

Часто задаваемые вопросы о Debian GNU/Linux

14 февраля 2026 г.

Часто задаваемые вопросы о Debian GNU/Linux

Published October 2024

Copyright © 1996-2024 Software in the Public Interest

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this document into another language, under the above conditions for modified versions, except that this permission notice may be included in translations approved by the Free Software Foundation instead of in the original English.

Оглавление

1	Определения и краткий обзор	1
1.1	О чём данные ЧаВо?	1
1.2	Что такое Debian GNU/Linux?	1
1.3	Я понял, что такое Debian, а что такое Linux?!	2
1.4	Debian работает только с GNU/Linux?	2
1.5	Чем Debian GNU/Linux отличается от других дистрибутивов Linux? Почему я должен отдать предпочтение Debian, а не какому-то другому дистрибутиву?	2
1.6	Как соотносятся проект Debian и проект GNU Free Software Foundation?	3
1.7	Как произносится слово Debian и что оно означает?	3
2	Получение и установка Debian GNU/Linux	5
2.1	Какая версия Debian является последней?	5
2.2	Обновляются ли пакеты в «stable»?	5
2.3	Где и как получить установочные образы Debian?	6
2.4	Как установить Debian с компакт-дисков?	6
2.5	Зачем на CD официального стабильного дистрибутива нужны символьные ссылки «frozen» и «unstable»? Я думал, что на этих CD только стабильный дистрибутив!	6
2.6	Можно ли установить Debian прямо с Интернет-сайта?	7
2.7	Есть ли другие способы запуска программы установки системы?	7
3	Выбор дистрибутива Debian	9
3.1	Какой дистрибутив Debian (стабильный/тестируемый/нестабильный) лучше всего мне подойдёт?	9
3.1.1	Вы предлагаете установить стабильный дистрибутив, но при его использовании не обнаруживается или не работает такое-то аппаратное обеспечение. Что делать?	9
3.1.2	Есть ли разница между версиями пакетов в различных дистрибутивах?	10
3.1.3	В стабильных дистрибутивах содержатся устаревшие версии программ. Только взгляните на Kde, Gnome, Xorg или даже ядро. Они очень старые. Почему?	10
3.1.4	Возможно ли позже перейти на другой дистрибутив и как это сделать?	10
3.1.5	Не могли бы вы подсказать мне какой выпуск следует устанавливать, стабильный, тестируемый или нестабильный?	10
3.1.6	Вы упомянули, что тестируемый дистрибутив иногда ломается. Что имеется в виду?	11
3.1.7	Почему тестируемый выпуск может быть сломан в течение нескольких месяцев? Разве исправления, добавляемые в нестабильный выпуск, не переходят в тестируемый?	11
3.1.8	С точки зрения администратора, какой дистрибутив требует большего внимания?	12
3.1.9	Что происходит при выходе новой версии дистрибутива?	12
3.1.10	У меня на настольном компьютере/кластере установлен Debian. Как узнать, какой дистрибутив используется?	13
3.1.11	I am currently using the stable version. Can I change to testing or unstable? If so, how?	13
3.1.12	Сейчас я использую тестируемый дистрибутив (forky). Что произойдёт после выпуска следующей версии? У меня по-прежнему будет отслеживаться тестируемый дистрибутив, или на моей машине будет новый стабильный дистрибутив?	14
3.1.13	Всё равно непонятно. Так что же нужно устанавливать?	14
3.2	But what about Kali, Knoppix, Linux Mint, Ubuntu, and others?	14
3.2.1	I know that Kali/Knoppix/Linux Mint/Ubuntu/... is Debian-based. So after installing it on the hard disk, can I use 'apt' package tools on it?	14
3.2.2	I installed Kali/Knoppix/Linux Mint/Ubuntu/... on my hard disk. Now I have a problem. What should I do?	15

3.2.3	I'm using Kali/Knoppix/Linux Mint/Ubuntu/... and now I want to use Debian. How do I migrate?	15
4	Проблемы совместимости	17
4.1	На каких архитектурах/системах работает Debian GNU/Linux?	17
4.2	На каких ядрах работает Debian GNU/Linux?	18
4.3	Насколько Debian совместим с другими дистрибутивами Linux?	18
4.4	Насколько исходный код Debian совместим с другими системами Unix?	18
4.5	Можно ли использовать пакеты Debian (файлы «.deb») в системе RedHat/Slackware/... Linux? Можно ли использовать пакеты RedHat (файлы «.rpm») в системе Debian GNU/Linux?	19
4.6	Как установить не-Debian программу?	19
5	Программное обеспечение, доступное в системе Debian	21
5.1	Какие виды приложений и средств разработки программ доступны в Debian GNU/Linux?	21
5.2	Кто написал все эти программы?	21
5.3	Как получить текущий список программ, для которых в Debian собраны пакеты?	22
5.4	Как установить окружение разработчика для сборки пакетов?	22
5.5	Чего не хватает в Debian GNU/Linux?	22
5.6	Почему иногда выдаются сообщения «ld: cannot find -lfoo» при компиляции программ? Почему не хватает каких-то файлов libfoo.so в пакетах Debian с библиотеками?	22
5.7	Поддерживает ли Debian Java (и если да, то как)?	22
5.8	How can I check that I am using a Debian system, and what version it is?	23
5.9	Как в Debian поддерживаются языки, отличные от английского?	23
5.10	I have a device that requires non-free firmware. What should I do?	24
5.11	У меня есть беспроводная сетевая карта, которая не работает в Linux. Что делать?	24
6	The Debian archives	25
6.1	Сколько существует дистрибутивов Debian?	25
6.2	Что означают названия вроде etch, lenny и т. д.?	25
6.2.1	Какие кодовые имена уже использовались?	25
6.2.2	Откуда берутся эти кодовые имена?	25
6.3	Что такое «sid»?	26
6.4	Что содержится в каталоге stable?	26
6.5	Что содержится в тестируемом дистрибутиве?	27
6.5.1	Что происходит с «testing»? Как его «замораживают»?	27
6.6	Что содержится в нестабильном дистрибутиве?	27
6.7	What are all those directories at the Debian archives?	28
6.8	Что находится в каталогах внутри dists/stable/main?	28
6.9	Где находятся исходные коды?	28
6.10	Что находится в каталоге pool?	29
6.11	Для чего нужен каталог «incoming»?	29
6.12	Where can I find old packages of previous releases?	29
6.13	Как создать свой собственный apt-репозиторий?	29
7	Основы системы управления пакетами Debian	31
7.1	Что такое пакет Debian?	31
7.2	Какой формат у двоичных пакетов Debian?	32
7.3	Почему имена файлов пакетов Debian такие длинные?	32
7.4	Зачем нужен файл control?	32
7.5	Зачем нужен файл conffile?	33
7.6	Зачем нужны сценарии preinst, postinst, prerm и postrm?	33
7.7	Что такое <i>Пакет первой необходимости (Essential)</i> , <i>Необходимый (Required)</i> , <i>Важный (Important)</i> , <i>Стандартный (Standard)</i> , <i>Необязательный (Optional)</i> или <i>Дополнительный (Extra)</i> пакет?	34
7.8	Что такое виртуальный пакет?	34
7.9	Что имеется в виду, когда говорят, что пакет <i>Рекомендует (Recommends)</i> , <i>Предлагает (Suggests)</i> , <i>Заменяет (Replaces)</i> , <i>Ломает (Breaks)</i> или <i>Предоставляет (Provides)</i> другой пакет, <i>Зависит (Depends)</i> от него или <i>Конфликтует (Conflicts)</i> с ним?	35
7.10	Что значит Пред-зависит (Pre-Depends)?	35

