AllowTcpForwarding entry to yes in /etc/ssh/sshd_config of the remote host.

### 9.5.3 Connecting with fewer passwords – RSA

One can avoid having to remember a password for each remote system by using RSAAuthentication (SSH1 protocol) or PubkeyAuthentication (SSH2 protocol).

On the remote system, set the respective entries, "RSAAuthentication yes" or "PubkeyAuthentication yes", in /etc/ssh/sshd_config.

Then generate authentication keys locally and install the public key on the remote system:

```
$ ssh-keygen          # RSAAuthentication: RSA1 key for SSH1
$ cat .ssh/identity.pub | ssh user1@remote \
       "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: RSA key for SSH2
$ cat .ssh/id_rsa.pub | ssh user1@remote \
       "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: DSA key for SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
       "cat - >>.ssh/authorized_keys"
```

One can change the passphrase later with "ssh-keygen -p". Make sure to verify settings by testing the connection. In case of any problem, use "ssh -v".

You can add options to the entries in authorized_keys to limit hosts and to run specific commands. See sshd(8) for details.

Note that SSH2 has HostbasedAuthentication. For this to work, you must adjust the settings of HostbasedAuthentication to yes in both /etc/ssh/sshd_config on the server machine and /etc/ssh/ssh_config or $HOME/.ssh/config on the client machine.

### 9.5.4 Dealing with alien SSH clients

There are a few free SSH clients available for non-Unix-like platforms.

Windows puTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/) (GPL)

Windows (cygwin) SSH in cygwin (http://www.cygwin.com/) (GPL)

Macintosh Classic macSSH (http://www.macssh.com/) (GPL) [Note that Mac OS X includes OpenSSH; use ssh in the Terminal application]

See also SourceForge.net, site documentation (http://www.sourceforge.net/docman/?group_id=1), "6. CVS Instructions".

### 9.5.5 Setting up ssh-agent

It is safer to protect your SSH authentication key with a passphrase. If it was not set, use ssh-keygen -p to set it.

Place your public key (e.g. ~/.ssh/id_rsa.pub) into ~/.ssh/authorized_keys on a remote host using a password-based connection to the remote host as described in 'Connecting with fewer passwords – RSA' на стр. 161.

```
$ ssh-agent bash # or run zsh/tcsh/pdksh program instead.
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/osamu/.ssh/id_rsa:
Identity added: /home/osamu/.ssh/id_rsa (/home/osamu/.ssh/id_rsa)
$ scp foo user@remote.host:foo
 ... no passphrase needed from here on :-)
$^D
 ... terminating ssh-agent session
```

For the X server, normal Debian startup scripts execute ssh-agent as parent process. So you only need to execute ssh-add once.

For more, read ssh-agent(1)and ssh-add(1).

### 9.5.6 Troubleshooting SSH

If you have problems, check the permissions of configuration files and run ssh with the "-v" option.

Use the "-P" option if you are root and have trouble with a firewall; this avoids the use of server ports 1–1023.

If ssh connections to a remote site suddenly stop working, it may be the result of tinkering by the sysadmin, most likely a change in host_key during system maintenance. After making sure this is the case and nobody is trying to fake the remote host by some clever hack, one can regain a connection by removing the host_key entry from $HOME/.ssh/known_hosts on the local machine.

## 9.6 Mail

Mail configuration divides into three categories:

- mail transfer agent (MTA): exim, postfix, sendmail, qmail, ssmtp, nullmailer, ...
- mail utilities: procmail, fetchmail, mailx, ...
- mail user agent (MUA): mutt, emacs+gnus, ...

### 9.6.1 Mail transport agents (MTAs)

For a full-featured MTA, use exim. References:
- exim-doc and exim-doc-html packages
- http://www.exim.org/

The only reasonable alternative MTA is postfix if you care about security. sendmail and qmail are available as Debian packages but are not recommended.

If you do not need the relay capability of an MTA as in the case of a satellite system such as a laptop PC, you may consider using one of these lightweight packages:
- ssmtp: needs an SMTP connection and is alias-capable, or
- nullmailer: can spool but is not alias-capable.

At this moment, I find exim to be more suitable even for my personal workstation machine, which is a laptop PC.

You may need to remove exim for the installation of these conflicting packages:

```
# dpkg -P --force-depends exim
# apt-get install nullmailer         # or ssmtp
```

### Smarthost

If you are running exim on a host which is connected through the consumer grade services, please make sure to send outgoing mail through a smarthost offered by your ISP or some others. [6] There are few good reasons:

- to ensure SMTP retries since your ISP's smarthost usually have more reliably connection.
- to avoid sending mail directly from a dynamic IP address which will likely be blocked by dial-up spam lists.
- to save your local bandwidth to send mails with multiple recipients.

The only conceivable exceptions are:

- the emergency cure for your ISP's SMTP service trouble.
- an experiment for the educational purpose.
- your host being a professionally hosted server.

### Basic configuration of Exim

In order to use exim as your MTA, configure the following:

```
/etc/exim/exim.conf      "eximconfig" to create and edit
/etc/inetd.conf          comment out smtp to run exim as daemon
/etc/email-addresses     Add spoofed source address lists
check filters using exim -brw, -bf, -bF, -bV, ... etc.
```

### Setting up a catchall for nonexistent email addresses under Exim

In /etc/exim/exim.conf (Woody or later), in the DIRECTORS part, at the end (after the localuser: director) add a catch-all director that matches all addresses that the previous directors couldn't resolve (per Miquel van Smoorenburg):

```
catchall:
  driver = smartuser
    new_address = webmaster@mydomain.com
```

If one wants to have more a detailed recipe for each virtual domain, etc., add the following at the end of /etc/exim/exim.conf (per me, not well tested):

```
*@yourdomain.com ${lookup{$1}lsearch*{/etc/email-addresses} \
        {$value}fail} T
```

Then have an "*" entry in /etc/email-addresses.

[6] You must follow this rule for any hosts on dial-up, DSL, cable services or LAN through some broadband router. Even if your home host has a fixed IP from your ISP, it is still a good idea to follow this rule. Most workstations and home servers fall into this category.

Configuring selective address rewriting for outgoing mail under Exim

Selective address rewrite for outgoing mail to produce proper "From:" headers can be done using exim by configuring near the end of /etc/exim/exim.conf:

```
*@host1.something.dyndns.org \
  "${if eq {${lookup{$1}lsearch{/etc/passwd}{1}{0}}} {1}  \
   {$0}{$1@somethig.dyndns.org}}"  frFs
```

This rewrites all addresses matching *@host1.something.dyndns.org.

1 It searches through /etc/password to see if the local part ($1) is a local user or not.

2 If it is a local user, it rewrites the address to the same thing it was in the first place ($0).

3 If it is not a local user, it rewrites the domain part.

Configuring SMTP authentication under Exim

Some SMTP services such as yahoo.com require SMTP auth. Configure /etc/exim/exim.conf as follows:

```
remote_smtp:
  driver = smtp
  authenticate_hosts = smtp.mail.yahoo.com
...

smarthost:
  driver = domainlist
  transport = remote_smtp
  route_list = "* smtp.mail.yahoo.com bydns_a"
...

plain:
  driver = plaintext
  public_name = PLAIN
  client_send = "^cmatheson3^this_is_my_password"
```

Do not forget double quotes in the last line.

### 9.6.2 Fetching mail – Fetchmail

fetchmail is run in daemon mode to fetch mail from a POP3 account with an ISP into the local mail system. Configure:

```
/etc/init.d/fetchmail
/etc/rc?.d/???fetchmail run update-rc.d fetchmail default priority 30
/etc/fetchmailrc        configuration file (chown 600, owned by fetchmail)
```

Information on how to start fetchmail as a daemon from the init.d script for Potato is confusing (Woody fixed this). See the sample /etc/init.d/fetchmail and /etc/fetchmailrc files in the example scripts (http://www.debian.org/doc/manuals/debian-reference/examples/).

If your email headers are contaminated by ^M due to your ISP's mailer, add "stripcr" to your options in $HOME/.fetchmailrc:

```
options fetchall no keep stripcr
```

### 9.6.3 Processing mail – Procmail

procmail is a local mail delivery and filter program. One needs to create $HOME/.procmailrc for each account that uses it. Example: _procmailrc (http://www.debian.org/doc/manuals/debian-reference/examples/)

### 9.6.4 Reading mail – Mutt

Use mutt as the mail user agent (MUA) in combination with vim. Customize with ~/.muttrc; for example:

```
# use visual mode and "gq" to reformat quotes
set editor="vim -c 'set tw=72 et ft=mail'"
#
# header weeding taken from the manual (Sven's Draconian header weeding)
#
ignore *
unignore from: date subject to cc
unignore user-agent x-mailer
hdr_order from subject to cc date user-agent x-mailer
auto_view application/msword
....
```

Add the following to /etc/mailcap or $HOME/.mailcap to display HTML mail and MS Word attachments inline:

```
text/html; lynx -force_html %s; needsterminal;
application/msword; /usr/bin/antiword '%s'; copiousoutput;
description="Microsoft Word Text"; nametemplate=%s.doc
```

## 9.7 Localization

Debian is internationalized, offering support for a growing number of languages and local usage conventions. The next subsection lists some of the forms of diversity that Debian currently supports, and the following subsections discuss localization, the process of customizing your working environment to allow current input and output of your chosen language(s) and conventions for dates, numeric and monetary formats, and other aspects of a system that differ according to your region.

### 9.7.1 Basics of localization

There are several aspects to customizing for localization and national language support.

#### Localizing the keyboard

Debian is distributed with keymaps for nearly two dozen keyboards. In Woody, reconfigure the keyboard by:

- dpkg-reconfigure —priority=low console-data # console
- dpkg-reconfigure —priority=low xserver-xfree86 # XF4
- dpkg-reconfigure —priority=low xserver-common-v3 # XF3

#### Localizing data files

The vast majority of Debian software packages support data handling of non-US-ASCII characters through the LC_CTYPE environment variable offered by the locale technology in glibc.

- 8-bit clean: practically all programs
- other Latin character sets (e.g. ISO-8859-1 or ISO-8859-2): the majority of programs
- multibyte languages such as Chinese, Japanese, or Korean: many new applications

#### Localizing the display

X can display any coding, including UTF-8, and supports all fonts. The list includes not only all the 8-bit fonts but also 16-bit fonts such as Chinese, Japanese, or Korean. Multibyte character input method is supported by the 'Alternative X input methods' на стр. 174 mechanism. See 'Example for a multilingual X window system' на стр. 171 and 'UTF-8 support for the X terminal emulator' на стр. 175.

Japanese EUC code display is also available in a (S)VGA graphics console through the kon2 package. There is an alternative new Japanese display, jfbterm, which uses a frame-buffer console, too. In these console environments, the Japanese input method must be supplied by the application. Use egg package for Emacs and use japanized jvim package for a Vim environment.

Installation of non Unicode fonts to X will help in displaying documents with any encoding in X. So do not worry too much about encoding of fonts.

Localizing messages and documentation

Translations exist for many of the text messages and documents that are displayed in the Debian system, such as error messages, standard program output, menus, and manual pages. Currently, support for manual pages in German, Spanish, Finnish, French, Hungarian, Italian, Japanese, Korean, Polish, Portuguese, Chinese, and Russian is provided through the manpages-LANG packages (where LANG is a comma-separated list of two-letter ISO country codes. Use apt-cache search manpages-|less to get a list of available Unix manual pages.)

To access an NLS manual page, the user must set the environment variable LC_MESSAGES to the appropriate string. For example, in the case of the Italian-language manual pages, LC_MESSAGES needs to be set to it. The man program will then search for Italian manual pages under /usr/share/man/it/.

### 9.7.2 Locales

Debian supports locale technology. Locale is a mechanism that allows programs to provide suitable output and functionality according to local conventions such as character set, format for date and time, currency symbol, and so on. It uses environment variables to determine the appropriate behavior. For example, assuming you have both the American English and German locales installed on your system, the error messages of many programs can be multilingual:

```
$ LANG="en_US" cat foo
cat: foo: No such file or directory
$ LANG="de_DE" cat foo
cat: foo: Datei oder Verzeichnis nicht gefunden
```

Glibc offers support for this functionality to programs as a library. See locale(7).

### 9.7.3 Introduction to locales

Full locale description consists of 3 parts: xx_YY.ZZZZ.

- xx: ISO 639 language codes (lower case)
- YY: ISO 3166 country codes (upper case)
- ZZZZ: codeset, i.e., character set or encoding identifier.

For language codes and country codes, see pertinent description in the info gettext.

Please note this codeset part may be normalized internally to achieve cross platform compatibility by removing all - and by converting all characters into lower case. Typical codesets are:

- UTF-8: Unicode for all regions, mostly in 1-3 Octets (new de facto standard)
- ISO-8859-1: western Europe (de facto old standard)
- ISO-8859-2: eastern Europe (Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Serbian, Slovak, Slovenian)
- ISO-8859-3: Maltese

- ISO-8859-5: Macedonian, Serbian
- ISO-8859-6: Arabic
- ISO-8859-7: Greek
- ISO-8859-8: Hebrew
- ISO-8859-9: Turkish
- ISO-8859-11: Thai (=TIS-620)
- ISO-8859-13: Latvian, Lithuanian, Maori
- ISO-8859-14: Welsh
- ISO-8859-15: western Europe with euro
- KOI8-R: Russian
- KOI8-U: Ukrainian
- CP1250: Czech, Hungarian, Polish (MS Windows origin)
- CP1251: Bulgarian, Byelorussian (MS Windows origin)
- eucJP: Unix style Japanese (=ujis)
- eucKR: Unix style Korean
- GB2312: Unix style Simplified Chinese (=GB, =eucCN) for zh_CN
- Big5: Traditional Chinese for zh_TW
- sjis: Microsoft style Japanese (Shift-JIS)

As for the meaning of basic encoding system jargons:

- ASCII: 7 bits (0-0x7f)
- ISO-8859-?: 8 bits (0-0xff)
- ISO-10646-1: Universal Character Set (UCS) (31 bits, 0-0x7fffffff)
- UCS-2: First 16 bit of UCS as straight 2 Octets (Unicode: 0-0xffff)
- UCS-4: UCS as straight 4 Octets (UCS: 0-0x7fffffff)
- UTF-8: UCS encoded in 1-6 Octets (mostly in 3 Octets)
- ISO-2022: 7 bits (0-0xff) with the escape sequence. ISO-2022-JP is the most popular encoding for the Japanese e-mail.
- EUC: 8 bits + 16 bits combination (0-0xff), Unix style
- Shift-JIS: 8 bits + 16 bits combination (0-0xff), Microsoft style.

ISO-8859-?, EUC, ISO-10646-1, UCS-2, UCS-4, and UTF-8 share the same code with ASCII for the 7 bit characters. EUC or Shift-JIS uses high-bit characters (0x80-0xff) to indicate that part of encoding is 16 bit. UTF-8 also uses high-bit characters (0x80-0xff) to indicate non 7 bit character sequence bytes and this is the most sane encoding system to handle non-ASCII characters.

Please note the byte order difference of Unicode implementation:

- Standard UCS-2, UCS-4: big endian
- Microsoft UCS-2, UCS-4: little endian for ix86 (machine-dependent)

For more see Introduction to i18n (http://www.debian.org/doc/manuals/intro-i18n/).

### 9.7.4 Activating locale support

Debian does not come with all available locales pre-compiled. Check /usr/lib/locale to see which locales (besides the default "C") are compiled for your system. If the one you need is not present, you have two options:

- Edit /etc/locale.gen to add the desired locale, then run locale-gen as root to compile it. See locale-gen(8) and the manpages listed in its "SEE ALSO" section.
- Run dpkg-reconfigure locales to reconfigure the locales package. Or if it is not already installed, installing locales will invoke the debconf interface to let you choose needed locales and compile the database.

### 9.7.5 Activating a particular locale

The following environment variables are evaluated in this order to provide particular locale values to programs:

1 LANGUAGE: This environment variable consists of a colon-separated list of locale names in order of priority. Used only if the POSIX locale is set to a value other than "C" [in Woody; the Potato version always has priority over the POSIX locale]. (GNU extension)

2 LC_ALL: If this is non-null, the value is used for all locale categories. (POSIX.1) Usually "" (null).

3 LC_*: If this is non-null, the value is used for the corresponding category (POSIX.1). Usually "C".

LC_* variables are:
- LC_CTYPE: Character classification and case conversion.
- LC_COLLATE: Collation order.
- LC_TIME: Date and time formats.
- LC_NUMERIC: Non-monetary numeric formats.
- LC_MONETARY: Monetary formats.
- LC_MESSAGES: Formats of informative and diagnostic messages and interactive responses.
- LC_PAPER: Paper size.
- LC_NAME: Name formats.
- LC_ADDRESS: Address formats and location information.
- LC_TELEPHONE: Telephone number formats.
- LC_MEASUREMENT: Measurement units (Metric or Other).
- LC_IDENTIFICATION: Metadata about the locale information.

4 LANG: If this is non-null and LC_ALL is undefined, the value is used for all LC_* locale categories with undefined values. (POSIX.1) Usually "C".

Note that some applications (e.g., Netscape 4) ignore LC_* settings.

The locale program can display active locale settings and available locales; see locale(1). (NOTE: locale -a lists all the locales that your system knows about; this does not mean that all of them are compiled! See 'Activating locale support' на стр. 169.)

### 9.7.6 ISO 8601 date format locale

The locale support for the international date standard of yyyy-mm-dd (ISO 8601 date format) is provided by the locale called en_DK, "English in Denmark", which is a bit of joke :-) This seems to work only in a console screen for ls.

### 9.7.7 Example for the US (ISO-8859-1)

Add the following lines to ~/.bash_profile:

```
LC_CTYPE=en_US.ISO-8859-1
export LC_CTYPE
```

### 9.7.8 Example for France with Euro sign (ISO-8859-15)

Add the following lines to ~/.bash_profile:

```
LANG=fr_FR@euro
export LANG
LC_CTYPE=fr_FR@euro
export LC_CTYPE
```

Configure the keyboard for French "AZERTY" as described in 'Localizing the keyboard' на стр. 167; add French manual pages by installing manpages-fr. The Right-Alt key in the US is called Alt-Gr in Europe. Pressing this together with other keys creates numerous accented and special characters. For example, Alt-Gr+E creates a Euro sign.

Most western European languages can be configured similarly.