7.11	Что означают слова <i>неизвестно</i> (<i>unknown</i>), <i>установить</i> (<i>install</i>), <i>удалить</i> (<i>remove</i>), <i>вычистить</i> (<i>purge</i>), <i>зафиксировать</i> (<i>hold</i>) в строке состояния пакета?	36
7.12	Как зафиксировать (<i>hold</i>) пакет?	36
7.13	Как установить пакет исходного кода?	37
7.14	Как собрать двоичный пакет из пакета исходного кода?	37
7.15	Как самому создать пакет Debian?	38
8	Инструменты управления пакетами Debian	39
8.1	Какие программы для управления пакетами имеются в Debian?	39
8.1.1	dpkg	39
8.1.2	APT	40
8.1.3	aptitude	41
8.1.4	synaptic	42
8.1.5	tasksel	42
8.1.6	Другие инструменты управления пакетами	42
8.1.6.1	dpkg-deb	42
8.2	Говорят, что Debian способен обновить работающую программу; как это делается?	43
8.3	Как узнать, какие пакеты установлены в системе Debian?	43
8.4	How do I display the files of an installed package?	43
8.5	Как определить пакет, которому принадлежит определённый файл?	43
8.6	Why is "foo-data" not removed when I uninstall "foo"? How do I make sure old unused library-packages get purged?	44
9	Поддержание обновлений системы Debian	45
9.1	Как поддерживать систему Debian в актуальном состоянии?	45
9.1.1	aptitude	45
9.1.2	apt-get и apt-cdrom	46
9.2	Нужно ли для обновления пакета переходить в однопользовательский режим?	46
9.3	Нужно ли хранить на диске все файлы-архивы .deb?	46
9.4	How can I keep a log of the packages I added to the system? I'd like to know when upgrades and removals have occurred and on which packages!	47
9.5	Возможно ли автоматическое обновление системы?	47
9.6	Есть несколько машин, как скачивать для них обновления только один раз?	47
10	Debian и ядро	49
10.1	Можно ли установить и собрать ядро без некоторых специфичных для Debian настроек?	49
10.2	Какие инструменты для самостоятельной сборки ядра есть в Debian?	49
10.3	What special provisions does Debian provide to deal with modules?	49
10.4	Можно ли безопасно удалить пакет со старой версией ядра и как это сделать?	49
10.5	Where can I get more information about Linux packages for Debian?	50
11	Пользовательская настройка системы Debian GNU/Linux	51
11.1	Как убедиться, что все программы используют одинаковый формат бумаги?	51
11.2	Как обеспечить доступ к файлам устройств без нарушения защиты системы?	51
11.3	Как в Debian правильно сменить консольный шрифт при запуске системы?	51
11.4	Как настроить параметры по умолчанию для приложений X11?	51
11.5	How does a Debian system boot?	52
11.6	And how about Debian and traditional System V init?	52
11.7	And are there yet other ways of booting a Debian system?	53
11.8	Как система управления пакетами работает с пакетами, содержащими настроечные файлы для других пакетов?	53
11.9	Как заменить программу, установленную из какого-либо пакета, на свою собственную?	54
11.10	Как включить самостоятельно собранный пакет в список доступных пакетов, известных системе управления пакетами?	54
11.11	Некоторым пользователям нравится mawk, другим gawk; некоторым vim, другим elvis; некоторым trn, другим tin. Как осуществляется поддержка предпочтений в Debian?	55

12	Поддержка Debian GNU/Linux	57
12.1	Какая ещё документация существует по системе Debian?	57
12.2	Есть ли какие-нибудь онлайн-ресурсы для обсуждения Debian?	58
12.2.1	Списки рассылки	58
12.2.1.1	Как вести себя в списках рассылки?	58
12.2.2	Web forum	59
12.2.3	Вики	59
12.2.4	Сопровождающие	59
12.2.5	Группы новостей Usenet	59
12.3	Как быстро найти информацию по Debian GNU/Linux?	59
12.4	Существуют ли журналы известных ошибок?	59
12.5	Как сообщить об ошибке в Debian?	60
13	Помощь проекту Debian	61
13.1	How can I become a Debian member/Debian developer?	61
13.2	Как помочь проекту Debian своими ресурсами?	61
13.3	Как оказать финансовую помощь проекту Debian?	62
13.3.1	Software in the Public Interest	62
13.3.2	Другие организации	62
14	Распространение Debian GNU/Linux в коммерческом продукте	63
14.1	Можно ли штамповать CD-диски Debian и продавать их?	63
14.2	Можно ли укомплектовать Debian несвободным ПО?	63
14.3	Я делаю специальный дистрибутив Linux для «вертикального маркетинга». Могу я использовать Debian GNU/Linux как основу этой системы и добавлять мои собственные приложения поверх него?	63
14.4	Можно ли создать «пакет» Debian из коммерческой программы, чтобы её можно было без проблем установить в любой системе Debian?	64
15	Изменения, ожидаемые в следующем выпуске Debian	65
15.1	Улучшение безопасности системы	65
15.2	Расширенная поддержка не-англоговорящих пользователей	65
15.3	Улучшения в программе установки Debian	66
15.4	Ещё больше архитектур	66
15.5	Больше ядер	66
16	Общая информация о ЧаВо	67
16.1	Авторы	67
16.2	Обратная связь	67
16.3	Где взять	67
16.4	Формат документа	68

Аннотация

В этом документе содержатся ответы на часто задаваемые вопросы о Debian GNU/Linux.

Глава 1

Определения и краткий обзор

1.1 О чём данные ЧаВо?

Здесь собраны вопросы (с ответами!) о дистрибутиве Debian (Debian GNU/Linux и других) и о проекте Debian. Там, где есть другая документация по определённому вопросу, даётся ссылка: мы не хотим цитировать здесь большие части сторонней документации. В ряде ответов предполагается наличие некоторых знаний по Unix-подобным операционным системам. Однако допуская, что знаний у читателя может быть совсем немного, мы постараемся отвечать на общие вопросы новичков как можно проще.

Если в этих ЧаВо вы не нашли то, что искали, см. Раздел [12.1](#). Если даже это не помогло, см. Раздел [16.2](#).

1.2 Что такое Debian GNU/Linux?

Debian GNU/Linux — это один из *дистрибутивов* операционной системы Linux с большим количеством пакетов.

Debian GNU/Linux:

- **самодостаточен:** сейчас в Debian входит более 64961 пакетов программ. Пользователи могут выбирать нужные пакеты для установки; для этого в Debian есть специальный инструмент. Список и описания входящих в Debian пакетов можно найти на любом из [серверов-зеркал](https://www.debian.org/mirror/list) (<https://www.debian.org/mirror/list>) Debian.
- **свободен для использования и дальнейшего распространения:** для участия в его распространении и разработке не требуется членства или оплаты. Все пакеты, которые формально являются частью Debian GNU/Linux, свободны для дальнейшего распространения обычно на условиях универсальной общественной лицензии GNU (GNU General Public License).
The Debian archives also carry approximately 1082 software packages (in the `non-free` and `contrib` sections), which are distributable under specific terms included with each package.
- **dynamic:** With about 1400 volunteers constantly contributing new and improved code, Debian is evolving rapidly. The archives are updated twice every day.

Большинство пользователей Linux работают с одним из *дистрибутивов* Linux, таким же как и Debian GNU/Linux. В принципе, пользователь может взять ядро Linux из Интернет или ещё откуда-нибудь и собрать его сам. Таким же образом он может найти исходный код многих приложений, собрать программы и установить их на своей системе. В случае сложных приложений это не только может отнять много времени, но и чревато ошибками. Чтобы избежать этого, пользователи часто используют операционную систему и пакеты приложений, предлагаемые одним из распространителей Linux. Разные распространители (дистрибьюторы) Linux предлагают разный набор программного обеспечения (ПО), протоколы и методы упаковки, установки и сопровождения пакетов приложений в пользовательских системах, включая инструментарий для установки и управления, документацию и другие вещи.

Debian GNU/Linux — это результат усилий добровольцев по созданию свободной, высококачественной Unix-совместимой операционной системы с полным набором приложений. Идея свободной Unix-подобной системы исходит от проекта GNU, и многие приложения Debian GNU/Linux, сделавшие дистрибутив таким удобным, были разработаны в рамках проекта GNU.

В Debian термин *свободное* используется в значении GNU (см. [Критерии Debian по определению Свободного ПО](https://www.debian.org/social_contract#guidelines) (https://www.debian.org/social_contract#guidelines)). Когда мы говорим о свободном ПО, мы имеем в виду свободу, а не цену. Свободное ПО предполагает, что у вас имеется свобода распространять копии свободного ПО, что вы получаете не только двоичные файлы, но и исходный код, либо можете получить его, если вам это нужно, что вы можете изменять ПО или использовать части этого ПО в новых свободных программах; и что вы знаете, что можете всё это делать.

Проект Debian был создан Иэном Мёрдоком (Ian Murdock) в 1993 году при спонсорском участии проекта GNU Free Software Foundation. Сегодня, разработчики Debian считают его прямым наследником проекта GNU.

Хотя Debian GNU/Linux и сам по себе является свободным ПО, на его основе можно строить и другие дистрибутивы Linux. Предлагая надёжную, самодостаточную основу системы, Debian предоставляет пользователям Linux улучшенную совместимость, и позволяет создателям дистрибутивов Linux избежать двойной работы и сфокусироваться на вещах, отличающих их собственные дистрибутивы от прочих. Подробности об этом см. в Раздел [14.3](#).

1.3 Я понял, что такое Debian, а что такое Linux?!

В двух словах, Linux — это ядро Unix-подобной операционной системы. Исходно оно разрабатывалось для машин с процессорами 386 (и более новых), а сейчас может работать и на десятке других архитектур. Ядро Linux было написано Линусом Торвальдсом (Linus Torvalds) и многими компьютерщиками со всего мира.

Кроме ядра, в «Linux»-систему обычно входит:

- файловая система, соответствующая стандарту иерархии файловой системы Linux (Linux Filesystem Hierarchy Standard), <https://www.pathname.com/fhs>.
- разнообразные Unix-утилиты, многие из которых были разработаны проектом GNU и Free Software Foundation.

Ядро Linux, файловая система, утилиты GNU, FSF и другие разрабатываются в соответствии со стандартом POSIX (IEEE 1003.1), см. Раздел [4.4](#).

Более подробную информацию о том, [Что такое Linux](https://www.linux.org/info/) (<https://www.linux.org/info/>), см. на сайте [Linux Online](https://www.linux.org/info/) (<https://www.linux.org/info/>).

1.4 Debian работает только с GNU/Linux?

Currently, Debian is only available for Linux, but with Debian GNU/Hurd we are offering a non-Linux-based OS as a development, server and desktop platform, too. However, this non-linux port is not officially released yet.

Hurd — это набор серверов, запущенных поверх микроядра GNU Mach. Эта комбинация составляет основу операционной системы GNU.

Дополнительную информацию о GNU/Hurd вообще можно найти на <https://www.gnu.org/software/hurd>, а о Debian GNU/Hurd — на <https://www.debian.org/ports/hurd/>.

See <https://www.debian.org/ports/#portlist-other> for more information about hurd-amd64 and previous non-linux ports.

1.5 Чем Debian GNU/Linux отличается от других дистрибутивов Linux? Почему я должен отдать предпочтение Debian, а не какому-то другому дистрибутиву?

Вот ключевые возможности, отличающие Debian от других дистрибутивов Linux:

Свобода Как заявлено в **общественном договоре** (https://www.debian.org/social_contract), Debian всегда будет свободным на 100%. В Debian распространяется только действительно свободное программное обеспечение. Такое ПО должно отвечать **критериям Debian по определению свободного ПО** (https://www.debian.org/social_contract#guidelines).

Система управления пакетами Debian Система в целом или любая её часть может быть обновлена без переформатирования, без потери пользовательских настроек и (в большинстве случаев) без перезагрузки системы. В большинстве современных дистрибутивов Linux тоже есть свои системы управления пакетами; система управления пакетами Debian уникальна и хорошо продумана (см. Глава 7).

Открытая модель разработки Многие дистрибутивы Linux разрабатываются индивидами, небольшими закрытыми группами, либо коммерческими компаниями. Debian же представляет собой крупный дистрибутив Linux, который разрабатывается объединением индивидов, общей целью которых является создание свободной операционной системы в духе Linux и другого свободного ПО.

More than 1000 volunteer package maintainers are working on over 64961 packages and improving Debian GNU/Linux. The Debian developers contribute to the project not by writing new applications (in most cases), but by packaging existing software according to the standards of the project, by communicating bug reports to upstream developers, and by providing user support. Additionally, more than 400 volunteer in the project doing other tasks. Examples of these tasks include: creating documentation, contributing to the website and translating. See also the additional information on how to become a contributor in Глава 13.

Универсальная операционная система Debian поставляется с **более чем 64961 пакетами** (<https://packages.debian.org/stable/>) и работает на **7 архитектурах** (<https://www.debian.org/ports/>). Это намного больше, чем доступно в любом другом дистрибутиве GNU/Linux. Обзор поставляемого ПО см. в Раздел 5.1, а описание поддерживаемых платформ см. в Раздел 4.1.

Система отслеживания ошибок В связи с территориальной рассредоточенностью разработчиков Debian для ускорения разработки системы и быстрого исправления ошибок им требуется подходящий инструментарий. Пользователи присылают сообщения об обнаруженных ошибках в специальном формате, которые становятся доступны через WWW-архивы и электронную почту. Для дополнительной информации о ведении журнала ошибок см. Раздел 12.4.

Политика Debian В Debian есть всестороннее описание стандартов качества — политика Debian (Debian Policy). Этот документ определяет качества и стандарты, согласно которым мы разрабатываем пакеты Debian.

Дополнительную информацию об этом можно найти на странице **причин выбора Debian** (https://www.debian.org/intro/why_debian).

1.6 Как соотносятся проект Debian и проект GNU Free Software Foundation?

Система Debian строится на основе идеалов свободного ПО, которые были предложены **Фондом свободного ПО** (<https://www.gnu.org/>) и в частности **Ричардом Столлманом** (<https://www.stallman.org/>). Мощные инструменты системной разработки, утилиты и приложения Фонда свободного ПО являются ключевой частью системы Debian.

Проект Debian не является частью FSF, однако мы регулярно общаемся и совместно работаем над различными проектами. FSF попросила, чтобы мы называли нашу систему «Debian GNU/Linux», и мы счастливы были выполнить их просьбу.

1.7 Как произносится слово Debian и что оно означает?

Имя проекта произносится как Дебиан, с коротким е и ударением на первом слоге. Это слово образовано из имён Debra и Ian Murdock, которые основали проект. (Кажется, словари дают несколько двусмысленную транскрипцию слова Ian (!), но Иэн предпочитает транскрипцию ee'-en.)

Глава 2

Получение и установка Debian GNU/Linux

Официальным документом с инструкциями по установке является [руководство по установке Debian GNU/Linux](https://www.debian.org/releases/stable/installmanual) (<https://www.debian.org/releases/stable/installmanual>). Здесь мы приведём дополнительную информацию о получении и установке Debian GNU/Linux.

2.1 Какая версия Debian является последней?

В настоящее время существует три версии Debian GNU/Linux:

выпуск 13, также известен как «стабильный (stable)» дистрибутив или trixie Это стабильное и хорошо протестированное ПО, которое изменяется, если в него включаются существенные исправления, касающиеся безопасности или удобства использования.

«тестируемый» (testing) дистрибутив, в настоящий момент называется fork Он включает пакеты, которые войдут в следующую «стабильную» версию; эти пакеты прошли некоторое тестирование в нестабильной версии, но для выпуска пока не готовы. Данная версия обновляется намного чаще, чем «стабильная», но не так часто, как «нестабильная».

нестабильный дистрибутив This is the version currently under development; it is updated continuously. You can retrieve packages from the "unstable" archive on any Debian mirror site and use them to upgrade your system at any time, but you may not expect the system to be as usable or as stable as before - that's why it's called "unstable"!

Подробности см. в Раздел [6.1](#).

2.2 Обновляются ли пакеты в «stable»?

Generally speaking, no new functionality is added to the stable release. Once a Debian version is released and tagged "stable" most packages will only get security updates. That is, packages for which a security vulnerability has been found after the release will be upgraded. All the security updates are served through security.debian.org (<https://security.debian.org>).

However, there are some cases in which packages will be updated in stable. For example:

- When an urgent update is required to ensure the software continues working.
- The package is a data package and the data must be updated in a timely manner.
- The package needs to be current to useful to end user (e.g. some security software, such as anti-malware products).
- The software is a leaf package and is broken by changes external to the distribution.

Users that wish to run updated versions of the software in stable have the option to use "backports". Backports are recompiled packages from testing (mostly) and unstable (in a few cases only, e.g. security updates), so they will run without new libraries (wherever it is possible) on a stable Debian distribution. Users can configure their system to use the backports repository and download specific software. However, it is recommended to pick out single backports which fit the specific needs, and not to use all backports available. For more information read the [Wiki entry describing software available to Debian users](https://wiki.debian.org/DebianSoftware) (<https://wiki.debian.org/DebianSoftware>) and [Wiki entry on backports](https://wiki.debian.org/Backports) (<https://wiki.debian.org/Backports>).

Обновления безопасности служат одной цели: исправить уязвимость. Это не какой-то хитрый способ протащить изменения в стабильный выпуск в обход обычной процедуры выпуска. Поэтому исправление проблем безопасности в пакетах не приводит к обновлению ПО. Команда безопасности Debian переносит изменения для исправления уязвимости именно в ту версию, которая входит в «стабильный» выпуск.