See Debian Euro HOWTO (http://www.debian.org/doc/manuals/debian-euro-support/) for adding support for the new Euro currency and Utiliser et configurer Debian pour le français (http://www.debian.org/doc/manuals/fr/debian-fr-howto/) for more details in French.

### 9.7.9 Example for a multilingual X window system

Let us set up a multilingual X window system which simultaneously supports Japanese, English, German and French with EUC, UTF-8 and ISO-8859-1 encodings in different consoles.

I will show you a customization using the Debian menu system. See the details of Debian menu system in /usr/share/doc/menu/html/index.html. I also create a shortcut to the mozilla web browser in this example. [7]

[7] In this example, 2 bug work arounds are deployed for the version of blackbox in 2003. I use sh -c in command. Also ~/.menu/* entry is not used but root requiring /etc/menu/* was used instead.

- add locale support for the Japanese ja_JP.eucJP locale and other required locales using the method described at 'Localization' на стр. 167. (for all)
- install Kana-to-Kanji conversion system and dictionary (for Japanese):
  - canna – Local server ("free-beer" license), or
  - freewnn-jserver – Network-extensible server (Public Domain)
- install Japanese input method system (for Japanese):
  - kinput2-canna – for X, or
  - kinput2-canna-wnn – for X, and
  - egg – directly works with Emacsen even in console (optional)
- Install compatible terminals (for all):
  - xterm – X (for ISO-8859-1 and UTF-8),
  - kterm – X (for Japanese EUC), and
  - mlterm – X (multilingual).
- add all the required font packages. (for all)
- create the ~/.xsession that sets the user-specific X environment as described in 'Custom X sessions' на стр. 150 (for all):

```
#!/bin/sh
# This makes X work when I su to root.
if [ -z "$XAUTHORITY" ]; then
     XAUTHORITY=$HOME/.Xauthority
     export XAUTHORITY
fi

# Set specific environment through debian menu system.
# Reset locale
unset LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
unset LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
unset LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
# set locale default in X
LANG=C
# export locale
export LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
export LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
export LC_IDENTIFICATION LC_ALL LANG LANGUAGE PAGER
###
# activate input method for Japanese with kinput2
kinput2 &
XMODIFIERS="@im=kinput2"
export XMODIFIERS
# How about blackbox window manager (lightweight)
exec blackbox
```

```
#exec xfwm
#exec wmaker
```

- set locale in ~/.bash_profile for Linux consoles (for all).
- remove locale settings from ~/.bashrc, if existed (for all).
- create few files in /etc/menu/ (for all).
  - /etc/menu/xterm-local: (add new entries to menu) [8]

```
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (en_US.ISO-8859-1)"\
 title="XTerm (en_US.ISO-8859-1)"\
 command="sh -c 'LC_ALL=en_US.ISO-8859-1 xterm'"
?package(xterm):\
 needs=x11\
 section=XShells\
 longtitle="XTerm: terminal emulator (de_DE.ISO-8859-1)"\
 title="XTerm (de_DE.ISO-8859-1)"\
 command="sh -c 'LC_ALL=de_DE.ISO-8859-1 xterm -T xterm-de'"
?package(xterm):\
 needs=x11\
 section=XShells\
longtitle="XTerm: terminal emulator for X with Unicode support (Japanese)"\
 title="UXTerm (ja_JP.UTF-8)"\
 command="sh -c 'LC_ALL=ja_JP.UTF-8 uxterm'"
```

  - /etc/menu/kterm: (override the system default) [9]

```
?package(kterm):\
 needs="x11"\
 section="XShells"\
command="sh -c 'LC_ALL=ja_JP.eucJP PAGER=w3m /usr/X11R6/bin/kterm -
xim'" \
 title="Kanji Terminal"
?package(kterm):\
 needs="x11"\
 section="XShells"\
 command="sh -c 'LANG=ja_JP.eucJP \
                                   LC_MESSAGES=en_US.ISO-8859-
1 PAGER=w3m /usr/X11R6/bin/kterm -xim'" \
 title="Kanji Terminal (bilingal)"
```

  - /etc/menu/mozilla-local: (add a new shortcut) [10]

[8] Use a file name which does not overwrap with any package names.

[9] Use a file name which overwraps with the package name.

[10] The slash in section="/" enables entry to the initial menu, and the leading space in title=" Mozilla Navigator" enables entry to the top of the list.

```
?package(mozilla-browser):needs="x11" section="/" \
  title=" Mozilla Navigator" command="mozilla-1.5" hints="Web browsers" \
    icon=/usr/share/pixmaps/mozilla.xpm
```

  – run update-menus from the root account.

- add the following lines to ~/.muttrc (for Japanese):

```
# UTF-8 support is not popular in popular Japanese EMACS environment
# 7-bit encoding of iso-2022-jp is easier for everyone.
# default encoding order = us-ascii --> iso-8859-1 --> iso-2022-jp
set send_charset="us-ascii:iso-8859-1:iso-2022-jp"
set allow_8bit=no
```

- activate XIM kinput2 for X applications (for Japanese):
  – add *inputMethod: kinput2 and KTerm*VT100*OpenIm: true to your X resources file, ~/.Xresources (it looks like Debian takes care of this automatically somehow).
  – Some applications (such as mlterm) also allow you to set up *inputMethod: and other information dynamically at runtime (press Ctrl-MouseButton-3 in mlterm).
- start X by typing startx or from one of the display managers (xdm, gdm, kdm, wdm, ...) (for all).
- start a Japanese-compatible application such as Vim 6, (x)emacs21, mc-4.5, mutt-1.4, ... in kterm (for Japanese). (Emacs seems to be the most popular platform, though I do not use it.)
- press Shift+Space to toggle Japanese character input mode on and off (for Japanese).
- read the localized manual page by starting command in localized console (for all).

For other CJK language supports, see the following sections and SuSE pages for CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html).

### 9.7.10 Alternative X input methods

There are many alternative X input methods support packages available:

```
Language    LC_CTYPE      XIM server XMODIFIERS              Start key
Japanese    ja_JP*        kinput2    "@im=kinput2"           Shift-Space
Korean      ko_KR*        ami        "@im=Ami"               Shift-Space
Chinese(T) zh_TW.Big5    xcin       "@im=xcin-zh_TW.big5"   Ctrl-Space
Chinese(S) zh_CN.GB2312 xcin       "@im=xcin-zh_CN.GB2312" Ctrl-Space
```

Japanese input method kinput2 is offered by the packages such as kinput2-canna-wnn, kinput2-canna, and kinput2-wnn. Japanese needs dictionary server such as canna and freewnn-jserver to be practical.

### 9.7.11 X terminal emulators

There are many X consoles which support simple 8 bit encodings when pertinent font packages are installed:

- xterm – The X terminal emulator
- gnome-terminal – xterm for Gnome
- konsole – xterm for KDE
- rxvt – VT102 terminal (lighter)
- aterm – VT102 for Afterstep WM
- eterm – VT102 for Enlightment WM
- wterm – VT102 for WindowMaker WM

Multi-byte encoding supports of X console are provided by xterm through UTF-8 encoding (‘UTF-8 support for the X terminal emulator’ на стр. 175). Other traditional encoding supports are in progress (as of 2003). Following packages offer traditional encoding supports:

- aterm-ml – Multi-lingual
- kterm – Multi-lingual (Japanese, . . . )
- rxvt-ml – Multi-lingual
- wterm-ml – Multi-lingual
- cxterm-big5 – Chinese (Trad., Big5)
- cxterm-gb – Chinese (Simp., GB)
- cxterm-ks – Chinese (KS)
- cxterm-jis – Japanese
- hanterm-classic – Korean (Hangul)
- hanterm-xf – Korean (Hangul)
- hztty – Chinese (GB, Big5, zW/HZ)

For kterm (and possibly others), you may want to activate XIM through menu after Ctrl-middle-click mouse action.

### 9.7.12 UTF-8 support for the X terminal emulator

UTF-8 support for X terminal emulator is provided by the uxterm program in the xterm package for XFree86 4.x. It enables support for all languages. It is a wrapper around the xterm(1) program that invokes the latter program with the ”UXTerm“ X resource class set.

For example, to enable nice large display of English, Russian, Japanese, Chinese and Korean characters, add following to your ~/.Xresources after installing all the pertinent fonts:

```
! set large font
UXTerm*font: -misc-fixed-medium-r-normal-*-18-120-100-100-c-90-iso10646-1
! Use XIM for Japanese
*inputMethod: kinput2
```

Then run xrdb -merge ~/.Xresources to update X resources as described in ‘X resources’ на стр. 154.

Although most of the popular console program packages such as vim, mutt, and emacs have been made compatible with UTF-8 recently (Woody-Sarge). Program such as mc still is not UTF-8 compatible but simply 8-bit clean. If you are editing 7 bit ASCII part of unknown or mixed encoding file, it is safer to use the locale unaware 8-bit clean editor.

See The Unicode HOWTO (http://www.tldp.org/HOWTO/Unicode-HOWTO.html).

### 9.7.13 Example for UTF-8 in a framebuffer console

UTF-8 support on a FB console is provided by bterm used in the debian-installer.

### 9.7.14 Beyond locales

When you are first setting the system up for a national language environment, please consider using tasksel or aptitude to find out what packages are selected by choosing the corresponding language environment task. The package choice made is useful even for a multilingual setup. If you encounter any package dependency conflicts during the install to your carefully configured system, avoid installing any software that conflicts with the existing system. You may have to use update-alternative to regain the original state for some commands since a newly installed one may have higher priority than existing ones.

Newer major programs are using glibc 2.2 and are mostly internationalized. So a specially localized version such as jvim for Vim may not be needed as its functionality is offered by vim version 6.0 in X. In reality, it is still somewhat rough-edged. Since jvim has a version compiled with direct Japanese input method (canna) support even in the console and addresses many other Japanese-specific issues maturely, you may still want it :-)

Programs may need to be configured beyond locale configuration to enable a comfortable working environment. The language-env package and its command set-language-env greatly eases this process.

Also see the internationalization document, Introduction to i18n (http://www.debian.org/doc/manuals/intro-i18n/). It is aimed at developers but is also useful for system administrators.

# Глава 10

# Network configuration

This chapter focuses on network administration in Debian. For a general introduction to GNU/Linux networking read the Net-HOWTO (http://www.tldp.org/HOWTO/Net-HOWTO/index.html).

In order for a Debian host to be able to access the Internet its network interfaces need to be supported by the kernel and properly configured.

The first requirement is kernel support for network interface devices such as Ethernet cards, Wi-Fi cards, and modems. To obtain this support you may need to recompile the kernel or add modules to it as described in 'The Linux kernel under Debian' на стр. 101.

Configuration of network devices is explained below. The information in this chapter has been updated for Sarge. Much of it does not apply to earlier releases.

## 10.1 Basics of IP networking

A Debian host may have several interfaces each with a different Internet Protocol (IP) address. Interfaces may be of several different types, including:

- Loopback: lo
- Ethernet: eth0, eth1, ...
- Wi-Fi: wlan0, wlan1, ... [1]
- Token Ring: tr0, tr1, ...
- PPP: ppp0, ppp1, ...

There is a wide range of other network devices available, including SLIP, PLIP (serial and parallel line IP), "shaper" devices for controlling the traffic on certain interfaces, frame relay, AX.25, X.25, ARCnet, and LocalTalk.

Every network interface connected directly to the Internet (or to any IP-based network) is identified by a unique 32 bit IP address. [2] The IP address can be divided into the part that

[1] Note that a Wi-Fi interface is really an alias for an Ethernet interface that gives access to the configuration parameters peculiar to Wi-Fi. These parameters are controlled using the iwconfig program.

[2] This is true if IP version 4 is being used. In IPv6 addresses are 128 bits. See http://www.ipv6.org/.

addresses the network and the part that addresses the host. If you take an IP address, set to 1 the bits that are part of the network address and set to 0 the bits that are part of the host address then you get the net mask of the network.

Traditionally, IP networks were grouped into classes whose net address parts were 8, 16 or 24 bits in length. [3]

| | IP addresses | net mask | length |
|---|---|---|---|
| Class A | 1.0.0.0 - 126.255.255.255 | 255.0.0.0 | = /8 |
| Class B | 128.0.0.0 - 191.255.255.255 | 255.255.0.0 | = /16 |
| Class C | 192.0.0.0 - 223.255.255.255 | 255.255.255.0 | = /24 |

IP addresses not in these ranges are used for special purposes.

There are address ranges in each class reserved for use on local area networks (LANs). These addresses are guaranteed not to conflict with any addresses on the Internet proper. (By the same token, if one of these addresses is assigned to a host then that host must not access the Internet directly but must access it through a gateway that acts as a proxy for individual services or else does Network Address Translation.) These address ranges are given in the following table along with the number of ranges in each class.

| | network addresses | length | how many |
|---|---|---|---|
| Class A | 10.x.x.x | /8 | 1 |
| Class B | 172.16.x.x - 172.31.x.x | /16 | 16 |
| Class C | 192.168.0.x - 192.168.255.x | /24 | 256 |

The first address in an IP network is the address of the network itself. The last address is the broadcast address for the network. [4] All other addresses may be allocated to hosts on the network. Of these, the first or the last address is usually allocated to the Internet gateway for the network.

The routing table contains the kernel's information on how to send IP packets to their destinations. Here is a sample routing table printout for a Debian host on a local area network (LAN) with IP address 192.168.50.x/24. Host 192.168.50.1 (also on the LAN) is a router for the corporate network 172.20.x.x/16 and host 192.168.50.254 (also on the LAN) is a router for the Internet at large.

```
# route
Kernel IP routing table
Destination   Gateway        Genmask        Flags Metric Ref Use Iface
127.0.0.0     *              255.0.0.0      U     0      0   2 lo
```

[3] This system was inflexible and wasted many IP addresses, so today IPv4 networks are allocated with network address parts of varying length.

[4] The address of the network can be obtained by bitwise ANDing an address on the network with the net mask. The broadcast address can be obtained by bitwise ORing the network address with the 1's complement of the net mask.

```
192.168.50.0  *              255.255.255.0 U     0      0   137 eth0
172.20.0.0    192.168.50.1   255.255.0.0   UG    1      0     7 eth0
default       192.168.50.254 0.0.0.0       UG    1      0    36 eth0
```

- The first line after the heading says that traffic destined for network 127.x.x.x will be routed through lo, the loopback interface.
- The second line says that traffic destined for hosts on the LAN will be routed through eth0.
- The third line says that traffic destined for the corporate network will be routed toward gateway 192.168.50.1 also through eth0.
- The fourth line says that traffic destined for the Internet at large will be routed toward gateway 192.168.50.254 also through eth0.

IP addresses in the table may also appear as names that are obtained by looking up addresses in /etc/networks or by using the C Library resolver.

In addition to routing, the kernel can perform network address translation, traffic shaping and filtering.

See the Net-HOWTO (http://www.tldp.org/HOWTO/Net-HOWTO/index.html) and other networking HOWTOs (http://www.tldp.org/HOWTO/Networking-Overview-HOWTO.html) for more background information.

## 10.2 Low level network configuration

The traditional low level network configuration tools on GNU/Linux systems are the ifconfig and route programs which come in the net-tools package. These tools have officially been superseded by ip which comes in the iproute package. The ip program works with Linux 2.2 and higher and is more capable than the old tools. However, the old tools still work and are more familiar to many users.

### 10.2.1 Low level network configuration – ifconfig and route

Here is an illustration of how to change the IP address of interface eth0 from 192.168.0.3 to 192.168.0.111 and to make eth0 the route to network 10.0.0.0 via 192.168.0.1. We begin by running ifconfig and route without interface arguments in order to display the current status of all network interfaces and routing.

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
     inet addr:192.168.0.3  Bcast:192.168.255.255  Mask:255.255.0.0
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:23363 errors:0 dropped:0 overruns:0 frame:0
     TX packets:21798 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:100
     RX bytes:13479541 (12.8 MiB)  TX bytes:20262643 (19.3 MiB)
```

```
      Interrupt:9

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:16436  Metric:1
      RX packets:230172 errors:0 dropped:0 overruns:0 frame:0
      TX packets:230172 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:22685256 (21.6 MiB)  TX bytes:22685256 (21.6 MiB)
# route
Kernel IP routing table
Destination  Gateway      Genmask          Flags Metric Ref Use Iface
192.168.0.0  *            255.255.0.0      U     0      0   0 eth0
default      192.168.0.1  255.255.255.255  UG    0      0   0 eth0
```

First we bring down the interface.

```
# ifconfig eth0 inet down
# ifconfig
lo    Link encap:Local Loopback
  ... (no more eth0 entry)
# route
  ... (no more routing table entries)
```

Then we bring it up with the new IP address and new routing.

```
# ifconfig eth0 inet up 192.168.0.111 \
        netmask 255.255.255.0 broadcast 192.168.0.255
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.1 dev eth0
```

The result:

```
# ifconfig
eth0 Link encap:Ethernet  HWaddr 08:00:46:7A:02:B0
      inet addr:192.168.0.111  Bcast:192.168.0.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      ...

lo    Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      ...
# route
Kernel IP routing table
Destination  Gateway      Genmask        Flags Metric Ref Use Iface
192.168.0.0  *            255.255.255.0  U     0      0   0 eth0
10.0.0.0     192.168.0.1  255.0.0.0      UG    0      0   0 eth0
```

For more information see ifconfig(8) and route(8).

### 10.2.2 Low level network configuration – ip

The ip equivalents of the preceding ifconfig and route commands are:

- ip link show
- ip route list
- ip link set eth0 down
- ip addr del dev eth0 local 192.168.0.3
- ip addr add dev eth0 local 192.168.0.111/24 broadcast 192.168.0.255
- ip link set eth0 up
- ip route add dev eth0 to 10.0.0.0/8 src 192.168.0.111 via 192.168.0.1

The ip program prints its command syntax when run with the argument help. For example, ip link help prints:

```
Usage: ip link set DEVICE { up | down | arp { on | off } |
                     dynamic { on | off } |
                     multicast { on | off } | txqueuelen PACKETS |
                     name NEWNAME |
                     address LLADDR | broadcast LLADDR |
                     mtu MTU }
       ip link show [ DEVICE ]
```

See also ip(8).

### 10.2.3 Configuring a Wi-Fi interface

For Wi-Fi interfaces the iwconfig program which comes in the wireless-tools package is used in addition to either ifconfig or ip.

See iwconfig(8).

### 10.2.4 Configuring a PPP interface

If you access the Internet through a modem connected to a dial-up telephone line then the connection is negotiated using the Point-to-Point Protocol (PPP). Such connections are accessed as network interface ppp0, ppp1, and so on.

A PPP interface is managed by the PPP daemon pppd which comes in the ppp package. Thus, for the user, configuring a PPP interface means configuring pppd.

Configuring pppd manually

For a network link to be established, a communication port (usually a serial port) needs to be opened, commands have to be sent to a communication device (usually a modem), a telephone number may have to be dialed, identity has to be authenticated to a foreign PPP daemon, a PPP interface has to be created and then routing tables have to be modified so that traffic can be sent over the link. pppd can do all of this and consequently has a very long list of operating options. These options are described in pppd(8).

On a Debian system, global options are set up in /etc/ppp/options. User-specific options are set up in ~/.ppprc. Options that must depend on the communication port used are stored in /etc/ppp/options.portname. For example, suppose you have two modems—a built-in Lucent LT modem accessed through /dev/LT-modem and an external modem accessed through /dev/ttyS0. Create the following two options files.

```
# cat > /etc/ppp/options.LT-modem <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-LT-modem"
EOF
# cat > /etc/ppp/options.ttyS0 <<EOF
115200
init "/usr/sbin/chat -f /etc/chatscripts/setup-ttyS0"
EOF
```

These refer to the following chat scripts. First, /etc/chatscripts/setup-LT-modem.

```
ABORT ERROR
'' ATZ
OK 'ATW2X2 S7=70 S11=55'
OK AT
```

Second, /etc/chatscripts/setup-ttyS0.

```
ABORT ERROR
'' ATZ
OK 'ATL1M1Q0V1W2X4&C1&D2 S6=4 S7=70 S11=55 S95=63 S109=1 +FCLASS=0'
OK AT
```

The contents of these files must depend on your hardware, of course.

Options can also be given to pppd as arguments.

In Debian pppd is usually started using the pon command. When pon is used its first argument names an options file in /etc/ppp/peers/ which is also read by pppd. [5] This is where you set up

[5] This options file is included using the call option.

options that are specific to a particular peer—for example, a particular Internet Service Provider (ISP).

Suppose for example you commute between Amsterdam and Den Haag. In each city you have access to two ISP services—Planet and KPN. First create a basic options file for each ISP.

```
# cat > /etc/ppp/peers/KPN <<EOF
remotename KPN
noauth
user kpn
noipdefault
ipparam KPN
EOF
# cat > /etc/ppp/peers/Planet <<EOF
remotename Planet
auth
user user3579@planet.nl
noipdefault
mru 1000
mtu 1000
ipparam Planet
EOF
```

These files set options that differ between the two ISPs. Options common to both ISPs can be placed in /etc/ppp/options or in one of the interface-specific options files as appropriate.

Now create options files for each ISP in each city. In our example the only difference between connecting to an ISP in one location versus connecting in another is the chatscript that is required. (The chatscript is different because the local access telephone number is different.)

```
# cat > /etc/ppp/peers/KPN-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-Amsterdam"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/KPN-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/KPN-DenHaag"
file /etc/ppp/peers/KPN
EOF
# cat > /etc/ppp/peers/Planet-Amsterdam <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-Amsterdam"
file /etc/ppp/peers/Planet
EOF
# cat > /etc/ppp/peers/Planet-DenHaag <<EOF
connect "/usr/sbin/chat -v -f /etc/chatscripts/Planet-DenHaag"
file /etc/ppp/peers/Planet
EOF
```

The file directives each include one of the options files shown earlier. The connect directive specifies the command that pppd uses to make the connection. Normally one uses the chat program for this, adapting the chatscript to the ISP. Here are the chatscripts for Den Haag; the chatscripts for Amsterdam might be similar except for the telephone number or they might be different if the ISP offers service through another company there.

```
# cat > /etc/chatscripts/KPN-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676012321
CONNECT \d\c
EOF
# cat > /etc/chatscripts/Planet-DenHaag <<EOF
ABORT BUSY
ABORT 'NO CARRIER'
ABORT VOICE
ABORT 'NO DIALTONE'
ABORT 'NO DIAL TONE'
ABORT 'NO ANSWER'
ABORT ERROR
OK-AT-OK ATDT 0676002505
CONNECT \d\c
EOF
```

To be able to connect to these ISPs you need client names and passwords that pppd can supply to the peer on demand. This information is stored either in /etc/ppp/pap-secrets (if the PAP protocol is used) or in /etc/ppp/chap-secrets (if the CHAP protocol is used). Although CHAP is more secure, PAP is still more widely used. Because these files contain secrets, group and world should not have permission to read or write them. The format of these files is explained in pppd(8). A "secret" (third field) is looked up in the file by finding the client name (first field) and/or the server name (second field). When connecting to an ISP one generally doesn't know the server name, so one supplies a client name instead; this was done on the user lines in peers /KPN and peers/Planet above.

```
# client name        server name  secret
kpn                  *            kpn
user3579@planet.nl   *            myfavoritepet
```

See /usr/share/doc/ppp/README.Debian.gz for more information.

#### Configuring pppd using pppconfig

A quick way to configure pppd is to use the pppconfig program which comes in the package of the same name. pppconfig sets up files like those above after asking the user questions through a menu interface.

#### Configuring a PPP interface using wvdial

A different approach to using pppd is to run it from wvdial which comes in the wvdial package. Instead of pppd running chat to dial in and negotiate the connection, wvdial does the dialing and initial negotiating and then starts pppd to do the rest. Given only phone number, username, and password wvdial succeeds in making the connection in most cases.

## 10.3 Naming the computer

### 10.3.1 Hostname

A Debian system sometimes needs to identify itself by name. For this purpose a hostname is maintained by the kernel.

The initscript /etc/init.d/hostname.sh sets the hostname at boot time (using the hostname command) to the name stored in /etc/hostname. This file should contain only the hostname, not a fully qualified domain name.

To print out the current hostname run hostname without an argument.

### 10.3.2 Mailname

The mailname of a host is the name that mail-related programs use to identify the host. The file /etc/mailname contains of this name followed by a newline. The mailname is usually one of the host's fully qualified domain names. See mailname(5).

What the recipient of e-mail sees in the From: header of mail sent by your Debian host depends on how Mail User Agents (MUA) and Mail Transfer Agents (MTA) are configured. Suppose a local user foo sends a mail from a host with mailname myhost.dom. The From: header of outgoing e-mail will be:

- "From: foo@myhost.dom" if the MUA has no From: header set;
- "From: bar@myhost.dom" if the MUA has "From: bar" set;
- "From: bar@bogus.dom" if the MUA has "From: bar@bogus.dom" set.

Even when the MUA has a From: header set the MTA may add a "Sender:foo@herman.dom" header to indicate its true origin.

Of course when any involved MTA performs address rewriting as discussed in 'Setting up a catchall for nonexistent email addresses under Exim' на стр. 164 and 'Configuring selective address rewriting for outgoing mail under Exim' на стр. 165, the e-mail address seen by the recipient can be changed to anything.

## 10.4 Domain Name Service (DNS)

Hosts are referred to by domain name as well as by IP address. DNS is a client-server system in which name resolvers consult nameservers in order to associate domain names with IP addresses and other properties of hosts. The GNU C Library resolver(3) can also look up IP addresses in files or consult Network Information Services (NIS).

To see what domain name is associated with the local host, use the hostname —fqdn command. This prints out the first fully qualified domain name that the resolver finds for the local hostname. [6]

### 10.4.1 The resolver

The job of finding out what IP addresses are associated with a particular domain name is the job of a resolver. The most commonly used resolver is the set of functions that go by that name (resolver(3)) in the GNU C Library. Another is the FireDNS resolver which comes in the libfiredns package.

How the LIBC resolver resolves names is governed by the hosts line in the /etc/nsswitch.conf configuration file. This line lists the services that should be used to resolve a name: e.g., dns, files, nis, nisplus. [7] See nsswitch.conf(5). Insofar as the files service is used, the behavior of the resolver is also governed by the /etc/hosts configuration file. See hosts(5).

All of the above files are static and can be edited with your favorite editor.

Insofar as the dns service is used, the behavior of the resolver is also governed by the /etc /resolv.conf configuration file. See resolv.conf(5). One of the important functions of resolv.conf is to list the IP addresses of nameservers that will be contacted to resolve the name. This list often has to depend upon the network environment and the network environment may change from time to time while your computer is running. Programs such as pppd and dhclient are able to manipulate resolv.conf to add and remove lines, but these features do not always work properly and they conflict with one another. The resolvconf package solves the problem better by providing a standard framework for updating this file. See 'Managing nameserver information – resolvconf' на стр. 186.

### 10.4.2 Managing nameserver information – resolvconf

The resolvconf package provides a framework for dynamic management of information about available nameservers. It solves the long standing problem of how to maintain dynamic lists of nameservers for the resolver and DNS caches to use. Resolvconf sets itself up as the intermediary between programs that control network interfaces and supply nameserver information, and applications that need nameserver information.

[6] Technically, it is the FQDN returned by gethostbyname(2) for the hostname returned by gethostname(2).

[7] How the resolver resolves names is also alleged to be governed by the /etc/host.conf configuration file. The order line in this file lists the methods that should be used to resolve a name: e.g., bind, hosts, nis. See host.conf(5). I believe that this line has been superseded by the hosts line in nsswitch.conf but I am not sure.

resolvconf is designed to work without any manual configuration needing to be done. However, the package is quite new and may require some manual intervention to get it to work properly. This is certainly true if you have ever customized packages so that they update /etc/resolv.conf: you will need to disable your customizations. See /usr/share/doc/resolvconf/README.gz for details.

### 10.4.3 Caching looked-up names – nscd, dnsmasq, pdnsd, bind9

If your nameserver is slow to respond then you may want to use nscd to cache the results of things that are looked up using the libc6 resolver.

If you want to cache results for other hosts on your local network then you may want to run a caching forwarding nameserver such as dnsmasq or pdnsd.

If you wish you can also use bind9's named as a caching forwarding nameserver. It is a heavy program, though, so unless you need its advanced features you are better off with one of the packages mentioned earlier.

All of these packages work well with resolvconf.

### 10.4.4 Providing Domain Name Service – bind

If you need to provide authoritative name service for a domain then you need a fully fledged nameserver such as named which comes in the bind9 package.

If you install bind9 you should also install dnsutils. You may also want to install these utility packages: bind9-host; dns-browse; dnscvsutil; nslint. You may also want to install this documentation package: bind9-doc. You may also want to install these development packages: libbind-dev; libnet-dns-perl. If you configure interfaces using DHCP then you may find this package useful: dhcp-dns.

Install bind9 or dpkg-reconfigure it to do the basic set-up. Configuration consists of editing named.conf. In Debian this file is found in /etc/bind/ and is used mainly to define the basic DNS zones; it includes two other files: named.conf.local, used for defining local zones, and named.conf.options, used for setting options. (The latter is processed by resolvconf to produce /var/run/bind/named.options which is the same as the original except that the forwarders specification is a list of the currently available non-local nameservers. To make use of this, change the include line in named.conf so that it includes /var/run/bind/named.options. See 'Managing nameserver information – resolvconf' на стр. 186.)

Database files named in named.conf* without a full pathname will be stored in /var/cache/bind /. This is the right place to store files generated by named: for example, database files for zones for which the daemon is secondary. Static database files in /etc/bind/ are and must be referred to in named.conf by their full path names. See /usr/share/doc/bind9/README.Debian.gz for details.

## 10.5 Configuring network interfaces using DHCP

Low-level configuration of network interfaces can be automated by means of the Dynamic Host Configuration Protocol (DHCP). Your firewall or router box or your broadband ISP may furnish IP addresses and other parameters this way.

To make this work you must install one of the following packages:

- dhcp3-client (version 3, Internet Software Consortium)
- dhcpcd (Yoichi Hariguchi and Sergei Viznyuk)
- pump (Red Hat)

pump is simple and widely used. dhcp3-client is complex but more configurable. [8]

## 10.6 High level network configuration in Debian

In order to make network configuration easier Debian provides a standard high level network configuration tool consisting of the ifup and ifdown programs and the /etc/network/interfaces file. [9] If you choose to use ifupdown to do your network configuration then normally you should not use low-level commands too. [10] The ifupdown program was written with the intent that it alone be used to configure and deconfigure network interfaces.

To update interface configuration do this:

```
# ifdown eth0
# editor /etc/network/interfaces  # tweak as you wish
# ifup eth0
```

For more information see interfaces(5), /usr/share/doc/ifupdown/examples/network-interfaces.gz, and ifup(8).

### 10.6.1 Configuring an interface with a static IP address

Suppose you want to configure an Ethernet interface such that it has a fixed IP address of 192.168.0.111. This address begins with 192.168.0 so it must be on a LAN. Suppose further that 192.168.0.1 is the address of the LAN's gateway to the Internet. Edit /etc/network/interfaces so that it includes a stanza like this:

[8] As of April 2004 there is also a dhcp-client package available. This contains version 2 of the ISC DHCP Client. This has been superseded by version 3 which is currently packaged as dhcp3-client. The maintainers plan to rename dhcp3-client to dhcp-client after the release of Sarge. Make sure you do not have the experimental versions of dhcp-client installed. ifupdown does not work with them.

[9] The /etc/network/interfaces file format for current versions of ifupdown is slightly incompatible with the file format for earlier Potato versions of the package. The ifupdown post-installation script should upgrade the file automatically if necessary. However, it is a good idea to check over the converted file.

[10] This means also that you should not use other high level configuration tools such as whereami that call low level configuration tools.

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
```

You can configure other aspects of the interface or perform other actions after the interface is brought up or before it is brought down by specifying appropriate commands on "up" and "down" lines.

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        up route add -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        down route del -net 10.0.0.0 netmask 255.0.0.0 gw 192.168.0.2 dev $IFACE
        up echo Interface $IFACE going up | /usr/bin/logger -t ifup
        down echo Interface $IFACE Going down | /usr/bin/logger -t ifdown
```

Alternatively, commands can be inserted into scripts in the /etc/network/if-up.d and /etc /network/if-down.d directories. Such scripts can also implement extended options. See interfaces(5) for details. For example, the resolvconf package includes scripts that allow you to add options specifying DNS information to be included in /etc/resolv.conf while the interface is up:

```
iface eth0 inet static
        address 192.168.0.111
        netmask 255.255.255.0
        gateway 192.168.0.1
        dns-search somedomain.org
        dns-nameservers 195.238.2.21 195.238.2.22
```

The argument somedomain.org of the dns-search option corresponds to the argument of a search option in resolv.conf(5). The arguments 195.238.2.21 and 195.238.2.22 of the dns-nameservers option correspond to the arguments of nameserver options. Other recognized options are dns-domain and dns-sortlist. See 'Managing nameserver information – resolvconf' на стр. 186.

### 10.6.2 Configuring an interface using DHCP

To configure an interface using DHCP edit /etc/network/interfaces so that it includes a stanza like this:

```
iface eth0 inet dhcp
```

In order for this to work you must have installed one of the DHCP clients mentioned in 'Configuring network interfaces using DHCP' на стр. 188.

### 10.6.3 Configuring a Wi-Fi interface

The wireless-tools package includes a hook script /etc/network/if-pre-up.d/wireless-tools which makes it possible to configure Wi-Fi (802.11a/b/g) hardware before the interface is brought up. Configuration is done using the iwconfig program; see iwconfig(8). For each possible command parameter of iwconfig you can include an option in /etc/network/interfaces named like that parameter with a "wireless-" prefix. For example, to set the ESSID of eth0 to myessid and the encryption key to 123456789e prior to bringing eth0 up using DHCP, edit /etc/network /interfaces so that it includes a stanza like this:

```
iface eth0 inet dhcp
        wireless-essid myessid
        wireless-key 123456789e
```

Note that you should not use this method of setting the ESSID and key if you are running waproamd for this interface. By the time ifup is run waproamd has already set the ESSID and key. See 'Triggering network configuration – waproamd' на стр. 198.

### 10.6.4 Configuring a PPP interface

The ifup and ifdown programs use pon and poff to add and remove PPP interfaces so first read 'Configuring a PPP interface' на стр. 181.

Suppose you have set up PPP to work with peer myisp. Edit /etc/network/interfaces so that it includes a stanza like this:

```
iface ppp0 inet ppp
        provider myisp
```

With this stanza in place, ifup ppp0 does

```
pon myisp
```

Unfortunately it is currently not possible to provide additional pppd options in a ppp stanza in /etc/network/interfaces. [11]

It is currently not possible to use ifupdown to perform auxiliary configuration of PPP interfaces. Because pon exits before pppd has finished making the connection, ifup runs up scripts before the PPP interface is ready for use. Until this bug [12] is fixed it remains necessary to do auxiliary configuration in /etc/ppp/ip-up or /etc/ppp/ip-up.d/.

[11] See bug #196877 (http://bugs.debian.org/196877).
[12] See bug #127786 (http://bugs.debian.org/127786).

### 10.6.5 Configuring a PPPoE interface

Many broadband Internet Service Providers (ISPs) use PPP to negotiate connections even though customer machines are connected to them through Ethernet and/or ATM networks. This is accomplished by means of PPP over Ethernet (PPPoE) which is a technique for the encapsulation of PPP streams inside of Ethernet frames. Suppose your ISP is called myisp. First configure PPP and PPPoE for peer myisp. The easiest way to do this is to install the pppoeconf package and to run pppoeconf from the console. Then edit /etc/network/interfaces so that it includes a stanza like this:

```
iface eth0 inet ppp
        provider myisp
```

There are sometimes Maximum Transmit Unit (MTU) issues with PPPoE over Digital Subscriber Line (DSL). See DSL-HOWTO (http://www.tldp.org/HOWTO/DSL-HOWTO/) for details.

Note that if your broadband modem contains a router then the modem/router handles the PPPoE connection itself and appears on the LAN side as a simple Ethernet gateway to the Internet.

### 10.6.6 Configuring multiple Ethernet interfaces for a gateway

Suppose eth0 is connected to the Internet with a DHCP-configured IP address and eth1 is connected to the LAN with static IP address 192.168.1.1. Edit /etc/network/interfaces so that it includes stanzas like these:

```
iface eth0 inet dhcp

iface eth1 inet static
        address 192.168.1.1
        netmask 255.255.255.0
```

If you activate NAT on this host as described in 'Building a gateway router' на стр. 201 then you can share the Internet connection with all the hosts on the LAN.

### 10.6.7 Configuring virtual interfaces

Using virtual interfaces you can configure a single Ethernet card to be an interface to several IP subnetworks. For example, suppose your host is on LAN network 192.168.0.x/24. You want to connect the host to the Internet using a public IP address provided via DHCP using your existing Ethernet card. Edit /etc/network/interfaces so that it includes stanzas like these:

```
iface eth0 inet static
```

```
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255

iface eth0:0 inet dhcp
```

The interface eth0:0 is a virtual interface. When it is brought up, so will its parent eth0.

## 10.7 Network configuration using logical interface definitions

In the following it will be important for the reader to understand the difference between a physical interface and a logical interface. [13] A physical interface is what we have been calling "the interface", the thing that the kernel names eth0, eth1, ppp0, or what have you. A logical interface is a set of values that can be assigned to the variable parameters of a physical interface. If you find that confusing, replace the expression "configured as logical interface X" with the expression "configured with interface profile X" as you read.

The iface definitions in /etc/network/interfaces are actually definitions of logical interfaces, not of physical interfaces. [14] If you never want to reconfigure your interfaces then you can ignore this fact since the physical interface foo will by default be configured as logical interface foo.

However, suppose your computer is a laptop that you transport between home and work. When you connect the computer to the corporate network or to your home LAN you need to configure eth0 accordingly.

First define two logical interfaces home and work (instead of eth0 as we did earlier) which describe how the interface should be configured for the home network and the work network, respectively.