Дополнительную информацию о поддержке безопасности см. в [ЧАВО по безопасности](https://www.debian.org/security/faq) (<https://www.debian.org/security/faq>) и в [Руководстве по безопасности Debian](https://www.debian.org/doc/manuals/securing-debian-howto/) (<https://www.debian.org/doc/manuals/securing-debian-howto/>).

2.3 Где и как получить установочные образы Debian?

Very likely, clicking the "Download" link on the [main page of the Debian website](https://www.debian.org/) (<https://www.debian.org/>) will get you what you need. However, for more options, get the various installation images from the [Download Debian](https://www.debian.org/distrib/) (<https://www.debian.org/distrib/>) page.

Более подробная информация об образах CD (и DVD) представлена на странице [Debian GNU/Linux на компакт-дисках](https://www.debian.org/CD/) (<https://www.debian.org/CD/>).

2.4 Как установить Debian с компакт-дисков?

Установка Debian с компакт-диска весьма проста: настройте свою систему на загрузку с компакт-диска, вставьте ваш установочный компакт-диск, перезагрузите компьютер. Ваша система запустит программу установки Debian. Дополнительную информацию см. в [Руководстве по установке Debian GNU/Linux](https://www.debian.org/releases/stable/installmanual) (<https://www.debian.org/releases/stable/installmanual>).

2.5 Зачем на CD официального стабильного дистрибутива нужны символьные ссылки «frozen» и «unstable»? Я думал, что на этих CD только стабильный дистрибутив!

На официальных образах Debian CD действительно есть символьные ссылки

```
/dists/frozen -> trixie/
/dists/stable -> trixie/
/dists/testing -> trixie/
/dists/unstable -> trixie/
```

Они нужны для того, чтобы эти диски работали, если в sources.list есть запись типа

```
deb cdrom:[<b''иб''b''mb''b''яb'' b''xb''b''ab''b''xb'' b''нb''b''ab'' b''mb''b' ←
'eb''b''tb''b''kb''b''eb'' cd>]/ unstable main [b''...b'']
```

Наличие этих символьных ссылок вовсе *не* означает, что это «нестабильный», «тестируемый» или какой-то другой дистрибутив. Чтобы узнать, какой дистрибутив Debian на нём записан, прочтите метку CD-диска в файле `/.disk/info`. Эта информация есть и на CD в файле `/README.txt`.

О том, какой дистрибутив сейчас считается стабильным, а какой тестируемым, можно узнать [здесь](https://www.debian.org/releases) (<https://www.debian.org/releases>).

2.6 Можно ли установить Debian прямо с Интернет-сайта?

Yes. You can boot the Debian installation system from a set of files you can download from our archive site and its mirrors.

Можно скачать маленький файл CD-образа, создать из него загрузочный CD, установить с него базовую систему, а оставшуюся часть установить из сети. Более подробно об этом написано [здесь](https://www.debian.org/CD/netinst/) (<https://www.debian.org/CD/netinst/>).

2.7 Есть ли другие способы запуска программы установки системы?

Yes. Apart from CD or DVD, you can install Debian GNU/Linux by booting from USB memory stick, directly from hard disk, or using TFTP net booting. For installing on multiple computers it's possible to do fully automatic installations. NB: not all methods are supported by all computer architectures. Once the installer has booted, the rest of the system can be downloaded over the network, or installed from local media. See the [Debian GNU/Linux Installation Guide](https://www.debian.org/releases/stable/installmanual) (<https://www.debian.org/releases/stable/installmanual>) for more information.

Глава 3

Выбор дистрибутива Debian

Существует несколько различных дистрибутивов Debian. И очень важно сделать правильный выбор. В этой главе приводится информация, полезная для тех пользователей, которые хотят выбрать дистрибутив, наиболее подходящий для своей системы, а также рассматриваются ответы на вопросы, которые могут возникнуть в этом случае. Здесь речь идёт не о том, «почему лучше выбрать Debian», а о том, «какой из дистрибутивов Debian больше вам подходит».

Подробности о доступных дистрибутивах смотрите Раздел [6.1](#).

3.1 Какой дистрибутив Debian (стабильный/тестируемый/нестабильный) лучше всего мне подойдёт?

Ответить на этот вопрос не так-то просто. На самом деле это зависит от того, для чего он вам нужен. Лучше всего было бы спросить друга, который уже знаком с Debian. Но это не значит, что решение нельзя принять самостоятельно. Фактически, вы сможете принять решение после прочтения этой главы.

- Если для вас очень важна безопасность или стабильность — устанавливайте стабильный. Точка. Это самый лучший вариант достичь желаемого.
- Если вы новичок, и вам нужна настольная рабочая система, то начните со стабильного выпуска. Некоторые программы в нём могут оказаться старыми, но зато в стабильном выпуске меньше ошибок. Вы легко можете перейти на более современный нестабильный (или тестируемый) выпуск как только будете более уверенно чувствовать себя при работе с системой.
- If you are a desktop user with a lot of experience in the operating system and do not mind facing the odd bug now and then, or even full system breakage, use unstable. It has all the latest and greatest software, and bugs are usually fixed swiftly.
- Если вы настраиваете сервер, особенно такой, требования к стабильности которого довольно серьёзны, или если он доступен из сети Интернет, то устанавливайте стабильный. Это, безусловно, самый правильный и безопасный выбор.

Надеемся, что ответы на дальнейшие вопросы больше прояснят ситуацию. Если после прочтения всех ЧаВо вам всё ещё трудно принять решение, остановитесь на стабильном дистрибутиве.

3.1.1 Вы предлагаете установить стабильный дистрибутив, но при его использовании не обнаруживается или не работает такое-то аппаратное обеспечение. Что делать?

Попытайтесь обратиться к системам поиска в веб, вероятно кто-то смог добиться работы такого оборудования в стабильном выпуске. Большая часть оборудования вполне нормально работает в стабильном выпуске. Если же у вас какое-то очень свежее оборудование, то оно может не работать в стабильном выпуске. Если это так, то вы можете установить тестируемый или нестабильный выпуск, либо выполнить обновление до одного из них.

Список работающих в Linux ноутбуков можно найти на **этом** (<https://www.linux-on-laptops.com/>) замечательном сайте. Там не описывается работоспособность именно в Debian, но это потрясающий ресурс. Для настольных компьютеров такого сайта не найти.

Another option would be to ask in the debian-user mailing list by sending an email to debian-user@lists.debian.org. Messages can be posted to the list even without subscribing. The archives can be read through <https://lists.debian.org/debian-user/>. Information regarding subscribing to the list can be found at the location of archives. You are strongly encouraged to post your questions on the mailing-list rather than on **irc** (<https://www.debian.org/support>). The mailing-list messages are archived, so the solution to your problem can help others with the same issue.

3.1.2 Есть ли разница между версиями пакетов в различных дистрибутивах?

Да. В нестабильном дистрибутиве находятся самые новые (последние) версии. Но пакеты в нём недостаточно хорошо протестированы и могут содержать ошибки.

С другой стороны, стабильный дистрибутив содержит старые версии пакетов. Но пакеты в нём были хорошо протестированы и, по всей вероятности, не содержат неизвестных ошибок.

Пакеты в тестируемом дистрибутиве — что-то среднее между двумя этими крайностями.

3.1.3 В стабильных дистрибутивах содержатся устаревшие версии программ. Только взгляните на Kde, Gnome, Xorg или даже ядро. Они очень старые. Почему?

Да, в общем вы правы. Возраст пакетов в стабильном дистрибутиве зависит от времени выпуска. Так как обычно между выпусками проходит больше года, отсюда и получаются старые версии пакетов. Однако, они были хорошо протестированы на момент выпуска и работают даже сейчас. Можно уверенно сказать, что в пакетах нет неизвестных серьёзных ошибок, проблем с безопасностью и т. д. Пакеты в стабильном дистрибутиве очень тесно подогнаны друг к другу. Все перечисленные плюсы очень важны для рабочих серверов, которые функционируют 24 часа в день, 7 дней в неделю.

On the other hand, packages in testing or unstable can have hidden bugs, security holes etc. Moreover, some packages in testing and unstable might not be working as intended. Usually people working on a single desktop prefer having the latest and most modern set of packages. Unstable is the solution for this group of people.

Как видите, стабильность и новизна находятся на разных концах спектра. Если нужна стабильность, устанавливайте стабильный дистрибутив. Если хотите работать с самыми новыми версиями пакетов, ставьте нестабильный.

3.1.4 Возможно ли позже перейти на другой дистрибутив и как это сделать?

Да, но это односторонний процесс. Вы можете перейти со стабильного на тестируемый, а затем на нестабильный. Но обратно вернуться невозможно. Лучше дважды подумать, прежде чем устанавливать/переходить на нестабильный дистрибутив.

Actually, if you are an expert and if you are willing to spend some time and if you are real careful and if you know what you are doing, then it might be possible to go from unstable to testing and then to stable. The installer scripts are not designed to do that. So in the process, your configuration files might be lost and...

3.1.5 Не могли бы вы подсказать мне какой выпуск следует устанавливать, стабильный, тестируемый или нестабильный?

No. This is a rather subjective issue. There is no perfect answer as it depends on your software needs, your willingness to deal with possible breakage, and your experience in system administration. Here are some tips:

- Стабильный выпуск стабилен как скала. Он не ломается и имеет полную поддержку безопасности. Но он может не работать на очень новом оборудовании.

- Тестируемый выпуск содержит более свежее ПО, чем стабильный, а ломается значительно реже, чем нестабильный выпуск. Но он всё равно может ломаться, иногда требуется длительное время для того, чтобы всё снова заработало. Иногда для этого требуются дни, а иногда даже месяцы. Кроме того, для него не обеспечивается поддержка безопасности.
- Нестабильный выпуск поддерживает самое свежее ПО и сильно меняется. Следовательно, он может сломаться в любой момент. Тем не менее, исправления выпускаются зачастую в течение пары дней, а ПО в нём всегда самое свежее из того, что имеется в Debian.

Если вы выбираете между тестируемым и нестабильным выпусками, имейте в виду, что иногда полезнее использовать тестируемый выпуск. Один из авторов этой документации испытал подобную ситуацию, которая возникла из-за смены версии gcc с gcc3 на gcc4. Он попытался установить пакет `labplot` на машину с нестабильным выпуском, но этот пакет нельзя было установить в нестабильном выпуске, так как для некоторых зависимостей этого пакета переход на gcc4 уже был выполнен, а для других — ещё нет. Но в тестируемом выпуске этот пакет можно было установить, поскольку пакеты, перешедшие на gcc4, ещё не "просочились" в тестируемый выпуск.

3.1.6 Вы упомянули, что тестируемый дистрибутив иногда ломается. Что имеется в виду?

Бывает, что пакет невозможно установить с помощью системы управления пакетами. Иногда пакет может быть недоступен совсем, может быть (временно) удалён из-за ошибок или неудовлетворённых зависимостей. Иногда пакет устанавливается, но работает неправильно.

Когда такое случается, говорят, что дистрибутив сломан (по крайней мере, применительно к этому пакету).

3.1.7 Почему тестируемый выпуск может быть сломан в течение нескольких месяцев? Разве исправления, добавляемые в нестабильный выпуск, не переходят в тестируемый?

The bug fixes and improvements introduced in the unstable distribution trickle down to testing after a certain number of days. Let's say this threshold is 5 days. The packages in unstable go into testing only when there are no RC-bugs reported against them. If there is a RC-bug filed against a package in unstable, it will not go into testing after the 5 days.

The idea is that, if the package has any problems, it would be discovered by people using unstable and will be fixed before it enters testing. This keeps testing in a usable state for most of the time. Overall a brilliant concept, if you ask me. But things aren't always that simple. Consider the following situation:

- Предположим, что вам нужен пакет XYZ.
- Также представим, что на 10 июня его версия в тестируемом дистрибутиве XYZ-3.6, а в нестабильном XYZ-3.7.
- After 5 days, XYZ-3.7 from unstable migrates into testing.
- So on June 15, both testing and unstable have XYZ-3.7 in their repositories.
- Let's say, the user of testing distribution sees that a new XYZ package is available and updates the XYZ-3.6 to XYZ-3.7.
- Теперь, 25 июня кто-то использующий тестируемый или нестабильный дистрибутив обнаруживает RC-ошибку в XYZ-3.7 и пишет письмо об этом в BTS.
- Сопровождающий XYZ исправляет эту ошибку и загружает исправленную версию в нестабильный дистрибутив, скажем, 30 июня. Здесь предполагается, что потребовалось 5 дней, чтобы сопровождающий исправил ошибку и закачал новую версию. Число 5 не следует воспринимать как постоянную величину. Оно может быть меньше или больше, в зависимости от сложности имеющейся RC-ошибки.
- This new version in unstable, XYZ-3.8 is scheduled to enter testing on July 5th.
- But on July 3rd some other person discovers another RC-bug in XYZ-3.8.

- Предположим, что сопровождающий XYZ исправил эту новую RC-ошибку и закачал новую версию XYZ через 5 дней.
- So on July 8th, testing has XYZ-3.7 while unstable has XYZ-3.9.
- This new version XYZ-3.9 is now rescheduled to enter testing on July 13th.
- Now since you are running testing, and since XYZ-3.7 is buggy, you could probably use XYZ only after July 13th. That is you essentially ended up with a broken XYZ for about one month.

The situation can get much more complicated, if say, XYZ depends on 4 other packages. This could in turn lead to an unusable testing distribution for months. While the scenario above is imaginary, similar things can occur in real life, though they are rare.

3.1.8 С точки зрения администратора, какой дистрибутив требует большего внимания?

One of the main reasons why many people choose Debian over other Linux distributions is that it requires very little administration. People want a system that just works. In general one can say that stable requires very little maintenance, while testing and unstable require constant maintenance from the administrator. If you are running stable, all you need to worry about is keeping track of security updates. If you are running either testing or unstable it is a good idea to be aware of the new bugs discovered in the installed packages, new bugfixes/features introduced etc.

3.1.9 Что происходит при выходе новой версии дистрибутива?

Этот вопрос не поможет вам в выборе дистрибутива Debian. Но рано или поздно он встанет перед вами.

The stable distribution is currently trixie; The next stable distribution will be called forky. Let's consider the particular case of what happens when forky is released as the new stable version.

- Старый стабильный (oldstable) = bookworm; стабильный (stable) = trixie; тестируемый (testing) = forky; нестабильный (unstable) = sid
- Нестабильный всегда указывает на sid, независимо от того, вышла ли новая версия или нет.
- Пакеты постоянно переносятся из sid в тестируемый (то есть в forky). А пакеты в стабильном (то есть в trixie) не меняются (за исключением обновлений безопасности).
- По прошествии какого-то времени тестируемый замораживают. Но он всё равно пока будет называться тестируемым. В этот период никакие новые пакеты из нестабильного дистрибутива в тестируемый перемещаться не могут, за исключением лишь тех, что содержат исправления ошибок, критических для выпуска (release-critical — RC).
- When testing is frozen, all the new bugfixes introduced have to be manually checked by the members of the release team. This is done to ensure that there won't be any unknown severe problems in the frozen testing.
- Количество критических ошибок в 'замороженном тестируемом выпуске' необходимо снизить до нуля, либо если их число больше нуля, то эти ошибки отмечаются как игнорируемые для этого выпуска или как отложенные до следующей редакции этого выпуска
- The 'frozen testing' with no rc-bugs will be released as the new stable version. In our example, this new stable release will be called forky.
- На этой стадии старый стабильный = trixie, стабильный = forky. Содержимое стабильного и «замороженного тестируемого» в этот момент одинаково.
- Новый тестируемый выпуск основывается на старом тестируемом выпуске.
- Пакеты начинают поступать из sid в тестируемый, и сообщество Debian начинает работать над следующим стабильным выпуском.

3.1.10 У меня на настольном компьютере/кластере установлен Debian. Как узнать, какой дистрибутив используется?

В большинстве случаев это очень легко сделать. Посмотрите файл `/etc/apt/sources.list`. Там будет строка, подобная этой:

```
deb http://deb.debian.org/debian/ unstable main contrib
```

Третье поле («unstable» в вышеприведённом примере) указывает на отслеживаемый дистрибутив Debian, установленный в системе.

Также вы можете использовать команду `lsb_release` (из пакета `lsb-release`). Если вы запустите эту программу на компьютере с нестабильной системой, то получите:

```
$ lsb_release -a
LSB Version:    core-2.0-noarch:core-3.0-noarch:core-3.1-noarch:core-2.0-ia32: ←
                core-3.0-ia32:core-3.1-ia32
Distributor ID: Debian
Description:    Debian GNU/Linux unstable (sid)
Release:        unstable
Codename:       sid
```

Однако, это не всегда так легко. В некоторых системах могут быть файлы `sources.list` с несколькими строками, указывающими на различные дистрибутивы. Так бывает, когда администратор следит за различными пакетами из различных дистрибутивов Debian. Это часто называется `apt-pinning`. На таких компьютерах может использоваться смесь дистрибутивов.