```
iface home inet static
        address 192.168.0.123
        netmask 255.255.255.0
        gateway 192.168.0.1

iface work inet static
        address 81.201.3.123
        netmask 255.255.0.0
        gateway 81.201.1.1
```

Then physical interface eth0 can be brought up for the home network with the appropriate configuration by specifying it on the command line:

[13] This terminology is used in the ifupdown documentation.

[14] Note that the interfaces named on auto lines must be physical interfaces, not logical interfaces.

```
# ifup eth0=home
```

To reconfigure eth0 for the work network issue the commands:

```
# ifdown eth0
# ifup eth0=work
```

Note that with the interfaces file written as above it will no longer be possible to bring up eth0 by doing ifup eth0 alone. The reason is that ifup uses the physical interface name as the default logical interface name and now in our example no eth0 logical interface is defined.

## 10.8 Magic network configuration

Interface names can be "mapped" to other names when ifup runs. How names are mapped can be made to depend on circumstances. Thus ifup can be so configured that it brings up a given physical interface as the appropriate logical interface among a set of predefined alternatives.

Logical interface name mapping occurs as follows:

- If no logical interface name is given on the ifup command line then the physical interface name is used as the initial logical interface name.
- If the logical interface name matches the glob-pattern of a mapping stanza then that mapping is applied to generate a new logical interface name. This is done for each mapping stanza in turn.
- If the final logical interface name is the label of a logical interface definition in /etc/network/interfaces then the physical interface is brought up as that logical interface. Otherwise ifup prints a message that it is "Ignoring unknown interface" and exits.

The syntax of a mapping stanza is:

```
mapping glob-pattern
        script script-name
        [map script input]
```

The script named in the mapping stanza is always run with the physical interface name as its argument and with the contents of all following "map" lines in the stanza (without the word "map" itself) provided to it on its standard input. The script prints the result of the mapping on its standard output before exiting.

For example, the following mapping stanza will cause ifup to bring up interface eth0 as the home logical interface.

```
mapping eth0
        script /usr/local/sbin/echo-home
```

where /usr/local/sbin/echo-home is:

```
#!/bin/sh
echo home
```

Because mapping is done with a script it is possible to select the logical interface automatically — based on some sort of test. See 'Logical interface selection using guessnet' на стр. 194 for an example of this.

### 10.8.1 Logical interface selection using guessnet

Install guessnet and then add a stanza like the following to /etc/network/interfaces:

```
mapping eth0
        script guessnet-ifupdown
        map home
        map work
```

Now when you ifup eth0, guessnet will check whether eth0 can be brought up as home or work. To do this it uses information stored in the logical interface definitions.

### 10.8.2 Automatic network configuration using laptop-net

The laptop-net package takes a different approach to automagic network reconfiguration. Laptop-net does not make use of ifupdown's logical interfaces but instead has its own system of configuration "schemes" and system "profiles". Laptop-net still uses ifup and ifdown to configure physical interfaces, though. For more information consult the well written documentation in laptop-net-doc.

## 10.9 Dealing with inconsistent naming of interfaces by the kernel

The names eth0, eth1, etc. are assigned by the kernel in the order that the kernel creates the interfaces that go by those names. While adapters that are detected at boot time are usually detected in the same order every time, and are therefore assigned the same names every time, the same is not true of adapters that are hot plugged. These can be detected in any order and end up getting assigned different names by the kernel on different occasions.

Because of this fact, on a system into which network adapters are hot plugged it won't always do to define logical interfaces in /etc/network/interfaces with names eth0, eth1, etc., and to rely on the default mapping. Instead you must give distinct names to the logical interfaces and use one of the following two methods to restrict which logical interfaces can be assigned to which adapters.

One method is to use either the nameif utility (in the net-tools package) or the more flexible ifrename utility (in the ifrename package) to make the kernel assign names to interfaces according to properties of the underlying adapters. With this naming scheme in effect, the physical interface name can be used to infer which adapter underlies it.

Another method is to use ifup's mapping mechanism in such a way that a logical interface is chosen for a physical interface being brought up according to some property of the adapter that underlies it.

Suppose, for example, you have two different network adapters which you use with networks net1 and net2, respectively. The /usr/share/doc/ifupdown/examples/ directory contains a mapping script that can be used to select a logical interface based on the Media Access Controller address (MAC address) of the adapter. First install the script to an appropriate directory.

```
# install -m770 /usr/share/doc/ifupdown/examples/get-mac-address.sh \
  /usr/local/sbin/
```

Then add a stanza like the following to /etc/network/interfaces:

```
mapping eth0
        script /usr/local/sbin/get-mac-address.sh
        map 02:23:45:3C:45:3C net1
        map 00:A3:03:63:26:93 net2
```

See 'Multi-stage mapping' на стр. 199 for a more complex example.

In applying either method the property that is most commonly used to identify the adapter is the MAC address.

## 10.10 Triggering network configuration

We have seen how interfaces can be configured or reconfigured. This needs to be done at appropriate times.

Traditionally the network was configured during the boot sequence via the /etc/rcS.d /S40networking initscript and was rarely reconfigured. Services that depended on networking were started later in the boot sequence. On shutdown or reboot the initscripts were run in the opposite order.

Currently, however, there is a trend in GNU and Linux toward supporting hardware and circumstances that change dynamically. First support was added for hot swappable PCMCIA cards; more recently the hotplug mechanism has been added so that many more peripherals can be swapped in and out while the computer is running. This includes networking hardware. Note that services that depend on hardware that is hot swapped must only be started after the hardware is inserted and must be stopped when the hardware is removed. This means that such services must be removed from the control of the System V init system and put under the control of ifupdown instead.

For example, suppose service foo controlled by initscript /etc/init.d/foo depends on dynamically reconfigured network interface eth0.

- First remove foo from the control of the init system. If you are using the sysv-rc init system then do the following. [15]

  # rm /etc/rc[2345].d/S??foo
- Then put foo under the control of ifupdown by adding up and down options to the eth0 stanza in /etc/network/interfaces which contain calls to the foo initscript:

```
iface eth0 inet dhcp
        up /etc/init.d/foo start
        down /etc/init.d/foo stop
```

### 10.10.1 Triggering network configuration at boot time

On boot the /etc/rcS.d/S40networking init script runs the command ifup -a. This brings up all physical interfaces listed in auto stanzas in /etc/network/interfaces.

These days it is often better to handle network configuration using dynamic methods. Once mechanisms for supporting dynamically changing hardware are in place it becomes simplest to treat static hardware as if it were dynamic too. Booting can then be treated as just another hotplug event. (See 'Triggering network configuration – hotplug' на стр. 196.)

However, in almost all cases one wants at least the loopback interface lo to be brought up on boot. Therefore, make sure that /etc/network/interfaces includes the following stanzas.

```
auto lo

iface lo inet loopback
```

You can list additional physical interface names in auto stanzas if you want them to be brought up on boot too. Never list PCMCIA interfaces in auto stanzas. The PCMCIA cardmgr is started later in the boot sequence than when /etc/rcS.d/S40networking runs.

### 10.10.2 Triggering network configuration – hotplug

For hot-plug support install the hotplug package.

Networking hardware can be hot plugged either at boot time or after a card (e.g., a PCMCIA card) is inserted into the machine or after a utility such as discover runs and loads necessary driver modules.

When the kernel detects new hardware it initializes the driver for the hardware and then runs the hotplug program to configure it. Later if the hardware is removed then the kernel runs hotplug again with different environment variable settings. In Debian, when hotplug is called it runs scripts in /etc/hotplug/ and /etc/hotplug.d/. See hotplug(8) for details.

[15] Note that this leaves the "stop" links (/etc/rc?.d/K??foo) behind. See 'Уровни выполнения' на стр. 22 for more information.

Newly inserted network hardware is configured by the script /etc/hotplug/net.agent. [16] Suppose your PCMCIA network card has been inserted resulting in interface eth0 becoming available for use. /etc/hotplug/net.agent does the following [17] :

```
ifup eth0=hotplug
```

Unless you have added a logical interface definition or mapping named hotplug to /etc/network /interfaces, this command will do nothing. To make it so that the command will configure eth0, add the following stanza to /etc/network/interfaces:

```
mapping hotplug
        script echo
```

As explained in 'Network configuration using logical interface definitions' на стр. 192 this will map the command shown above so that it is equivalent to the following:

```
ifup eth0=eth0
```

(Do not include a mapping stanza like this if you are using ifplugd or waproamd instances started by hotplug to control the interface.)

If you want only eth0 and no other interfaces to be brought up on hot plug then use grep instead of echo as follows:

```
mapping hotplug
        script grep
        map eth0
```

See 'Magic network configuration' на стр. 193 and /usr/share/doc/hotplug/README.Debian for more tips.

### 10.10.3 Triggering network configuration – ifplugd

The ifplugd daemon brings an interface up or down according to whether or not its underlying hardware is plugged in to a network. The program can detect a live cable connected to an Ethernet interface or an access point associated to a Wi-Fi interface (although waproamd is probably what you want to use in the latter case). When ifplugd sees that the state of the link has changed it runs a proxy script which by default calls ifup or ifdown.

[16] It may also be configured by any hook scripts that have been installed in /etc/hotplug.d/net/. The ifplugd and waproamd packages install hook scripts there, for example.

[17] As of version 0.0.20040329-4 or so, hotplug can optionally be put into modes wherein it behaves differently from how it is described here as behaving. One such mode is so-called "all" mode wherein hotplug brings up all hot plugged interfaces. The other such mode is so-called "auto" mode wherein hotplug brings up interfaces only if they are listed on auto lines in /etc/network/interfaces. In these alternative modes ifup is invoked without the =hotplug suffix.

### 10.10.4 Triggering network configuration – waproamd

The waproamd daemon is just like ifplugd except that it is designed to be used with Wi-Fi cards. It actively scans for access points to which the Wi-Fi hardware is able to associate. When association is achieved, waproamd runs ifup.

If you are using waproamd then in general you configure the Wi-Fi card via waproamd and not via wireless-* options in /etc/network/interfaces.

### 10.10.5 Network configuration and PCMCIA

There are several possible approaches to configuring PCMCIA network interfaces (for 2.4 and 2.6 kernels).

- For 32 bit PCI (CardBus) PCMCIA network cards:
  - ifupdown controlled by hotplug
    - In Woody and Sarge you must locally enable hotplug's control of ifupdown by adding a mapping stanza to /etc/network/interfaces as described in 'Triggering network configuration – hotplug' на стр. 196.
- For 16 bit ISA PCMCIA network cards:
  - ifupdown controlled by hotplug with pcmcia-cs confined to loading modules
    - recommended
    - In Woody and Sarge you must locally disable pcmcia-cs's default behavior of controlling ifupdown by adding the line exit 0 to the beginning of /etc/pcmcia/network. Also, you must locally enable hotplug's control of ifupdown by adding a mapping stanza to /etc/network/interfaces as described in 'Triggering network configuration – hotplug' на стр. 196.
  - ifupdown controlled by pcmcia-cs via the default /etc/pcmcia/network
    - deprecated but still the default for Woody and Sarge
  - low level tools controlled by pcmcia-cs via special code in /etc/pcmcia/network
    - deprecated
    - In Woody and Sarge the special code is enabled by editing /etc/pcmcia/network.opts

The recommended approach for 16 bit cards takes advantage of the fact that the Linux 2.4 hotplug subsystem now supports PCMCIA. [18]

PCMCIA network cards are hot pluggable. Accordingly, any services that require networking through a PCMCIA card should be so configured that they get started on card insertion and get stopped on card removal. This is usually accomplished by arranging for the service to start on ifup and stop on ifdown. Some people, however, choose to confine themselves to cold plugging

[18] In past releases of Debian the standard way to configure PCMCIA network cards was through the cardmgr hook scripts /etc/pcmcia/network and /etc/pcmcia/network.opts. These hook scripts were developed in the era before Linux acquired a more general purpose hot plug capability. Some people still use the Debian Woody scripts in their default state wherein they simply call ifup after the interface is added and ifdown when the interface is removed. As noted above, it is now recommended to use hotplug to do this. Others still use the special system of calling low level network configuration commands that gets activated when certain variables in /etc/pcmcia/network.opts are set to "y". This system has several problems. It is afflicted by race conditions; it only works for 16 bit PCMCIA cards; it does what is better left to ifupdown to do. Consequently it is deprecated.

their PCMCIA network card: they insert the card before booting the system and they start services that require networking through the card in the boot sequence. If you are such a person then in order to ensure that the card is fully configured before the services are started you should do the following:

- Set CARDMGR_OPTS="-f" in /etc/default/pcmcia in order to force cardmgr to run in the foreground.
- Rename /etc/rc?.d/S20pcmcia to something like /etc/rc?.d/S12pcmcia.

This hack only works for 16 bit PCMCIA cards.

Note that pcmcia-cs is still needed if you use 16 bit PCMCIA cards. The cardmgr daemon that the package contains is responsible for managing the sockets and loading driver modules. We just don't want it to call network configuration programs via /etc/pcmcia/network.

In order for cardmgr to work properly you may need to edit /etc/pcmcia/config.opts in order to configure resources assigned to 16 bit PCMCIA cards. See 'PCMCIA' на стр. 103 and the Linux PCMCIA HOWTO (http://www.tldp.org/HOWTO/PCMCIA-HOWTO.html) for more information.

## 10.11 Multi-stage mapping

Suppose your network adapters are hotplugged and you enable automatic configuration as described in 'Triggering network configuration – hotplug' на стр. 196. Suppose further that you need to map logical interfaces to "physical" interfaces depending both on the adapter underlying the physical interface (as described in 'Dealing with inconsistent naming of interfaces by the kernel' на стр. 194) and on the network connected to the interface (as described, for example, in 'Logical interface selection using guessnet' на стр. 194). You can accomplish this with multi-stage mapping.

The first mapping stage takes the hotplug group name and outputs the kernel-assigned interface name if the interface is to be hot plugged. The second mapping stage takes a kernel-assigned interface name and outputs an adapter name. The third mapping stage maps adapter names to logical interface names based on the network environment.

```
# Allow hotplug to bring up interfaces
mapping hotplug
 script echo

# Determine whether interface is wired or Wi-Fi
mapping eth?
        script /usr/local/sbin/get-mac-address.sh
        map 02:23:45:3C:45:3C wired
        map 00:A3:03:63:26:93 wifi

# Detect which wired network is available
mapping wired
        script guessnet-ifupdown
```

```
        map work-wired
        map home

# Detect which Wi-Fi network is available
mapping wifi
        script ifscout
        map starbucks
        map work-wireless

iface work-wired inet static
 ...
```

## 10.12 Network service configuration

Typical network service configuration on the desktop or home server environment involves:

- The Internet super-server and TCP/IP daemon wrapper, see ‘Restricting access to services’ на стр. 139.
  - /etc/inetd.conf
- ssh: OpenSSH secure shell, see ‘SSH’ на стр. 159.
  - /etc/ssh/ssh_config
  - /etc/ssh/sshd_config
- exim: mail transport agent, see ‘Mailname’ на стр. 185 and ‘Mail transport agents (MTAs)’ на стр. 163.
  - /etc/exim/exim.conf
  - /etc/mailname
  - /etc/aliases
  - /etc/email-addresses
- fetchmail: daemon to fetch mail from a POP3 account, see ‘Fetching mail – Fetchmail’ на стр. 165.
  - /etc/fetchmailrc
- procmail: local mail delivery and filter program, see ‘Processing mail – Procmail’ на стр. 166.
  - ~/.procmailrc
- Hostname and DNS (proxy, cache, ...), see ‘Hostname’ на стр. 185 and ‘Domain Name Service (DNS)’ на стр. 186.
  - /etc/host.conf
  - /etc/hostname
  - /etc/hosts
  - /etc/hosts.allow
  - /etc/hosts.deny
  - /etc/resolv.conf
  - /etc/bind/named.conf (edit)
  - /etc/bind/db.lan (add for LAN hosts)
  - /etc/bind/db.192.168.0 (add for LAN reverse)

- DHCP, see ‘Configuring network interfaces using DHCP’ на стр. 188.
  - /etc/dhcp3/dhclient.conf (DHCP client side)
  - /etc/default/dhcp3-server (DHCP server side)
  - /etc/dhcp3/dhcpd.conf (DHCP server side)
- cvs: concurrent versions system, see ‘Concurrent Versions System (CVS)’ на стр. 213.
  - /etc/cvs-cron.conf
  - /etc/cvs-pserver.conf
- nfs-kernel-server: network file system, see ‘Конфигурация NFS’ на стр. 40. (for unix-like systems)
  - /etc/exports
- samba: network file and printer share for Windows, see ‘Настройка Samba’ на стр. 41 and ‘Samba’ на стр. 133.
  - /etc/samba/smb.conf
- Printer daemon system, see ‘Настройка принтера’ на стр. 41.
  - /etc/printcap (for lpr)
- apache and apache2: web server.
  - /etc/apache/*
  - /etc/apache2/*
- squid: web proxy cache server.
  - /etc/squid/*

## 10.13 Network troubleshooting

If you encounter problems then check the output of the following as the first reality check:

```
# ifconfig
# cat /proc/pci
# cat /proc/interrupts
# dmesg | more
```

Also see the sections following ‘Network testing basics’ на стр. 129.

If you have problems with certain websites, see ‘Странные проблемы доступа на некоторые web-сайты’ на стр. 45.

## 10.14 Building a gateway router

A Debian host can be an all-purpose gateway machine that does Network Address Translation (NAT, also known as masquerading), mail transfer, DHCP, DNS caching, HTTP proxy caching, CVS service, NFS serving, and Samba serving. See ‘Хосты и IP-адреса для использования в локальных сетях’ на стр. 31 for the example of such set up.

### 10.14.1 Netfilter configuration

The netfilter/iptables project is a firewalling subsystem for Linux 2.4 and after. See Netfilter (http://www.netfilter.org/), where many network configuration issues are explained.

#### Basics of netfilter

Netfilter process packets use five built-in chains: PREROUTING, INPUT, FORWARD, OUTPUT, and POSTROUTING.

```
                  routing
                  decision
IN ------> PRE ---> ------> FORWARD -----> ----> POST -----> OUT
interface  ROUTING  \       filter       /       ROUTING     interface
           DNAT     |       tracking     ^       SNAT
           REDIRECT |                    |       MASQUERADE
                    v                    |
                  INPUT                OUTPUT
                    | filter             ^ filter,DNAT
                    v                    |
                    \--> Local Process --/
                         user-space programs
```

#### Netfilter table

Packets are processed at each built-in chain using the following tables.

- filter (packet filter, default)
  - INPUT (for packets coming into the box itself)
  - FORWARD (for packets being routed through the box)
  - OUTPUT (for locally generated packets).
- nat (network address translation )
  - PREROUTING (for altering packets as soon as they come in)
  - OUTPUT (for altering locally generated packets before routing)
  - POSTROUTING (for altering packets as they are about to go out)
- mangle (network address mangling, good only after 2.4.18)
  - all five built-in chains.

#### Netfilter target

Firewall rules have several targets:

- four basic targets:
  - ACCEPT means to let the packet through.
  - DROP means to drop the packet.
  - QUEUE means to pass the packet to userspace (if supported by the kernel).

  - RETURN means stop traversing this chain and resume at the next rule in the previous (calling) chain.
- extended targets:
  - LOG turns on kernel logging.
  - REJECT sends back an error packet and drops the packet.
  - SNAT alters the source address of the packet and is used only in the POSTROUTING chain. (nat table only)
    --to-source ipaddr[-ipaddr][:port-port]
  - MASQUERADE is the same as SNAT but for dynamically assigned IP (dialup) connections. (nat table only)
    --to-ports port[-port]
  - DNAT alters the destination address of the packet and is used in the PREROUTING and OUTPUT chains, and user-defined chains which are only called from those chains. (nat table only)
    --to-destination ipaddr[-ipaddr][:port-port]
  - REDIRECT alters the destination IP address to send the packet to the machine itself.
    --to-ports port[-port]

Netfilter commands

The basic commands of iptables are:

```
iptables -N chain                # create a chain

iptables -A chain \              # add rule to chain
        -t table \               # use table (filter, nat, mangle)
        -p protocol \            # tcp, udp, icmp, or all,
        -s source-address[/mask] \
        --sport port[:port] \    # source port if -p is tcp or udp
        -d destination-address[/mask] \
        --dport port[:port] \    # dest. port if -p is tcp or udp
        -j target \              # what to do if match
        -i in-interface-name \   # for INPUT,  FORWARD, PREROUTING
        -o out-interface-name    # for FORWARD, OUTPUT, POSTROUTING
```

Network Address Translation

Machines on a LAN can access Internet resources through a gateway that translates IP address on the LAN to IP addresses usable on the Internet.

```
# apt-get install ipmasq
```

Apply example rules to strengthen the ipmasq protection. See /usr/share/doc/ipmasq/examples/stronger/README. For Debian kernel-image-2.4 under woody, make sure to load the proper modules. Sarge version of ipmasq fixed this issue. See 'Network function' на стр. 104 for configuration instructions.

For Debian kernel-image-2.2, edit Z92timeouts.rul in /etc/masq/rules as follows to ensure a longer connection to remote sites (good for large emails, etc.):

```
# tcp, tcp-fin, udp
# 2hr, 10 sec, 160 sec - default
# 1 day, 10 min, 10 min - longer example
$IPCHAINS -M -S 86400 600 600
```

Also, if the network is accessed through a PCMCIA NIC, ipmasq needs to be started either from /etc/pcmcia/network.opts (read: /usr/share/doc/ipmasq/ipmasq.txt.gz) or from /etc/network /interfaces (read: 'Network configuration and PCMCIA' на стр. 198 and 'Triggering network configuration' на стр. 195).

#### Redirect SMTP connection (2.4)

Suppose you have a notebook PC which is configured to use other LAN environments and you want to use your mail user agent on the notebook PC without reconfiguring it.

Adding the following rules through the iptables command to the gateway machine will redirect the SMTP connection to the gateway machine.

```
# iptables -t nat -A PREROUTING -s 192.168.1.0/24 -j REDIRECT \
         -p tcp --dport smtp --to-port 25 # smtp=25, INPUT is open
```

For a more thorough redirect rule set consider installing the ipmasq package and adding M30redirect.def(http://www.debian.org/doc/manuals/debian-reference/examples/) to the /etc /ipmasq/rules/ directory.

### 10.14.2 Manage multiple net connections

[FIXME] Policy routing (by Phil Brutsche <pbrutsch@tux.creighton.edu>): See the iproute manual (http://lartc.org/) for details. Traffic control (tc) may also be interesting.

Environment:

```
eth0: 192.168.1.2/24; gateway 192.168.1.1
eth1: 10.0.0.2/24; gateway 10.0.0.1
No masquerading on this machine.
```

Special magic:

```
 1 ip rule add from 192.168.1.2 lookup 1
```

```
ip rule add from 10.0.0.2 lookup 2
ip route add to default via 10.0.0.1 metric 0
ip route add to default via 192.168.1.1 metric 1
ip route add table 1 to 192.168.1.0/24 via eth0
ip route add table 1 to 10.0.0.2/24 via eth1
ip route add table 1 to default via 192.168.1.1
ip route add table 2 to 192.168.1.0/24 via eth0
ip route add table 2 to 10.0.0.2/24 via eth1
ip route add table 2 to default via 10.0.0.2
```

[FIXME] I've never done this. How to set up dialup as backup to a fast connection with autodial features? Please send me a patch here :)

# Глава 11

# Editors

## 11.1 Popular editors

Linux offers many alternatives for console text editors. Among them:

- vim: Powerful and light BSD-heritage editor. VI iMproved.
- emacs: Ultimate and heavy GNU-heritage editor. RMS (Richard M. Stallman) original.
- xemacs: Emacs: The Next Generation, originally from Lucid.
- mcedit: Newbie GNU editor. Identical to mc internal editor. See 'Редактор в MC' на стр. 52.
- ae: Default small editor (Potato). Avoid this.
- nano: Default small GNU editor (Woody). Emulates pico.
- joe: For WordStar or TurboPascal old-timers.
- jed: Fast, full-featured menu-driven editor with Emacs key bindings.
- jove: Very small editor with Emacs key bindings.
- nvi: New vi. Bug-for-bug compatible with the original vi.

Use update-alternatives —config editor to change the default editor. Also, many programs use environment variables EDITOR or VISUAL to decide which editor to use. See 'Редактор в MC' на стр. 52.

Also a few X-based text editors are noteworthy:

- gvim: Vim with GUI (vim and vim-gtk package)
- emacs: The One True Emacs (auto-detect X).
- xemacs: Next generation Emacs (auto-detect X).

These xclient commands take standard options such as -fn a24, which makes life easy for older folks like me :) See 'X clients' на стр. 148.

## 11.2 Rescue editors

There are a few editors which reside in /bin. One of these should be installed to ease editing files when /usr is not accessible.

- elvis-tiny: Minimum vi editor (vi to start)
- nano-tiny: Minimum non-vi editor (nano-tiny to start)
- ed: Minimum editor (always there but tough to use)

## 11.3 Emacs and Vim

### 11.3.1 Vim hints

Read the "VIM - main help file" document by pressing <F1> while running the program.

| | |
|---|---|
| <F1> | Help |
| <esc> | Back to normal mode |
| V | Visual mode |
| i | Insert mode |
| : | Command-line commands |
| :set tw=72 | Set text width to 72 |
| <F11> | Insert (paste) mode |
| :r! date -R | Insert RFC-822 date |
| qa | Record keystrokes into register a |
| q | Stop keystroke recording |
| @a | Execute keystrokes from register a |
| :edit foo.txt | Edit another file by loading foo.txt |
| :wnext | Write current file and edit next file |

q and @ can be used for simple macro recording and playback. For instance, to create a macro that inserts HTML italics tags around the word at the cursor, you could enter qii<i>^[ea</i>^[q (where ^[ is the ESC key). Then typing @i at the start of a word would add the tags <i> and </i>.

See also 'Using GnuPG with Vim' на стр. 243.

### 11.3.2 Emacs hints

| | |
|---|---|
| <F1> | Help |
| <F10> | Menu |
| C-u M-! date -R | Insert RFC-822 date |

### 11.3.3 Starting the editor

| | | |
|---|---|---|
| start editor: | emacs filename | vim filename |
| start in vi compatible: | | vim -C |
| start in vi non-compatible: | | vim -N |
| start with compile default: | emacs -q | vim -N -u NONE |

### 11.3.4 Editor command summary (Emacs, Vim)

| | | |
|---|---|---|
| exit: | C-x C-c | :qa /:wq /:xa /:q! |
| Get back/command mode: | C-g | <esc> |
| Backward(left): | C-b | h |
| Forward(right): | C-f | l |
| Next(down): | C-n | j |
| Previous(up): | C-p | k |
| stArt of line(^): | C-a | 0 |
| End of line($): | C-e | $ |
| mUltiple commands: | C-u nnn cmd | nnn cmd |
| Multiple commands: | M-digitkey cmd | |
| save File: | C-x C-s | :w file |
| beginning of buffer: | M-< | 1G |
| end of buffer: | M-> | G |
| scroll forward 1 screen: | C-v | ^F |
| scroll forward 1/2 screen: | | ^D |
| scroll forward 1 line: | | ^E |
| scroll backward 1 screen: | M-v | ^B |
| scroll backward 1/2 screen: | | ^U |
| scroll backward 1 line: | | ^Y |
| scroll the other window: | M-C-v | |
| delete under cursor: | C-d | x |
| delete from cursor to eol: | C-k | D |
| iSearch forward: | C-s | |
| isearch Reverse: | C-r | |
| Search forward: | C-s enter | / |
| search Reverse: | C-r enter | ? |
| isearch regexp: | M-C-s | |
| isearch backward regexp: | M-C-r | |
| search regexp: | M-C-s enter | / |
| search backward regexp: | M-C-r enter | ? |
| Help: | C-h C-h | :help |
| Help Apropos: | C-h a | |
| Help key Bindings: | C-h b | :help [key] |
| Help Info: | C-h i | |
| Help Major mode: | C-h m | |
| Help tutorial: | C-h t | :help howto |
| Undo: | C-_ | u |
| Redo: | C-f | ^R |
| Mark cursor position: | C-@ | m{a-zA-Z} |
| eXchange Mark and position: | C-x C-x | |
| goto mark in current file: | | '{a-z} |
| goto mark in any file: | | '{A-Z} |
| copy region: | M-w | {visual}y |
| kill region: | C-w | {visual}d |

<table>
<tr><td>Yank and keep buffer:</td><td>C-y</td><td></td></tr>
<tr><td>Yank from kill buffer:</td><td>M-y</td><td>p</td></tr>
<tr><td>convert region to Upper:</td><td>C-x C-u</td><td>{visual}U</td></tr>
<tr><td>convert region to Lower:</td><td>C-x C-l</td><td>{visual}u</td></tr>
<tr><td>Insert special char:</td><td>C-q octalnum/keystroke</td><td></td></tr>
<tr><td></td><td></td><td>^V decimal/keystroke</td></tr>
<tr><td>replace:</td><td>M-x replace-string</td><td>:%s/aaa/bbb/g</td></tr>
<tr><td>replace regexp:</td><td>M-x replace-regexp</td><td>:%s/aaa/bbb/g</td></tr>
<tr><td>query replace:</td><td>M-%</td><td>:%s/aaa/bbb/gc</td></tr>
<tr><td>query replace:</td><td>M-x query-replace</td><td></td></tr>
<tr><td>query replace regexp:</td><td>M-x query-replace-regexp</td><td></td></tr>
<tr><td>Open file:</td><td>C-x C-f</td><td>:r file</td></tr>
<tr><td>Save file:</td><td>C-x C-s</td><td>:w</td></tr>
<tr><td>Save all buffers:</td><td>C-x s</td><td>:wa</td></tr>
<tr><td>Save as:</td><td>C-x C-w file</td><td>:w file</td></tr>
<tr><td>Prompt for buffer:</td><td>C-x b</td><td></td></tr>
<tr><td>List buffers:</td><td>C-x C-b</td><td>:buffers</td></tr>
<tr><td>Toggle read-only:</td><td>C-x C-q</td><td>:set ro</td></tr>
<tr><td>Prompt and kill buffer:</td><td>C-x k</td><td></td></tr>
<tr><td>Split vertical:</td><td>C-x 2</td><td>:split</td></tr>
<tr><td>Split horizontal:</td><td>C-x 3</td><td>:vsplit (ver. 6)</td></tr>
<tr><td>Move to other window:</td><td>C-x o</td><td>^Wp</td></tr>
<tr><td>Delete this window:</td><td>C-x 0</td><td>:q</td></tr>
<tr><td>Delete other window(s):</td><td>C-x 1</td><td>^Wo</td></tr>
<tr><td>run shell in bg:</td><td>M-x compile</td><td></td></tr>
<tr><td>kill shell run in bg:</td><td>M-x kill-compilation</td><td></td></tr>
<tr><td>run make:</td><td></td><td>:make Makefile</td></tr>
<tr><td>check error message:</td><td>C-x‘</td><td>:echo errmsg</td></tr>
<tr><td>run shell and record:</td><td>M-x shell</td><td>:!script -a tmp</td></tr>
<tr><td>...clean BS, ...</td><td></td><td>:!col -b <tmp >record</td></tr>
<tr><td>...save/recall shell record:</td><td>C-x C-w record</td><td>:r record</td></tr>
<tr><td>run shell:</td><td>M-! sh</td><td>:sh</td></tr>
<tr><td>run command:</td><td>M-! cmd</td><td>:!cmd</td></tr>
<tr><td>run command and insert:</td><td>C-u M-! cmd</td><td>:r!cmd</td></tr>
<tr><td>run filter:</td><td>M-| file</td><td>{visual}:w file</td></tr>
<tr><td>run filter and insert:</td><td>C-u M-| filter</td><td>{visual}:!filter</td></tr>
<tr><td>show option</td><td></td><td>:se[t] {option}?</td></tr>
<tr><td>reset option to default</td><td></td><td>:se[t] {option}&</td></tr>
<tr><td>reset boolean option</td><td></td><td>:se[t] no{option}</td></tr>
<tr><td>toggle boolean option</td><td></td><td>:se[t] inv{option}</td></tr>
<tr><td>wrap text at column 72</td><td></td><td>:se tw=72</td></tr>
<tr><td>do not wrap</td><td></td><td>:se tw=0</td></tr>
<tr><td>autoindent</td><td></td><td>:se ai</td></tr>
<tr><td>expand tab</td><td></td><td>:se et</td></tr>
<tr><td>specify comment (mail)</td><td></td><td>:se comments=n:>,n:\|</td></tr>
</table>

| | |
|---|---|
| run GDB | M-x gdb |
| describe GDB mode | C-h m |
| step one line | M-s |
| next line | M-n |
| step one instruction (stepi) | M-i |
| finish current stack frame | C-c C-f |
| continue | M-c |
| up arg frames | M-u |
| down arg frames | M-d |
| copy number from point, insert at the end | C-x & |
| set break point | C-x SPC |

### 11.3.5 Vim configuration

In order to use all Vim features and syntax highlighting, include the following lines in ~/.vimrc or /etc/vimrc:

```
set nocompatible
set nopaste
set pastetoggle=<f11>
syn on
```

Paste mode enables one to avoid autoindent interfering with cut-and-paste operations on a console terminal. It does more than just a simple ":set noai".

See 'Using GnuPG with Vim' на стр. 243 for GnuPG integration.

### 11.3.6 Ctags

apt-get install exuberant-ctags and run ctags on the source files. Type :tag function_name in Vim to jump to the line where function_name starts. The tags work for C, C++, Java, Python, and many other languages.

Emacs has the same ctags capabilities.

### 11.3.7 Convert a syntax-highlighted screen to HTML source

so \$VIMRUNTIME/syntax/2html.vim from Vim command mode will convert highlighted text to HTML text. Save with :w file.html and :q. Useful for C code, etc.

### 11.3.8 Split screen with vim

vim can edit multiple files in a multi-split-screen environment. Type :help usr_08.txt for details.

To split the screen display between different files, type at the vi command prompt:

```
:split another-file
:vsplit another-file
```

Or at a shell prompt:

```
$ vi -o file1.txt file2.txt   # Horizontal split
$ vi -O file1.txt file2.txt   # Vertical split
```

will provide multiwindow vi.