3.1.11 I am currently using the stable version. Can I change to testing or unstable? If so, how?

First of all, please bear in mind that the stable version is the one recommended for servers as well as for desktop computers, not only you will get security updates if you are running stable but also there are less changes which could potentially break the system or your setup.

If you are currently running stable, then in the `/etc/apt/sources.list` file the third field will be either `'trixie'` or `'stable'`. If you want to change to a different version, you need to change this to the distribution you want to run. If you want to run testing, then change the third field of `/etc/apt/sources.list` to `'testing'`. If you want to run unstable, then change the third field to `'unstable'`.

Currently testing is called `forky`. So, if you change the third field of `/etc/apt/sources.list` to `'forky'`, then also you will be running testing. But even when `forky` becomes stable, you will still be tracking `forky`.

Нестабильный всегда называется Sid. Поэтому, если вы измените значение третьего поля в `/etc/apt/sources.list` на `'sid'`, то у вас будет отслеживаться нестабильный выпуск.

Currently Debian offers does not offer security updates for testing nor for unstable. The **Debian Security Team** (<https://security-team.debian.org/>) focus their work on stable and old-stable. Nevertheless, just like any other type of fixes, security fixes in unstable are directly made to the main archive and trickle down to testing in due course. So if you are running unstable make sure that you remove the lines relating to security updates in `/etc/apt/sources.list`. If you are interested in knowing what known security bugs exist in these versions of the distributions, you will find this information in the **list of vulnerable source packages in testing** (<https://security-tracker.debian.org/tracker/status/release/testing>), and **unstable** (<https://security-tracker.debian.org/tracker/status/release/unstable>).

If there is a release notes document available for the distribution you are upgrading to (even though it has not yet been released) it would be wise to review it, as it might provide information on how you should upgrade to it. You will always find the **Release Notes for testing** (<https://www.debian.org/releases/testing/releasenotes>) available at the Debian website but depending on how close the testing version is to release the document might not cover all the potential changes or pitfalls.

Тем не менее, после того как были произведены вышеуказанные изменения, вы можете запустить `aptitude update` и затем устанавливать нужные вам пакеты. Заметим, что установка пакетов от другого дистрибутива может привести к обновлению половины системы. Если вы устанавливаете отдельные пакеты, то получите систему, работающую на смеси дистрибутивов.

В некоторых ситуациях лучше выполнить полное обновление до нового дистрибутива, запустив `apt full-upgrade`, `aptitude safe-upgrade` или `aptitude full-upgrade`. Подробнее об этом можно узнать из справочных страниц по `apt` и `aptitude`.

The Debian reference manual provides more insight on running testing and unstable in its section [Life with eternal upgrades](https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_life_with_eternal_upgrades) (https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#_life_with_eternal_upgrades).

3.1.12 Сейчас я использую тестируемый дистрибутив (forky). Что произойдёт после выпуска следующей версии? У меня по-прежнему будет отслеживаться тестируемый дистрибутив, или на моей машине будет новый стабильный дистрибутив?

Это зависит от записей в файле `/etc/apt/sources.list`. Если сейчас у вас отслеживается тестируемый дистрибутив, то там будут строки вида:

```
deb http://deb.debian.org/debian/ testing main
```

или

```
deb http://deb.debian.org/debian/ forky main
```

Если в третьем поле файла `/etc/apt/sources.list` стоит «testing», то даже после выхода нового выпуска у вас будет отслеживаться тестируемый дистрибутив. Поэтому после выхода `forky` вы будете работать на новом дистрибутиве Debian с другим кодовым именем. Сначала изменения будут незаметны, но они проявятся, как только новые пакеты начнут переходить из нестабильного дистрибутива в тестируемый.

Но если третье поле содержит «forky», то вы перейдёте на стабильный дистрибутив (так как `forky` станет новым стабильным дистрибутивом).

3.1.13 Всё равно непонятно. Так что же нужно устанавливать?

If unsure, the best bet would be the stable distribution.

3.2 But what about Kali, Knoppix, Linux Mint, Ubuntu, and others?

These distributions are not Debian; they are *Debian-based*. Though there are many similarities and commonalities between them, there are also crucial differences.

Over the years, many distributions have been developed over time reusing and/or rebuilding Debian packages and also adding custom packages on their own. Most of the distributions have been created for specific audiences or purposes. According to Distrowatch, Debian has spawned more than **420 derivatives** (<https://distrowatch.com/search.php?basedon=Debian&status=All#distrosearch>) and more than 120 of them are active at the time of writing.

All these distributions have their own merits and are suited to some specific set of users. For more information, read [Debian derivatives](https://www.debian.org/misc/children-distros) (<https://www.debian.org/misc/children-distros>) available at the Debian website. You can find a complete list of Debian derivatives, including those that are no longer under active development at [the Debian derivate Census](https://wiki.debian.org/Derivatives/Census) (<https://wiki.debian.org/Derivatives/Census>) in the Wiki.

3.2.1 I know that Kali/Knoppix/Linux Mint/Ubuntu/... is Debian-based. So after installing it on the hard disk, can I use 'apt' package tools on it?

These distributions are Debian-based, but they are not Debian. You will be still able to use `apt` package tools by pointing the `/etc/apt/sources.list` file to these distributions' repositories. In some cases some distributions might even have additional package managers that are used instead of `apt`.

В большинстве случаев, если вы начали использовать определённый дистрибутив, то должны использовать только его пакеты и не устанавливать пакеты из других дистрибутивов. Очень часто происходят поломки в работе из-за того, что люди пытаются установить в Debian пакеты из других дистрибутивов. Тот факт, что они используют одинаковый формат и расширение (`.deb`), не делает их совместимыми между собой.

For example, Knoppix is a Linux distribution designed to be booted as a live CD whereas Debian is designed to be installed on the hard-disk. Knoppix is great if you want to know whether a particular piece of hardware works, or if you want to experience how a GNU/Linux system 'feels' etc., Knoppix is good for demonstration purposes while Debian is designed to run 24/7. Moreover the number of packages available and the number of architectures supported by Debian are far more than that of Knoppix.

If you want Debian, it is best to install Debian from the get-go. Although it is possible to install Debian through other distributions, such as Knoppix, the procedure calls for expertise. If you are reading this FAQ, I would assume that you are new to both Debian and Knoppix. In that case, save yourself a lot of trouble later and install Debian right from the beginning.

3.2.2 I installed Kali/Knoppix/Linux Mint/Ubuntu/... on my hard disk. Now I have a problem. What should I do?

You are advised not to use the Debian forums (either mailing lists or IRC) for help on Debian derivatives as people there may base their suggestions on the assumption that you are running a Debian system. These "fixes" might not be suited to what you are running, and might even make your problem worse.

Используйте в первую очередь форумы того дистрибутива, который вы используете. Если вы не получаете помощи, или та помощь, которую вы получаете, не решает вашу проблему, вы можете попробовать спросить на форуме Debian, но помните о рекомендации из предыдущего параграфа.

3.2.3 I'm using Kali/Knoppix/Linux Mint/Ubuntu/... and now I want to use Debian. How do I migrate?

Считайте смену дистрибутива на основе Debian сменой одной операционной системы на другую. Вам нужно сделать резервную копию всех данных и переустановить операционную систему с нуля. Не пытайтесь выполнить «обновление» до Debian с помощью программ управления пакетами, так как это может привести к неработоспособности системы.

Если пользовательские данные (то есть /home) расположены на отдельном разделе, то перейти на Debian очень просто, вам нужно просто указать системе установки смонтировать (но без форматирования) этот раздел при переустановке. Не забудьте сделать резервную копию данных, а также файлов настроек предыдущей системы (то есть /etc/ и, может быть, /var/).

Глава 4

Проблемы совместимости

4.1 На каких архитектурах/системах работает Debian GNU/Linux?

Debian GNU/Linux содержит исходный код всех входящих в него программ, поэтому он должен работать на всех системах, которые поддерживаются ядром Linux. Подробности см. в [Linux FAQ](http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER) (<http://en.tldp.org/FAQ/Linux-FAQ/intro.html#DOES-LINUX-RUN-ON-MY-COMPUTER>).

Текущий выпуск Debian GNU/Linux 13 содержит двоичный код для следующих архитектур:

- *amd64*: системы с 64-битными процессорами AMD с расширением AMD64 и любыми процессорами Intel с расширением EM64T, и 64-битным пользовательским окружением;
- *arm64*: поддерживает современные устройства на 64-битных процессорах ARM.
- *armel*: машины ARM с процессором с обратным порядком байт;
- *armhf*: альтернатива *armel* для машин ARMv7 с аппаратной плавающей запятой.
- *i386*: ПК с процессорами Intel и совместимыми с ними, включая Intel 386, 486, Pentium, Pentium Pro, Pentium II (Klamath и Celeron), Pentium III и почти все совместимые процессоры от AMD, Cyrix и другие;
- *ia64*: компьютеры Intel IA-64 («Itanium»);
- *mips*: MIPS системы от SGI с прямым порядком байт — Indy и Indigo2; *mipsel*: MIPS системы с обратным порядком байт — Digital DECstation;
- *powerpc*: некоторые машины IBM/Motorola PowerPC, включая модели PowerMac Apple Macintosh и машины с открытой архитектурой CHRP и PReP;
- *ppc64el*: 64-битный перенос на PowerPC с порядком байтов от младшего к старшему, поддерживает некоторые новые процессоры PowerPC/POWER.
- *s390x*: 64-битный перенос для машин IBM System z, заменил s390.

The development of binary distributions of Debian for *hurd-i386* (for GNU Hurd kernel on i386 32-bit PCs), *mipsel64* (for 64 bit MIPS in little-endian mode), *powerpcspe* (port for the "Signal Processing Engine" hardware), *sparc64* (for 64 bit SPARC processors), *sh* (for Hitachi SuperH processors), and *x32* (for amd64/x86_64 CPUs using 32-bit pointers) is currently underway.

Поддержка архитектуры *m68k* была прекращена в выпуске Etch (Debian 4.0), так как она не удовлетворяла требованиям ответственных за выпуск Debian. К данной архитектуре относятся машины Amiga и ATARI с процессорами Motorola 680x0, где $x \geq 2$ означает встроенный модуль MMU. Однако, даже не будучи частью официального стабильного выпуска, данный перенос всё ещё остаётся активным и доступен для установки, и может быть возвращён в число действующих в будущих выпусках.

Поддержка *hppa* (машины Hewlett-Packard с архитектурой PA-RISC) и *alpha* (системы Compaq/Digital с архитектурой Alpha) была прекращена в выпуске Squeeze (Debian 6.0) по схожим причинам. Также в этом выпуске прекращена поддержка архитектуры *arm*, вследствие перехода на поддержку *armel*.

Support for the 32-bit s390 port (s390) was discontinued and replaced with s390x in Jessie (Debian 8). In addition, the ports to IA-64 and Sparc had to be removed from this release due to insufficient developer support.

Подробности см. на [странице переносов](https://www.debian.org/ports/) (<https://www.debian.org/ports/>).

Информация о загрузке, разметке диска, включении устройств PCMCIA (PC Card) и о прочих вопросах такого рода приведена в [руководстве по установке](https://www.debian.org/releases/stable/installmanual) (<https://www.debian.org/releases/stable/installmanual>).

4.2 На каких ядрах работает Debian GNU/Linux?

Beside Linux, Debian provides a complete, binary distribution for the following operating system kernels:

- FreeBSD: provided through the *kfreebsd-amd64* and *kfreebsd-i386* ports, for 64-bit PCs and 32-bit PCs respectively. These ports were first released in Debian 6.0 Squeeze as a *technology preview*. However they were not part of the Debian 8 Jessie release.

Кроме этого, ведётся работа над следующими переносами:

- *avr32* — перенос для 32-битной архитектуры Atmel,
- *hurd-i386* — перенос для 32-битных ПК. В данном переносе будет использована GNU Hurd — новая операционная система, собираемая группой GNU,
- *sh* — перенос для процессоров Hitachi SuperH.

Были попытки переноса дистрибутива на ядро NetBSD, где предоставлялись ядра *netbsd-i386* (для 32-битных ПК) и *netbsd-alpha* (для машин Alpha), но эти переносы никогда не выпускались, и в данный момент работа над ними свёрнута.

Подробности см. на [странице переносов](https://www.debian.org/ports/) (<https://www.debian.org/ports/>).

4.3 Насколько Debian совместим с другими дистрибутивами Linux?

Debian developers communicate with other Linux distribution creators in an effort to maintain binary compatibility across Linux distributions.¹ Most commercial Linux products run as well under Debian as they do on the system upon which they were built.

Debian GNU/Linux придерживается [Стандарта иерархии файловой системы Linux](https://www.pathname.com/fhs) (<https://www.pathname.com/fhs>). Тем не менее, некоторые правила этого стандарта можно интерпретировать по-разному, поэтому между Debian и другими системами Linux возможны различия.

4.4 Насколько исходный код Debian совместим с другими системами Unix?

Исходный код большинства приложений Linux совместим с другими системами Unix. Поддерживается почти всё, что есть в системах Unix System V, а также в свободных и коммерческих системах, происходящих от BSD. Однако, для бизнеса такое утверждение о Unix почти не имеет значения, потому что доказать это никак нельзя. В области разработки программного обеспечения необходима полная совместимость, а не совместимость «в большинстве случаев». Поэтому несколько лет тому назад назрела необходимость в стандартах, и теперь POSIX.1 (IEEE Стандарт 1003.1-1990) является одним из основных стандартов по совместимости исходного кода для Unix-подобных операционных систем.

¹The [Linux Standard Base](https://wiki.linuxfoundation.org/lsb/start/) (<https://wiki.linuxfoundation.org/lsb/start/>) is a specification for allowing the same binary package to be used on multiple distributions. After Jessie (Debian 8) was released, Debian [abandoned](https://sources.debian.org/src/lsb/9.20170808/debian/README.Debian/) (<https://sources.debian.org/src/lsb/9.20170808/debian/README.Debian/>) the pursuit of LSB compatibility. See this [July 3, 2015 message from Didier Raboud](https://lists.debian.org/4526217.myWFlvm1rM@gy1lingar) (<https://lists.debian.org/4526217.myWFlvm1rM@gy1lingar>) and the following discussion for background information.

ОС Linux ориентирована на соответствие POSIX.1, но стандарты POSIX стоят настоящих денег, а сертификация POSIX.1 (и FIPS 151-2) довольно дорога; это ещё более усложняет разработчикам Linux работу по достижению полного соответствия POSIX. Ввиду высокой стоимости сертификации маловероятно, что Debian получит официальное подтверждение о соответствии, даже если он полностью пройдёт аттестационный пакет. (Аттестационный пакет теперь свободно доступен, так что ожидается, что ещё больше людей будет работать над вопросами соответствия POSIX.1).

В Unifix GmbH (Брауншвейг, Германия) была разработана Linux-система, которая была сертифицирована как удовлетворяющая FIPS 151-2 (расширенный набор POSIX.1). Эта технология была доступна в собственном дистрибутиве Unifix, названном Unifix Linux 2.0, и в Lasermoon Linux-FT.

4.5 Можно ли использовать пакеты Debian (файлы «.deb») в системе RedHat/Slackware/... Linux? Можно ли использовать пакеты RedHat (файлы «.rpm») в системе Debian GNU/Linux?

В различных дистрибутивах Linux используются различные форматы пакетов и различные программы управления пакетами.

То, что вы, вероятно, сможете сделать Существует программа для распаковки пакета Debian на Linux-машине, работающей под «чужим» дистрибутивом, и она, в общем, работает, в том смысле, что файлы будут распакованы. Обратный вариант скорее всего тоже будет работать, т. е. программа сможет распаковать пакет для RedHat или Slackware на машине, работающей под Debian GNU/Linux, и разместить большинство файлов по нужным каталогам. Это в значительной степени стало возможным благодаря существованию (и строгому соблюдению) стандарта иерархии файловой системы Linux. Для преобразования различных форматов пакетов друг в друга используется пакет **Alien** (<https://packages.debian.org/alien>).

То, что вы, скорее всего, делать не захотите Большинство программ управления пакетами при распаковке архива сохраняют какие-то управляющие файлы. Эти файлы обычно не стандартизированы. Поэтому распаковка пакета Debian на «чужой» машине может непредсказуемым (и навряд ли положительным) образом повлиять на программу управления пакетами этой системы. Таким же образом и архивы других дистрибутивов могут быть успешно распакованы их утилитами в системе Debian, но это может привести к сбоям системы управления пакетами Debian, когда придёт время обновления или удаления каких-нибудь пакетов, или даже просто при выводе списка пакетов, установленных в системе.