```
$ vimdiff file.txt~ file.txt          # check recent changes of file.txt
$ vimdiff file.en.sgml file.fr.sgml  # check changes of translation
$ gvimdiff file.txt~ file.txt         # in X
```

will provide a nice view of differences between an original and a backup file. In SGML it matches tags, so comparing translations in this mode works very well.

Special cursor movements with Ctrl-W commands:

```
Ctrl-W +      increase the size of a window
Ctrl-W -      decrease the size of a window
Ctrl-W h      move to the window left
Ctrl-W j      move to the window below
Ctrl-W k      move to the window above
Ctrl-W l      move to the window right
...
```

Use the following to control screen scrolling:

```
:set scrollbind
:set noscrollbind
```

# Глава 12

# Version Control Systems

## 12.1 Concurrent Versions System (CVS)

Check /usr/share/doc/cvs/html-cvsclient, /usr/share/doc/cvs/html-info, /usr/share/doc /cvsbook with lynx or run info cvs and man cvs for detailed information.

### 12.1.1 Installing a CVS server

The following setup will allow commits to the CVS repository only by a member of the "src" group, and administration of CVS only by a member of the "staff" group, thus reducing the chance of shooting oneself.

```
# cd /var/lib; umask 002; mkdir cvs # [Woody] FSH
# apt-get install cvs cvs-doc cvsbook
# export CVSROOT=/var/lib/cvs
# cd $CVSROOT
# chown root:src .  # "staff" to restrict more for starting project.
# chmod 3775 .             # If above uses "staff", use 2775
# cvs -d /var/lib/cvs init # safer to specify -d here explicitly!
# cd CVSROOT
# chown -R root:staff .
# chmod 2775 .
# touch val-tags
# chmod 664 history val-tags
# chown root:src history val-tags
```

### 12.1.2 CVS session examples

The following will set up shell environments for CVS repository access.

Anonymous CVS (download only)

Read-only remote access:

```
$ export CVSROOT=:pserver:anonymous@cvs.sf.net:/cvsroot/qref
$ cvs login
$ cvs -z3 co qref
```

Use local CVS server

Local access from a shell on the same machine:

```
$ export CVSROOT=/var/lib/cvs
```

Use remote CVS pserver

Remote access without SSH (use RSH protocol capability in cvs):

```
$ export CVSROOT=:pserver:account@cvs.foobar.com:/var/lib/cvs
$ cvs login
```

This is prone to eavesdropping attack.

Use remote CVS through ssh

Remote access with SSH:

```
$ export CVSROOT=:ext:account@cvs.foobar.com:/var/lib/cvs
```

or for SourceForge:

```
$ export CVSROOT=:ext:account@cvs.sf.net:/cvsroot/qref
```

You can also use RSA authentication (‘Connecting with fewer passwords – RSA’ на стр. 161), which eliminates the password prompt.

Create a new CVS archive

For,

| ITEM | VALUE | MEANING |
|---|---|---|
| source tree: | ~/project-x | All source codes |
| Project name: | project-x | Name for this project |
| Vendor Tag: | Main-branch | Tag for the entire branch |
| Release Tag: | Release-initial | Tag for a specific release |

Then,

```
$ cd ~/project-x                 # dive into source directory
 ... create a source tree ...
$ cvs import -m "Start project-x" project-x Main-branch Release-initial
$ cd ..; rm -R ~/project-x
```

Work with CVS

To work with project-x using the local CVS repository:

```
$ cd                             # move to the work area
$ cvs co project-x               # get sources from CVS to local
$ cd project-x
 ... make changes to the content ...
$ cvs diff -u                    # similar to diff -u repository/ local/
$ cvs up -C modified_file        # undo changes to a file
$ cvs ci -m "Describe change"    # save local sources to CVS
$ vi newfile_added
$ cvs add newfile_added
$ cvs ci -m "Added newfile_added"
$ cvs up                         # merge latest version from CVS
 ... to create all newly created subdirectories from CVS, use
 ... "cvs up -d -P" instead
 ... watch out for lines starting with "C filename"
 ... unmodified code is moved to '.#filename.version'
 ... search for "<<<<<<<" and ">>>>>>>" in filename
$ cvs tag Release-1              # add release tag
 ... edit further ...
$ cvs tag -d Release-1           # remove release tag
$ cvs ci -m "more comments"
$ cvs tag Release-1              # re-add release tag
$ cd                             # move back to the work area
$ cvs co -r Release-initial -d old project-x
```

```
 ... get original version to old directory
$ cd old
$ cvs tag -b Release-initial-bugfixes # create branch (-b) tag
 ... now you can work on the old version (Tag=sticky)
$ cvs update -d -P              # don't create empty directories
 ... source tree now has sticky tag "Release-initial-bugfixes"
 ... work on this branch
$ cvs up -d -P # sync with files modified by others on this branch
$ cvs ci -m "check into this branch"
$ cvs update -kk -A -d -P
 ... remove sticky tag and forget contents
 ... update from main trunk without keyword expansion
$ cvs update -kk -d -P -j Release-initial-bugfixes
 ... Merge from Release-initial-bugfixes branch into the main
 ... trunk without keyword expansion.  Fix conflicts with editor.
$ cvs ci -m "merge Release-initial-bugfixes"
$ cd
$ tar -cvzf old-project-x.tar.gz old  # make archive, -j for bz2
$ cvs release -d old            # remove local source (optional)
```

Nice options to remember (use as first argument(s) to cvs):

```
-n     dry run, no effect
-t     display messages showing steps of cvs activity
```

Export files from CVS

To get the latest version from CVS, use ”tomorrow“:

```
$ cvs ex -D tomorrow module_name
```

Administer CVS

Add alias to a project (local server):

```
$ su - admin            # a member of staff
$ export CVSROOT=/var/lib/cvs
$ cvs co CVSROOT/modules
$ cd CVSROOT
$ echo "px -a project-x" >>modules
$ cvs ci -m "Now px is an alias for project-x"
$ cvs release -d .
$ exit                  # or Ctrl-D to get back from su
```

```
$ cvs co -d project px
 ... check out project-x (alias:px) from CVS to directory project
$ cd project
 ... make changes to the content ...
```

### 12.1.3 Troubleshooting CVS

File permissions in repository

CVS will not overwrite the current repository file but replaces it with another one. Thus, write permission to the repository directory is critical. For every new repository creation, run the following to ensure this condition if needed.

```
# cd /var/lib/cvs
# chown -R root:src repository
# chmod -R ug+rwX   repository
# chmod    2775     repository  # if needed, this and subdirectory
```

Execution bit

A file's execution bit is retained when checked out. Whenever you see execution permission problems in checked-out files, change permissions of the file in the CVS repository with the following command.

```
# chmod ugo-x filename
```

### 12.1.4 CVS commands

Here are CVS commands with their shortcuts.

```
{add|ad|new} [-k kflag] [-m 'message'] files...
{admin|adm|rcs} [rcs-options] files...
{annotate|ann} [options] [files...]
{checkout|co|get} [options] modules...
{commit|ci|com}   [-lnR]  [-m  'log_message'  |  -f  file] \
        [-r revision] [files...]
{diff|di|dif} [-kl] [rcsdiff_options] [[-r rev1 | -D date1] \
        [-r rev2 |  -D date2]] [files...]
{export|ex|exp} [-flNn] -r rev|-D date [-d dir] [-k kflag] module...
{history|hi|his} [-report] [-flags] [-options args] [files...]
{import|im|imp} [-options] repository vendortag releasetag...
{login|logon|lgn}
{log|lo|rlog} [-l] rlog-options [files...]
```

```
{rdiff|patch|pa} [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...
{release|re|rel} [-d] directories...
{remove|rm|delete} [-lR] [files...]
{rtag|rt|rfreeze} [-falnR] [-b] [-d] [-r tag | -D date] \
        symbolic_tag modules...
{status|st|stat} [-lR] [-v] [files...]
{tag|ta|freeze} [-lR] [-F] [-b] [-d] [-r tag | -D date] [-f] \
        symbolic_tag [files...]
{update|up|upd} [-AdflPpR] [-d] [-r tag|-D date] files...
```

## 12.2 Subversion

Subversion is a next-generation version control system that is intended to replace CVS. The developers currently consider it to be in the "alpha" stage, but it is probably stable enough for most uses. At the time of this writing, Subversion is only available in Debian unstable.

### 12.2.1 Installing a Subversion server

The subversion-server meta-package depends on the packages needed (libapache2-dav-svn and subversion-tools) to set up a server.

#### Setting up a repository

Currently, the subversion package does not set up a repository, so one must be set up manually. One possible location for a repository is in /var/local/repos.

Create the directory:

```
# mkdir -p /var/local/repos
```

Create the repository database:

```
# svnadmin create /var/local/repos
```

Make the repository writable by the WWW server:

```
# chown -R www-data:www-data /var/local/repos
```

Configuring Apache2

To allow access to the repository via user authentication, add (or uncomment) the following in /etc/apache2/mods-available/dav_svn.conf:

```
<Location /repos>
  DAV svn
  SVNPath /var/local/repos
  AuthType Basic
  AuthName "Subversion repository"
  AuthUserFile /etc/subversion/passwd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

Then, create a user authentication file with the command:

```
htpasswd2 -c /etc/subversion/passwd some-username
```

Restart Apache2, and your new Subversion repository will be accessible with the URL http://hostname/repos.

### 12.2.2 Moving a CVS repository to Subversion

### 12.2.3 Subversion usage examples

The following sections teach you how to use different commands in Subversion.

Create a new Subversion archive

To create a new Subversion archive, type the following:

```
$ cd ~/your-project          # go to your source directory
$ svn import http://localhost/repos your-project \
  project-name -m "initial project import"
```

This creates a directory named project-name in your Subversion repository which contains your project files. Look at http://localhost/repos/ to see if it's there.

Working with Subversion

Working with project-y using Subversion:

```
$ cd                                    # move to the work area
$ svn co http://localhost/repos/project-y  # Check out sources
$ cd project-y
 ... do some work ...
$ svn diff                              # similar to diff -u repository/ local/
$ svn revert modified_file              # undo changes to a file
$ svn ci -m "Describe changes"  # check in your changes to the repository
$ vi newfile_added
$ svn add newfile_added
$ svn add new_dir                       # recursively add all files in new_dir
$ svn add -N new_dir2                   # nonrecursively add the directory
$ svn ci -m "Added newfile_added, new_dir, new_dir2"
$ svn up                                # merge in latest version from repository
$ svn log                               # shows all changes committed
$ svn copy http://localhost/repos/project-y \
      http://localhost/repos/project-y-branch \
      -m "creating my branch of project-y"  # branching project-y
$ svn copy http://localhost/repos/project-y \
      http://localhost/repos/proj-y_release1.0 \
      -m "project-y 1.0 release"     # added release tag
 ... note that branching and tagging are the same. The only difference
 ... is that branches get committed whereas tags do not.

 ... make changes to branch ...

$ # merge branched copy back to main copy
$ svn merge http://localhost/repos/project-y \
   http://localhost/repos/project-y-branch
$ svn co -r 4 http://localhost/repos/project-y # get revision 4
```

# Глава 13

# Programming

Do not use ”test“ as the name of an executable test file. test is a shell built-in.

## 13.1 Where to start

References:

- Documents and examples under /usr/share/doc/package
- Unix / Programming Information (http://arioch.unomaha.edu/~jclark/#info)
- Linux Programming Bible (John Goerzen/IDG books)

Many long info documents can be obtained as paperbacks from GNU (http://www.gnu.org/).

The next four sections contain sample scripts in different languages for creating a text file of account information to be added to /etc/passwd using a batch processor such as the newusers program. Each script requires as input a file with lines of the form first_name last_name password. (Actual user home directories will not be created via these scripts.)

## 13.2 Shell

Reading shell scripts is the best way to understand how a Unix-like system works. Here, I give some pointers and reminders for shell programming. See Shell Mistakes (http://www.greenend.org.uk/rjk/2001/04/shell.html) to learn from mistakes.

### 13.2.1 Bash – GNU standard interactive shell

References for Bash:

- bash(1)
- info bash
- the LDP BASH Programming - Introduction HOWTO (http://www.tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html) as starter information.

- mc /usr/share/doc/bash/examples/ /usr/share/doc/bash/ (Install the bash-doc package to see the example files.)
- Learning the bash Shell, 2nd edition (O'Reilly)

Short program example (creates account entries for newusers from standard input):

```
#!/bin/bash
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
pid=1000;
while read n1 n2 n3 ; do
if [ ${n1:0:1} != "#" ]; then
let pid=$pid+1
echo ${n1}_${n2}:password:${pid}:${pid}:,,,/home/${n1}_${n2}:/bin/bash
fi
done
```

### 13.2.2 POSIX shells

Several packages provide a POSIX shell in Debian:

- dash (Sarge)
  - Priority: optional
  - Installed-Size: 176
  - Smallest and fastest by far – best for initial boot
- ash (Woody)
  - Priority: optional
  - Installed-Size: 180
  - Smaller and much faster – good for initial boot
- bash
  - Essential: yes
  - Priority: required
  - Installed-Size: 580
  - Larger and featureful – many extensions implemented
- pdksh
  - Priority: optional
  - Installed-Size: 408
  - Complete AT&T ksh look-alike

If you are writing a shell script for portability, it is best to write it as a POSIX shell script. Use /bin/sh linked to ash (or dash) to test its POSIX compliance. Avoid writing scripts with bashism or the zshism which seems to resemble csh syntax. For example, avoid:

- if [ foo == bar ] ; then ...
- diff -u file.c{.orig,}
- mkdir /foo{bar,baz}

The description for the shell in this document applies only for the POSIX type shells and thus does not apply for the csh type shells including tcsh.

### 13.2.3 Shell parameters

Several special parameters to remember:

| | |
|---|---|
| $0 | = name of the shell or shell script |
| $1 | = first(1) shell argument |
| ... | |
| $9 | = ninth(9) shell argument |
| $# | = number of positional parameters |
| "$*" | = "$1 $2 $3 $4 ... $n" |
| "$@" | = "$1" "$2" "$3" "$4" ... "$n" |
| $? | = exit status of the most recent command |
| $$ | = PID of this shell script |
| $! | = PID of most recently started background job |

Basic parameter expansions to remember:

| Form | If var is set | If var is not set |
|---|---|---|
| ${var:-string} | $var | string |
| ${var:+string} | string | null |
| ${var:=string} | $var | string (and run var=string) |
| ${var:?string} | $var | (echo string and then exit) |

Here, the colon ':' in all of these operators is actually optional.

- With ':' = operator test for "exist" and "not null".
- Without ':' = operator test for "exist" only.

Basic parameter substitutions to remember:

| Form | Result |
|---|---|
| ${var%suffix} | Remove smallest suffix pattern |
| ${var%%suffix} | Remove largest suffix pattern |
| ${var#prefix} | Remove smallest prefix pattern |
| ${var##prefix} | Remove largest prefix pattern |

### 13.2.4 Shell redirection

Basic redirection to remember (here the [n] is an optional number to specify the file descriptor):

| | |
|---|---|
| [n]> file | Redirect stdout (or n) to file. |
| [n]>> file | Append stdout (or n) to file. |
| [n]< file | Redirect stdin (or n) from file. |
| [n1]>&n2 | Redirect stdout (or n1) to n2. |
| 2> file >&2 | Redirect stdout and stderr to file. |
| > file 2>&1 | Redirect stdout and stderr to file. |
| \| command | Pipe stdout to command. |
| 2>&1 \| command | Pipe stderr and stdout to command. |

Here,

- stdin: standard input (file descriptor = 0)
- stdout: standard output (file descriptor = 1)
- stderr: standard error (file descriptor = 2)

The shell allows you to open files using the exec built-in with an arbitrary file descriptor.

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # open files
$ cat <&3 >&4       # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&-    # close files
$ cat bar
Hello
```

Here n<&- and n>&- mean to close the file descriptor n.

### 13.2.5 Shell conditionals

Each command returns an exit status which can be used for conditional expressions:

- Success: 0 (True)
- Error: 1–255 (False)

Note that the use here of a 0 value to mean "true" differs from the usual convention in some other areas of computing. Also, '[' is the equivalent of the test command, which evaluates its arguments up to ']' as a conditional expression.

Basic conditional idioms to remember are:

```
command && if_success_run_this_command_too || true
command || if_not_success_run_this_command_instead

if [ conditional_expression ]; then
 if_success_run_this_command
else
 if_not_success_run_this_command
fi
```

Here || true was needed to ensure this shell script will not exit at this line accidentally when shell is invoked with -e flag.

File comparison operators in the conditional expression are:

```
-e file       True if file exists.
-d file       True if file exists and is a directory.
-f file       True if file exists and is a regular file.
-w file       True if file exists and is writable.
-x file       True if file exists and is executable.
file1 -nt file2 True if file1 is newer than file2. (modification)
file1 -ot file2 True if file1 is older than file2. (modification)
file1 -ef file2 True if they are the same device and inode numbers.
```

String comparison operators in the conditional expression are:

| | |
|---|---|
| -z str | True if the length of str is zero. |
| -n str | True if the length of str is non-zero. |
| str1 == str2 | True if the strings are equal. |
| str1 = str2 | True if the strings are equal. |
| ("=" should be used in place of "==" for strict POSIX compliance) | |
| str1 != str2 | True if the strings are not equal. |
| str1 < str2 | True if str1 sorts before str2 (locale dependent). |
| str1 > str2 | True if str1 sorts after str2 (locale dependent). |

Arithmetic integer comparison operators in the conditional expression are -eq, -ne, -lt, -le, -gt, and -ge.

### 13.2.6 Command-line processing

The shell processes a script as follows:

- split into tokens by the metacharacters: SPACE, TAB, NEWLINE, ;, (, ), <, >, |, &
- check keyword if not within ”...“ or ’...’ (loop)
- expand alias if not within ”...“ or ’...’ (loop)
- expand brace, a{1,2} -> a1 a2, if not within ”...“ or ’...’
- expand tilde, ~user -> user’s home directory, if not within ”...“ or ’...’
- expand parameter, $PARAMETER, if not within ’...’
- expand command substitution, $(command), if not within ’...’
- split into words with $IFS if not within ”...“ or ’...’
- expand *?[] in pathname if not within ”...“ or ’...’
- look up command
  - function
  - built-in
  - file in $PATH
- loop

Single quotes within double quotes have no effect.

Executing set -x in the shell or invoking the shell with -x option make the shell to print all of commands executed. This is quite handy for debugging.

## 13.3 Awk

References for Awk:

- Effective awk Programming, 3rd edition (O’Reilly)
- Sed & awk, 2nd edition (O’Reilly)
- mawk(1) and gawk(1)
- info gawk

Short program example (creates newusers command entry):

```
#!/usr/bin/awk -f
# Script to create a file suitable for use in the 'newusers' command,
# from a file consisting of user IDs and passwords in the form:
# first_name last_name password
# Copyright (c) KMSelf Sat Aug 25 20:47:38 PDT 2001
# Distributed under GNU GPL v 2, or at your option, any later version.
# This program is distributed WITHOUT ANY WARRANTY.

BEGIN {
    # Assign starting UID, GID
    if ( ARGC > 2 ) {
        startuid = ARGV[1]
        delete ARGV[1]
    }
    else {
        printf( "Usage:  newusers startUID file\n" \
          "  where:\n" \
          "    startUID is the starting userid to add, and\n" \
          "    file is an input file in form:\n" \
          "      first_name last_name password\n" \
        )
        exit
    }

    infile = ARGV[1]
    printf( "Starting UID: %s\n\n", startuid )
}

/^#/ { next }

{
    ++record
    first = $1
    last = $2
    passwd = $3
    user= substr( tolower( first ), 1, 1 ) tolower( last )
    uid = startuid + record - 1
    gid = uid
    printf( "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n",  \
        user, passwd, uid, gid, first, last, user \
        )
}
```

Two packages provide POSIX awk in Debian:

- mawk
    - Priority: required

  - Installed-Size: 228
  - Smaller and much faster – good for default install
  - Compile-time limits exist
    * NF = 32767
    * sprintf buffer = 1020
- gawk
  - Priority: optional
  - Installed-Size: 1708
  - Larger and featureful – many extensions implemented
    * System V Release 4 version of UNIX
    * Bell Labs awk
    * GNU-specific

## 13.4 Perl

This is the interpreter on a Unix-like system.

References for Perl:
- perl(1)
- Programming Perl, 3rd edition (O'Reilly)

Short program example (creates newusers command entry):

```
#!/usr/bin/perl
# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
$pid=1000;
while (<STDIN>) {
        if (/^#/) { next;}
        chop;
        $pid++;
        ($n1, $n2, $n3) = split / /;
        print $n1,"_",$n2,":", $n3, ":",$pid,
                ":",$pid,",,,/home/",$n1,"_",$n2,":/bin/bash\n"
}
```

Install Perl module module_name:

```
# perl -MCPAN -e 'install module_name'
```

## 13.5 Python

It's a nice object-oriented interpreter.

References for Python:

- python(1)
- Learning Python (O'Reilly).

Short program example (creates newusers command entry):

```
#! /usr/bin/env python
import sys, string

# (C) Osamu Aoki Sun Aug 26 16:53:55 UTC 2001 Public Domain
# Ported from awk script by KMSelf Sat Aug 25 20:47:38 PDT 2001
# This program is distributed WITHOUT ANY WARRANTY.

def usages():
    print \
"Usage: ", sys.argv[0], " start_UID [filename]\n" \
"\tstartUID is the starting userid to add.\n" \
"\tfilename is input filename. If not specified, standard input.\n\n" \
"Input file format:\n"\
"\tfirst_name last_name password\n"
            return 1

def parsefile(startuid):
    #
    # main filtering
    #
    uid = startuid
    while 1:
        line = infile.readline()
        if not line:
            break
        if line[0] == '#':
            continue
        (first, last, passwd) = string.split(string.lower(line))
        # above crashes with wrong # of parameters :-)
        user = first[0] + last
        gid = uid
        lineout = "%s:%s:%d:%d:%s %s,,/home/%s:/bin/bash\n" % \
            (user, passwd, uid, gid, first, last, user)
        sys.stdout.write(lineout)
        +uid

if __name__ == '__main__':
    if len(sys.argv) == 1:
        usages()
    else:
        uid = int(sys.argv[1])
```

```
#print "# UID start from: %d\n" % uid
if len(sys.argv) > 1:
    infilename   = string.join(sys.argv[2:])
    infile = open(infilename, 'r')
    #print "# Read file from: %s\n\n" % infilename
else:
    infile = sys.stdin
parsefile(uid)
```

## 13.6 Make

References for Make:

- info make
- make(1)
- Managing Projects with make, 2nd edition (O'Reilly)

Simple automatic variables:

Rule syntax:

```
target: [ prerequisites ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore errors
 [TAB]  @command3 # suppress echoing
```

Here [TAB] is a TAB code. Each line is interpreted by the shell after make variable substitution. Use \ at the end of a line to continue the script. Use $$ to enter $ for environment values for a shell script.

Implicit rules for the target and prerequisites can be written, for example, as:

```
%: %.c header.h
```

or,

```
%.o: %.c header.h
```

Here, the target contains the character % (exactly one of them). The % can match any nonempty substring in the actual target filenames. The prerequisites likewise use % to show how their names relate to the actual target name.

Suffix rules are the obsolete way of defining implicit rules for make. They are still supported in GNU make for compatibility, but use equivalent pattern rules whenever possible:

```
old suffix rule --> new pattern rule
.c:             --> %  : %.c
.c.o:           --> %.o: %.c
```

Automatic variables for the rule:

```
foo.o: new1.c new2.c old1.c new3.c
$@ == foo.o                          (target)
$< == new1.c                         (first one)
$? == new1.c new2.c new3.c           (newer ones)
$^ == new1.c new2.c old1.c new3.c    (all)
$* == `%' matched stem in the target pattern.
```

Variable references:

```
foo1 := bar    # One-time expansion
foo2  = bar    # Recursive expansion
foo3 += bar    # Append
SRCS := $(wildcard *.c)
OBJS := $(foo:c=o)
OBJS := $(foo:%.c=%.o)
OBJS := $(patsubst %.c,%.o,$(foo))
DIRS  = $(dir directory/filename.ext) # Extracts "directory"
 $(notdir NAMES...), $(basename NAMES...), $(suffix NAMES...) ...
```

Run make -p -f/dev/null to see automatic internal rules.

## 13.7 C

Preparation:

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc
```

References for C:

- info libc (C library function reference)
- gcc(1)
- each_C_library_function_name(3)
- Kernighan & Ritchie, The C Programming Language, 2nd edition (Prentice Hall).

### 13.7.1 Simple C program (gcc)

A simple example to compile example.c with a library libm into an executable run_example:

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* prevent buffer overflow */
        y[10] = '\0'; /* fill to make sure string ends with '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

Here, -lm is needed to link library libm for sqrt(). The actual library is in /lib/ with filename libm.so.6, which is a symlink to libm-2.1.3.so.

Look at the last parameter in the output text. There are more than 10 characters even though %10s is specified.

The use of pointer memory operation functions without boundary checks, such as sprintf and strcpy, is deprecated to prevent buffer overflow exploits that leverage the above overrun effects. Instead, use snprintf and strncpy.

### 13.7.2 Debugging

Debugging with gdb

Preparation:

```
# apt-get install gdb
```

References for gdb:
- info gdb (tutorial)

- gdb(1)
- http://www.unknownroad.com/rtfm/gdbtut/gdbtoc.html

Use gdb to debug a program compiled with the -g option. Many commands can be abbreviated. Tab expansion works as in the shell.

```
$ gdb program
(gdb) b 1                # set breakpoint at line 1
(gdb) run arg1 arg2 arg3 # run program
(gdb) next               # next line
...
(gdb) step               # step forward
...
(gdb) p parm             # print parm
...
(gdb) p parm=12          # set value to 12
```

For debugging from within Emacs, refer to 'Editor command summary (Emacs, Vim)' на стр. 209.

Check dependency on libraries

Use ldd to find out a program's dependency on libraries:

```
$ ldd /bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

For ls to work in a chrooted environment, the above libraries must be available in your chrooted environment.

The following commands will also be useful:

- strace: trace system calls and signals
- ltrace: trace library calls

Debugging with memory leak detection tools

There are several memory leak detection tools available in Debian.

- njamd
- valgrind
- dmalloc
- electric-fence

- memprof
- memwatch (not packaged, get this from memwatch (http://directory.fsf.org/devel/debug/memwatch.html).)
- mpatrol
- leaktracer
- libgc6
- Insure++ from Parasoft (http://www.parasoft.com). (non-free, commercial for fee)

Also check out Debugging Tools for Dynamic Storage Allocation and Memory Management (http://www.cs.colorado.edu/homes/zorn/public_html/MallocDebug.html).

### 13.7.3 Flex – a better Lex

flex is a fast lexical analyzer generator.

References for flex:
- info flex (tutorial)
- flex(1)

You need to provide your own main() and yywrap(), or your program.l should look like this to compile without a library (yywrap is a macro; %option main turns on %option noyywrap implicitly):

```
%option main
%%
.|\n    ECHO ;
%%
```

Alternatively, you may compile with the -lfl linker option at the end of your cc command line (like AT&T-Lex with -ll). No %option is needed in this case.

### 13.7.4 Bison – a better Yacc

Several packages provide a Yacc-compatible LALR parser generator in Debian:
- bison: GNU LALR parser generator
- byacc: The Berkeley LALR parser generator
- btyacc: Backtracking parser generator based on byacc

References for bison:
- info bison (tutorial)
- bison(1)

You need to provide your own main() and yyerror(). main() calls yyparse() which calls yylex(), usually created with FleX.

```
%%

%%
```

### 13.7.5 Autoconf

autoconf is a tool for producing shell scripts that automatically configure software source code packages to adapt to many kinds of UNIX-like systems using the entire GNU build system.

autoconf produces the configuration script configure. configure automatically creates a customized Makefile using the Makefile.in template.

#### Compile and install a program

Debian does not touch files in /usr/local (see 'Поддержание многообразия программного обеспечения' на стр. 24). So if you compile a program from source, install it into /usr/local so it will not interfere with Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make
$ make install # this puts the files in the system
```

#### Uninstall program

If you still have the source and if it uses autoconf/automake and if you can remember how you configured it:

```
$ ./configure all-of-the-options-you-gave-it
# make uninstall
```

Alternatively, if you are absolutely sure that the install process puts files only under /usr/local and there is nothing important there, you can erase all its contents by:

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

If you are not sure where files are installed, you should consider using checkinstall, which provides a clean path for the uninstall.

## 13.8 Document preparation

### 13.8.1 roff typesetting

Traditionally, roff is the main Unix text processing system.

See roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7), and info groff.

A good tutorial on -me macros exists. If you have groff (1.18 or newer), find /usr/share/doc /groff/meintro.me.gz and do the following:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
     groff -Tascii -me - | less -R
```

The following will make a completely plain text file:

```
$ zcat /usr/share/doc/groff/meintro.me.gz | \
    GROFF_NO_SGR=1 groff -Tascii -me - | col -b -x > meintro.txt
```

For printing, use PostScript output.

```
$ groff -Tps meintro.txt | lpr
$ groff -Tps meintro.txt | mpage -2 | lpr
```

### 13.8.2 SGML

Preparation:

```
# apt-get install debiandoc-sgml debiandoc-sgml-doc
```

References for debiandoc-sgml:
- /usr/share/doc/debiandoc-sgml-doc
- debiandoc-sgml(1)
- DocBook: The Definitive Guide (/usr/share/doc/docbook-defguide/html/docbook.html), by Walsh and Muellner, (O'Reilly) (package docbook-defguide)

SGML enables management of multiple formats of a document. One easy SGML system is Debiandoc, which is used here. This requires minor conversion from original text files for the following characters:
- ”<“ –> <
- ”>“ –> >
- ” “ –>   (nonbreakable space)
- ”&“ –> &
- ”%“ –> %
- ”(c)“ –> ©
- ”–“ –> –
- ”—“ –> —

To mark a section as a nonprintable comment, enter:

```
<!-- State issue here ... -->
```

To mark a section with a switchable comment, enter:

```
<![ %FIXME; [ State issue here ... ]]>
```

In SGML, the first definition of an entity wins. For example:

```
<!entity % qref "INCLUDE">
<![ %qref; [ <!entity param "Data 1"> ]]>
<!entity param "Data 2">
&param;
```

This ends up as ”Data 1“. If the first line has ”IGNORE“ instead of ”INCLUDE“, this ends up as ”Data 2“ (the second line is a conditional statement). Also, repeating phrases can be defined in advance separately from the context.

```
<!entity whoisthis "my">
Hello &whoisthis; friend.
This is &whoisthis; book.
```

This results in the following:

```
Hello my friend.
This is my book.
```

See the short SGML example sample.sgml in the examples (http://www.debian.org/doc/manuals/debian-reference/examples/).

When SGML documents become bigger, sometimes TeX which is used as the backend text processor may cause errors. See ‘TeX/LaTeX’ на стр. 236.

### 13.8.3 TeX/LaTeX

Preparation:

```
# tasksel # select Miscellaneous --> TeX/LaTeX environment
```

References for LaTeX:

- The teTeX HOWTO: The Linux-teTeX Local Guide (http://www.tldp.org/HOWTO/TeTeX-HOWTO.html)
- tex(1)
- latex(1)
- The TeXbook, by Donald E. Knuth, (Addison-Wesley) [1]

[1] The TeX source of this book is available at ftp://ftp.dante.de/pub/tex/systems/knuth/tex/texbook.tex. ftp://ftp.dante.de/pub/tex/systems/knuth/lib/manmac.tex contains most of the required macros. You can process this document with tex after commenting lines 7 to 10 and adding \input manmac \proofmodefalse. It’s strongly recommended to buy this book (and all other books from Donald E. Knuth) instead of using the online version but the source is a great example of TeX input!

- LaTeX - A Document Preparation System, by Leslie Lamport, (Addison-Wesley)
- The LaTeX Companion, by Goossens, Mittelbach, Samarin, (Addison-Wesley)

This is the most powerful typesetting environment. Many SGML processors use this as their back end text processor. Lyx provided by lyx, lyx-xforms, or lyx-qt package offers nice WYSIWYG editing environment for LaTeX while many use Emacs and Vim as the choice for the source editor.

There are many online resources available:

- teTeX - A Documentation Guide (/usr/share/doc/texmf/newhelpindex.html) (tetex-doc package)
- A Quick Introduction to LaTeX (http://www.msu.edu/user/pfaffben/writings/)
- A Simple Guide to Latex/Lyx (http://www.stat.rice.edu/~helpdesk/howto/lyxguide.html)
- Word Processing Using LaTeX (http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html)
- Local User Guide to teTeX/LaTeX (http://supportweb.cs.bham.ac.uk/documentation/LaTeX/lguide/local-guide/local-guide.html)

When documents become bigger, sometimes TeX may cause errors. You must increase pool size in /etc/texmf/texmf.cnf (or more appropriately edit /etc/texmf/texmf.d/95NonPath and run update-texmf) to fix this.

### 13.8.4 Literate Programming

Instead of writing code containing documentation, the literate programmer writes documentation containing code. This approach ensures a good documentation for a program.

For more on literate-programming, see Literate Programming (http://www.literateprogramming.com/).

Noweb

Preparation:

```
# apt-get install nowebm
```

References for Noweb:

- Noweb — A Simple, Extensible Tool for Literate Programming (http://www.eecs.harvard.edu/~nr/noweb/)
- noweb(1)

This is a WEB-like literate-programming tool which is simpler while providing extensibility and language-independence. [2] When noweb is invoked, it writes the program source code to the output files mentioned in the noweb file, and it writes a TeX file for typeset documentation.

[2] This WEB has nothing to do with World Wide Web. WEB (for PASCAL) and CWEB (for C/C++) are traditional literate-programming tools.

The Debian ifupdown package is a fine example.

```
$ apt-get source ifupdown
$ cd ifupdown*
$ make ifupdown.pdf ifupdown.ps
```

Doxygen

Preparation:

```
# apt-get install doxygen doxygen-doc doxygen-gui
```

References for Doxygen (created by doxygen!):

- Homepage (http://www.doxygen.org/)
- /usr/share/doc/doxygen-doc/html/index.html

It can generate HTML, RTF, Unix manual pages, PostScript, and PDF (using LaTeX) documentation for C++, C, Java, IDL and to some extent PHP and C# programs. Doxygen is compatible to JavaDoc (1.1), Qt-Doc, KDOC and was specifically designed to be used for projects that make use of Troll Tech's Qt (http://www.trolltech.no/qt/) toolkit. It creates include dependency graphs, collaboration diagrams, and graphical class hierarchy graphs even for not documented programs. The output is similar to Qt's documentation.

## 13.9 Packaging

Preparation:

```
# apt-get install debian-policy developers-reference \
                  maint-guide dh-make debhelper
# apt-get install packaging-manual # if Potato
```

References for packaging:

- 'Система управления пакетами в Debian' на стр. 11 (basics)
- Debian New Maintainers' Guide (tutorial)
- dh-make(1)
- Debian Developer's Reference (best practice)
- Debian Policy Manual (authoritative)
- Packaging Manual (Potato)

### 13.9.1 Packaging a single binary

Quick-and-dirty method to Package a single binary per Joey Hess.

```
# mkdir -p mypkg/usr/bin mypkg/DEBIAN
# cp binary mypkg/usr/bin
# cat > mypkg/DEBIAN/control
Package: mypackage
Version: 1
Architecture: i386
Maintainer: Joey Hess <joeyh@debian.org>
Description: my little package
 Don't expect much.
^D
# dpkg-deb -b mypkg
```

### 13.9.2 Packaging with tools

Use dh_make from the dh-make package to create a baseline package. Then, proceed according to instructions in dh-make(1). This uses debhelper in debian/rules.

An older approach is to use deb-make from the debmake package. This uses no debhelper scripts and depends only on the shell.

For examples of multiple-source packages, see "mc" (dpkg-source -x mc_4.5.54.dsc), which uses "sys-build.mk" by Adam Heath (<doogie@debian.org>), and "glibc" (dpkg-source -x glibc_2.2.4-1.dsc), which uses another system by the late Joel Klecker (<espy@debian.org>).

# Глава 14

# GnuPG

References:

- gpg(1).
- /usr/share/doc/gnupg/README.gz
- GNU privacy handbook in /usr/share/doc/gnupg-doc/GNU_Privacy_Handbook/ (install gnupg-doc package)

## 14.1 Installing GnuPG

```
# gpg --gen-key                    # generate a new key
# gpg --gen-revoke my_user_ID      # generate revoke key for my_user_ID
# host -l pgp.net | grep www|less # figure out pgp keyservers
```

As of now, good keyservers are:

```
keyserver wwwkeys.eu.pgp.net
keyserver wwwkeys.pgp.net
```

Here one must be careful not to create more than 2 sub-keys. If you do, keyservers on pgp.net will corrupt your key. Use the newer gnupg (>1.2.1-2) to handle these corrupted subkeys. See http://fortytwo.ch/gpg/subkeys.

Also, only one keyserver can be specified in $HOME/.gnupg/options.

Unfortunately, the following does not work any more:

```
keyserver search.keyserver.net
keyserver pgp.ai.mit.edu
```

## 14.2 Using GnuPG

File handling:

```
$ gpg [options] command [args]
$ gpg {--armor|-a} {--sign|-s} file # sign file into a text file.asc
$ gpg --clearsign file          # clear-sign message
$ gpg --clearsign --not-dash-escaped patchfile # clear-sign patchfile
$ gpg --verify file             # verify clear-signed file
$ gpg -o file.sig {-b|--detach-sig} file # create detached signature
$ gpg --verify file.sig file    # verify file with file.sig
$ gpg -o crypt_file {--recipient|-r} name {--encrypt|-e} file
      # public-key encryption intended for name
$ gpg -o crypt_file {--symmetric|-c} file # symmetric encryption
$ gpg -o file --decrypt crypt_file # decryption
```

## 14.3 Managing GnuPG

Key management:

```
$ gpg --edit-key user_ID          # "help" for help, interactive
$ gpg -o file --exports           # export all keys to file
$ gpg --imports file              # import all keys from file
$ gpg --send-keys user_ID         # send key of user_ID to keyserver
$ gpg --recv-keys user_ID         # recv. key of user_ID from keyserver
$ gpg --list-keys user_ID         # list keys of user_ID
$ gpg --list-sigs user_ID         # list sig. of user_ID
$ gpg --check-sigs user_ID        # check sig. of user_ID
$ gpg --fingerprint user_ID       # check fingerprint of user_ID
$ gpg --list-sigs | grep '^sig' | grep '[User id not found]' \
  | awk '{print $2}' | sort -u | xargs gpg --recv-keys # get unknown keys
  # update keys for all unknown sigs.
```

Trust code:

| | |
|---|---|
| - | No ownertrust assigned / not yet calculated. |
| e | Trust calculation has failed. |
| q | Not enough information for calculation. |
| n | Never trust this key. |
| m | Marginally trusted. |
| f | Fully trusted. |
| u | Ultimately trusted. |

The following will upload my key ”A8061F32“ to multiple keyservers:

```
$ for xx in us es cz de dk uk ch net.uk earth.net.uk; \
$ do gpg --keyserver wwwkeys.$xx.pgp.net  --send-keys A8061F32; done
```

## 14.4 Using GnuPG with applications

### 14.4.1 Using GnuPG with Mutt

Add the following to ~/.muttrc to keep a slow GnuPG from automatically starting, while allowing it to be used by typing ‘S’ at the index menu.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

### 14.4.2 Using GnuPG with Vim

Add the contents of _vimrc obtained from the examples subdirectory (http://www.debian.org/doc/manuals/debian-reference/examples/) into ~/.vimrc to run GnuPG transparently.

# Глава 15

# Support for Debian

The following resources provide help, advice, and support for Debian. Try your best to use self-help resources before crying out loud in the mailing lists. :)

Note that you can access a lot of documentation on your system by using a WWW browser, via the dwww or dhelp commands, found in their respective packages.

## 15.1 References

The following references are available for Debian and Linux in general. If their contents conflict with each other, always rely more on primary information sources than on secondary ones such as this document.

- Installation Manual (primary)
  - Read before installation and upgrade.
  - Web: http://www.debian.org/releases/stable/installmanual
  - Web: http://www.debian.org/releases/testing/installmanual (work in progress, sometimes this may not exist)
  - Package: Not available in install-doc: Bug#155374
  - File: DebianCDunder/doc/
- Release Notes (primary)
  - A must-read before installation and upgrade even if you are experienced.
  - Web: http://www.debian.org/releases/stable/releasenotes
  - Web: http://www.debian.org/releases/testing/releasenotes (work in progress, sometimes this may not exist)
  - Package: Not available in install-doc: Bug#155374
  - File: DebianCDunder/doc/
- FAQ (secondary)
  - Frequently asked questions
  - Web: http://www.debian.org/doc/manuals/debian-faq/

  - Package: doc-debian
  - File: /usr/share/doc/debian/FAQ/index.html

- Debian Reference (secondary)
  - Most comprehensive post-install user manual
  - Web: http://www.debian.org/doc/manuals/debian-reference/
  - Package: debian-reference-en
  - File: /usr/share/doc/Debian/reference/

- APT HOWTO (secondary)
  - Detailed user guide for Debian package management. (Woody)
  - Web: http://www.debian.org/doc/manuals/apt-howto/
  - Package: apt-howto
  - File: /usr/share/doc/Debian/apt-howto/

- Securing Debian Manual (secondary)
  - Detailed user guide for securing and hardening of the default Debian installation. (Woody)
  - Web: http://www.debian.org/doc/manuals/securing-debian-howto/
  - Package: harden-doc
  - File: /usr/share/doc/harden-doc/html/securing-debian-howto/

- dselect Documentation for Beginners (secondary)
  - Tutorial for dselect
  - Web: http://www.debian.org/releases/woody/i386/dselect-beginner
  - Package: Not available in install-doc: Bug#155374
  - File: DebianCDunder/doc/

- Debian Policy Manual (primary)
  - Technical backbone of Debian.
  - Web: http://www.debian.org/doc/debian-policy/
  - Package: debian-policy
  - File: /usr/share/doc/debian-policy/

- Debian Developer's Reference (primary)
  - Basic knowledge for developers.
  - The rest of us should also browse this once.
  - Web: http://www.debian.org/doc/manuals/developers-reference/
  - Package: developers-reference
  - File: /usr/share/doc/developers-reference/

- Debian New Maintainers' Guide (primary)
  - Practical guide for developers.
  - Packaging tutorials for the rest of us.
  - Web: http://www.debian.org/doc/manuals/maint-guide/
  - Package: maint-guide
  - File: /usr/share/doc/maint-guide/

- Packaging Manual (Potato)
  - packaging-manual package in Potato. (Moved into appendix of Developer's Reference.)
- Unix-style manual pages (primary)
  - man package-name
- GNU-style info pages (primary)
  - info package-name
- Package-specific documents (primary)
  - Find them under /usr/share/doc/package-name
- LDP: Linux Documentation Project (secondary)
  - General Linux HOWTOs and mini-HOWTOs
  - Web: http://www.tldp.org/
  - Package: doc-linux-text
  - File: /usr/share/doc/HOWTO/
- DDP: Debian Documentation Project (secondary)
  - Debian-specific manuals
  - Web: http://www.debian.org/doc/
- Debian Developers' Corner (secondary)
  - Key information for Debian developers
  - Insightful for end users
  - Web: http://www.debian.org/devel/
- Source code (absolutely primary)
  - No one can argue with this :-)
  - Download source code following 'Исходный код' на стр. 11

The following references are available for Unix in general. Please note that there are some minor differences between different Unix systems. Device names and init methods need extra attention.

- The UNIX Programming Environment
  - The book to read to learn about how UNIX works.
  - By B. W. Kernighan and R. Pike
  - Published by Princeton Hall Software Series
- The C Programming Language (second edition)
  - The book to read to learn about ANSI C.
  - By B. W. Kernighan and D. M. Ritchie
  - Published by Princeton Hall Software Series
- UNIX Power Tools
  - The book to read to learn Unix tips.

  - By Jerry Peek, Tim O'Reilly, and Mike Loukides
  - Published by O'Reilly and Associates

- Essential System Administration (second edition)
  - The book to read to learn about Unix system administration for many Unix flavors.
  - By Aeleen Frisch
  - Published by O'Reilly and Associates

- Bell Labs: Computing Sciences Research
  - Rich archive of Unix history
  - Main: http://cm.bell-labs.com/cm/cs/
  - Selected technical reports: http://cm.bell-labs.com/cm/cs/cstr.html
  - Some papers: http://cm.bell-labs.com/cm/cs/papers.html

- Online Linux general support resources
  - Debian Planet (http://www.debianplanet.org/)
  - debianHELP (http://www.debianhelp.org/)
  - Linux.com (http://linux.com/)
  - The Linux Home Page at Linux Online (http://www.linux.org/)
  - Red Hat (commercial Linux vender) (http://www.redhat.com/) (RPM, Sys-V init)
  - SuSE, Inc. (commercial Linux vender) (http://www.suse.de/) (RPM, Sys-V init)
  - Slackware (http://www.slackware.com/) (TGZ, BSD-style init)

- Online general Unix guide resources
  - A UNIX Introductory Course from Ohio State University (http://www-wks.acs.ohio-state.edu/unix_course/unix.html)
  - UNIXhelp from The University of Edinburgh (http://unixhelp.ed.ac.uk/)
  - Unix / Programming Information (http://arioch.unomaha.edu/~jclark/#info)
  - comp.unix.questions FAQ (http://www.faqs.org/faqs/unix-faq/faq/)
  - comp.unix.user-friendly FAQ (http://www.camelcity.com/~noel/usenet/cuuf-FAQ.htm)
  - FreeBSD Documentation (http://www.freebsd.org/docs.html)
  - The FreeBSD Handbook (http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html)
  - UNIX GUIDE (http://ieee.uow.edu.au/documents/)
  - The Unix Heritage Society (http://www.tuhs.org/)

- Free software project home pages
  - GNU Project (http://www.gnu.org/)
  - The Linux Documentation Project (http://www.tldp.org/)
  - The Linux Kernel Archives (http://www.linux.org/)
  - The XFree86 Project, Inc (http://www.xfree86.org/)
  - GNOME (http://www.gnome.org/)
  - K Desktop Environment (http://www.kde.org/)
  - GNU software at Red Hat (http://sources.redhat.com/)
  - Mozilla (http://www.mozilla.org)
  - FreeBSD (http://www.freebsd.org/)

– OpenBSD (http://www.openbsd.org/)
– NetBSD (http://www.netbsd.org/)

## 15.2 Finding the meaning of a word

Many words used in Debian are cryptic jargon or acronyms. The following will solve most questions:

```
$ dict put-a-weird-word-here
```

## 15.3 Finding the popularity of a Debian package

Many packages exist in Debian and it is sometimes difficult to know which one to try first. See Debian Popularity Contest Results (http://popcon.debian.org/) to get insight into what others are using. Also install the popularity-contest package to contribute.

## 15.4 The Debian bug tracking system

The Debian distribution has a bug tracking system (BTS) (http://bugs.debian.org/) which files details of bugs reported by users and developers. Each bug is given a number, and is kept on file until it is marked as having been dealt with.

You should check to see whether your bug report has already been filed by someone else before submitting it. Lists of currently outstanding bugs are available on the World Wide Web (http://bugs.debian.org/) and elsewhere (http://www.debian.org/Bugs/Access). See also 'Программные ошибки в Debian и поддержка' на стр. 88.

There may be many release-critical bug reports marked with FTBFS. This means "Fails To Build From Source".

Instructions for reporting a bug are given at http://www.debian.org/Bugs/Reporting.

## 15.5 Mailing lists

Read at least "debian-devel-announce" (English, read-only and low-traffic) to stay current with Debian.

The mailing lists of most interest to Debian users are "debian-user" (English, open and high-traffic) and other "debian-user-language" lists (for other languages).

For information on these lists and details of how to subscribe, see http://lists.debian.org/. Please check the archives for answers to your question prior to posting and also adhere to standard list etiquette.

If you do not wish to get CCed for the reply to your mailing list posting, use the Mail-Followup-To: header which is a very effective measure. This is the informal convention of mailing lists as explained in http://cr.yp.to/proto/replyto.html.

## 15.6 Internet Relay Chat (IRC)

IRC (Internet Relay Chat) is a way to chat with people from all over the world in real time. IRC channels dedicated to Debian can be found on the freenode (http://www.freenode.info/) IRC network. To connect, you need an IRC client. Some of the most popular clients are XChat, BitchX, ircII, irssi, epic4 and KSirc, all of which have been packaged for Debian. Once you have the client installed, you need to tell it to connect to the server. In most clients, you can do that by typing:

```
/server irc.debian.org
```

Once you are connected, join channel #debian by typing

```
/join #debian
```

To leave channel #debian type

```
/part #debian
```

You can quit the irc client by typing

```
/quit
```

To send a private message ”Hello Mr. Foo“ to foo type

```
/msg foo Hello Mr. Foo
```

Note that anything you type without the preceding / is sent to the channel as a message.

Note: clients like XChat often have a different graphical user interface for joining servers/channels.

## 15.7 Search engines

There are many search engines that serve documentation related to Debian:

- Debian WWW search site (http://search.debian.org/).

- Google (http://www.google.com/): include "site:debian.org" as a search term.
- Google Groups (http://groups.google.com/): a search engine for newsgroups. Include "group:linux.debian.*" as a search term.
- AltaVista (http://www.altavista.com/)

For example, searching on the string "cgi-perl" gives a more detailed explanation of this package than the brief description field in its control file. See 'Программные ошибки в Debian и поддержка' на стр. 88 for related advice.

## 15.8 Websites

The following are a few random URLs I collected for specific issues.

- IBM developerWorks: Linux (http://www.ibm.com/developerworks/linux/)
- Adrian Bunk's latest packages (back port to stable) (http://www.fs.tum.de/~bunk/)
- Linux on Laptops (http://www.linux-laptop.net/)
- Xterm FAQ (http://dickey.his.com/xterm/xterm.faq.html)
- EXT3 File System mini-HOWTO (http://www.zip.com.au/~akpm/linux/ext3/ext3-usage.html)
- Large File Support in Linux (http://www.suse.de/~aj/linux_lfs.html)
- Window Managers for X (http://www.xwinman.org)
- Linux USB Project (http://www.linux-usb.org/)
- SuSE pages for CJK (http://www.suse.de/~mfabian/suse-cjk/suse-cjk.html)
- LNX-BBC (Business-card-sized boot CD project) (http://www.lnx-bbc.org/)
- Linux info by Karsten Self (partitioning, backup, browsers...) (http://kmself.home.netcom.com/Linux/index.html)
- Backup info HOWTO by Alvin Oga (http://www.Linux-Backup.net/)
- Security info HOWTO by Alvin Oga (http://www.Linux-Sec.net/)
- Various UNOFFICIAL sources for APT (http://www.apt-get.org/)
- Laptop Ethernet Configuration (http://www.orthogony.com/gjw/lap/lap-ether-intro.html)

# Приложение A

# Appendix

## A.1 Authors

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)
  - Rénald 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 <andrelop\#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.1st.ne.jp> (leader: ja)

QREF was short for the original document title, "Quick Reference…" and also is the project name at qref.sourceforge.net.

Many manual pages and info pages on the Debian system were used as the primary references to write this document. To the extent Osamu Aoki considered within the fair quotation stature, many parts of them, especially command definitions, were used as phrase pieces after careful editorial efforts to fit them into the style and the objective of this document.

Most of the contents of 'Основы Debian' на стр. 5 originally came from "The Debian FAQ" (March 2002):

- 5. The Debian FTP archives: ftparchives.sgml (entire chapter)
- 6. Basics of the Debian Package Management System: pkg_basics.sgml (entire chapter)
- 7. The Debian Package Management Tools: pkgtools.sgml (entire chapter)
- 8. Keeping Your Debian System Up To Date: uptodate.sgml (entire chapter)
- 9. Debian and the kernel: kernel.sgml (entire chapter)
- 10. Customizing your installation of Debian GNU/Linux: customizing.sgml (part of chapter)

These sections of "The Debian FAQ" were included in this document after major reorganization to reflect changes in the Debian system. Both documents are updated concurrently now.

The original "Debian FAQ" was made and maintained by J. H. M. Dassen (Ray) and Chuck Stickelman. Authors of the rewritten "Debian FAQ" are Susan G. Kleinmann and Sven Rudolph. After them, "The Debian FAQ" was maintained by Santiago Vila. The current maintainer is Josip Rodin.

Parts of the information for "The Debian FAQ" came from:

- The Debian-1.1 release announcement, by Bruce Perens (http://www.perens.com/).
- The Linux FAQ, by Ian Jackson (http://www.chiark.greenend.org.uk/~ijackson/).
- Debian Mailing List Archives (http://lists.debian.org/),
- the dpkg programmers' manual and the Debian Policy manual (see 'References' на стр. 245)
- many developers, volunteers, and beta testers, and
- the flaky memories of its authors. :-)

Some parts of "Tutorial" section were derived from

- "Debian Tutorial" by Havoc Pennington, Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer, and Ivan E. Moore II. (This document was derived from "Linux User's Guide" by Larry Greenfield.)
- "Debian GNU/Linux: Guide to Installation and Usage" by John Goerzen and Ossama Othman.

The authors would like to thank all those who helped make this document possible.

## 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.

Although I used to live in the USA, I am a non-native English user. Any grammatical corrections are welcomed.

The best feedback is a patch against the SGML version, but a patch against the text version is also welcomed. See 'Официальный документ' на стр. 1 for the official document site.

The original SGML files used to create this document are also available in CVS at: :pserver:anonymous@cvs.sf.net/cvsroot/qref or http://qref.sourceforge.net/Debian/qref.tar.gz.

## A.4 Document format

This document was written using the DebianDoc SGML DTD (rewritten from LinuxDoc SGML). The DebianDoc SGML system enables us to create files in a variety of formats from one source, e.g. this document can be viewed as HTML, plain text, TeX DVI, PostScript, PDF, and GNU info.

Conversion utilities for DebianDoc SGML are available in the Debian package debiandoc-sgml.

## A.5 The Debian maze

The Linux system is a very powerful computing platform for a networked computer. However, learning how to use all its capabilities is not easy. Setting up the printer is a good example.

There is a complete, detailed map called the "SOURCE CODE". This is very accurate but very hard to understand. There are also references called HOWTO and mini-HOWTO. They are easier to understand but tend to give too much detail and lose the big picture. I sometimes have a problem finding the right section in a long HOWTO when I need a few commands to invoke.

In order to navigate through this maze of Linux system configuration, I started writing down simple reminder memos in text file format as my quick reference. This list of memos grew larger and I learned debiandoc in the meantime. The product is this Справочник по Debian.

## A.6 The Debian quotes

Here are some interesting quotes from the Debian mailing list.

- "This is Unix. It gives you enough rope to hang yourself." —Miquel van Smoorenburg <miquels@cistron.nl>
- "Unix IS user friendly… It's just selective about who its friends are." —Tollef Fog Heen <tollef@add.no>