Лучший способ: Стандарт Файловой Системы Linux (а, следовательно, и Debian GNU/Linux) требует, чтобы подкаталоги в `/usr/local/` полностью находились в распоряжении пользователей. Поэтому пользователи могут распаковывать «чужие» пакеты в этот каталог, а затем лично управлять их настройкой, обновлением и удалением.

4.6 Как установить не-Debian программу?

Файлы в каталоге `/usr/local/` не контролируются системой управления пакетами Debian. Поэтому хорошим вариантом будет размещение исходного кода нужных вам программ в каталоге `/usr/local/src/`. Например, файлы из пакета «foo.tar» можно распаковать в каталог `/usr/local/src/foo`. После сборки поместите двоичные файлы в `/usr/local/bin/`, библиотеки в `/usr/local/lib/`, а файлы настроек в `/usr/local/etc/`.

Если ваша программа и/или файлы на самом деле должны находиться в каком-то другом каталоге, их всё равно можно сохранить в `/usr/local/`, а в нужных местах создать соответствующие символичные ссылки на файлы из `/usr/local/`. Можно, например, сделать так:

```
ln -s /usr/local/bin/foo /usr/bin/foo
```

В любом случае, если вы получили пакет, авторские права на который позволяют его дальнейшее распространение, вам стоило бы подумать о создании из него пакета для Debian и отправке его в систему Debian. О том, как стать разработчиком пакета, читайте в руководстве по политике Debian (см. Раздел 12.1).

Глава 5

Программное обеспечение, доступное в системе Debian

5.1 Какие виды приложений и средств разработки программ доступны в Debian GNU/Linux?

Как и многие другие дистрибутивы Linux, Debian GNU/Linux предоставляет:

- основные приложения GNU для разработки программного обеспечения, работы с файлами и обработки текста, включая gcc, g++, make, texinfo, Emacs, оболочку Bash и множество усовершенствованных утилит Unix;
- Perl, Python, Tcl/Tk и множество программ, модулей и библиотек для этих языков;
- TeX (LaTeX) и Lyx, dvips, Ghostscript;
- оконную систему Xorg, предоставляющую сетевой графический пользовательский интерфейс в Linux, и несметное количество X-приложений, в том числе окружения рабочих столов GNOME, KDE и Xfce;
- a full suite of networking applications, including servers for Internet protocols such as HTTP (WWW), FTP, NNTP (news), SMTP and POP (mail) and DNS (name servers); relational databases like PostgreSQL, MySQL; also provided are web browsers including the various Mozilla products,
- a complete set of office applications, including the LibreOffice productivity suite, Gnumeric and other spreadsheets, WYSIWYG editors, calendars.

More than 63879 packages, ranging from news servers and readers to sound support, FAX programs, database and spreadsheet programs, image processing programs, communications, net, and mail utilities, Web servers, and even ham-radio programs are included in the distribution. Other 1082 software suites are available as Debian packages, but are not formally part of Debian due to license restrictions.

5.2 Кто написал все эти программы?

Авторы программ, входящих в каждый из пакетов, указаны в соответствующих файлах `/usr/share/doc/ПАКЕТ` где вместо ПАКЕТ нужно подставить имя пакета.

Сопровождающие — это те, кто создаёт пакеты для системы Debian GNU/Linux; они перечислены в управляющем файле Debian (смотрите Раздел 7.4), который имеется в каждом пакете. В файле изменений Debian, `/usr/share/doc/ПАКЕТ/changelog.Debian.gz`, перечислены люди, которые также принимали участие в создании пакета Debian.

5.3 Как получить текущий список программ, для которых в Debian собраны пакеты?

Полный список доступен на любом из [зеркал Debian](https://www.debian.org/mirror/list) (<https://www.debian.org/mirror/list>), в файле `indices/Maintainers`. Этот файл содержит имена пакетов и имена, а также адреса электронной почты соответствующих сопровождающих.

[Веб-интерфейс для пакетов Debian](https://packages.debian.org/) (<https://packages.debian.org/>) удобно распределяет пакеты по приблизительно тридцати "разделам" архива Debian.

5.4 Как установить окружение разработчика для сборки пакетов?

Если вы хотите собирать пакеты в системе Debian, то вам нужно иметь минимальное окружение для разработки, включающее компилятор C/C++ и несколько других важных пакетов. Чтобы получить такое окружение, вам нужно просто установить `build-essential`. Это метапакет, в зависимости которого включены стандартные инструменты разработки, необходимые для сборки пакетов Debian.

Однако, для пересборки некоторых программ может потребоваться дополнительное программное обеспечение, в том числе заголовочные файлы библиотек или дополнительные инструменты, такие как, например, `autoconf` или `gettext`. Debian предоставляет множество инструментов, необходимых для упаковки другого ПО в пакеты Debian.

Иногда бывает довольно сложно выяснить, какие именно программы требуются для сборки, если вы, конечно, не планируете просто пересобирать пакеты Debian. Последнюю задачу решить довольно легко, так как в каждый официальный пакет должен включаться список дополнительного программного обеспечения (кроме пакетов из `build-essential`), необходимого для его сборки, то есть так называемые сборочные зависимости (`Build-Dependencies`). Для установки всех пакетов, необходимых для сборки определённого пакета исходного кода, и затем собственно для сборки этого пакета достаточно запустить:

```
# apt-get build-dep foo
# apt-get source --build foo
```

Учтите, что если вы хотите собирать ядра Linux, распространяемые в Debian, то вам также нужно установить пакет `kernel-package`. Подробности см. в Раздел [10.2](#).

5.5 Чего не хватает в Debian GNU/Linux?

There is a list of packages which still need to be packaged for Debian, the [Work-Needing and Prospective Packages list](https://www.debian.org/devel/wnpp/) (<https://www.debian.org/devel/wnpp/>).

For more details about adding missing things, see Глава [13](#).

5.6 Почему иногда выдаются сообщения «ld: cannot find -lfoo» при компиляции программ? Почему не хватает каких-то файлов `libfoo.so` в пакетах Debian с библиотеками?

Политика Debian требует, чтобы определённые символьные ссылки (на `libfoo.so.x.y.z` или подобные) размещались в отдельных пакетах — пакетах разработки. Эти пакеты обычно называются `libfoo-dev` или `libfooX-dev` (если библиотечный пакет называется `libfooX`, где X — целое число).

5.7 Поддерживает ли Debian Java (и если да, то как)?

В Debian доступно несколько *свободных* реализаций технологии Java в виде пакетов, предоставляющих Java Development Kits и Runtime Environments. В Debian можно и писать, и отлаживать, и запускать программы Java.

Running a Java applet requires a web browser with the capability to recognize and execute it. Several web browsers available in Debian, such as Mozilla or Konqueror, support Java plug-ins that enable running Java applets within them.

За дополнительной информацией обратитесь к [ЧАВО по Java в Debian](https://www.debian.org/doc/manuals/debian-java-faq/) (<https://www.debian.org/doc/manuals/debian-java-faq/>).

5.8 How can I check that I am using a Debian system, and what version it is?

Для того, чтобы убедиться, что ваша система была установлена с настоящих базовых дисков Debian, используйте команду

```
lsb_release -a
```

Она выведет название дистрибутива (в поле Distributor ID) и версию системы (в полях Release и Codename). Ниже приведён пример запуска этой команды в системе Debian:

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 7.4 (wheezy)
Release:        7.4
Codename:       wheezy
```

Также можно проверить наличие файла `/etc/debian_version`, который содержит однострочную запись с указанием номера версии выпуска, файл поставляется в пакете `base-files`.

Однако пользователи должны понимать, что система Debian состоит из многих частей, каждая из которых может обновляться (почти) независимо от других. Каждый «выпуск» Debian содержит строго определённое и неизменяемое содержимое. Обновления доступны отдельно. Чтобы получить состояние установки пакета `foo`, используйте команду `dpkg --get-selections foo`. Чтобы посмотреть версии установленных пакетов, запустите команду:

```
dpkg --get-selections foo
```

Для просмотра версий всех установленных пакетов, выполните команду:

```
dpkg -l
```

Note that the existence of the program `dpkg` shows that you should be able to install Debian packages on your system. However, since the program has been ported to many other operating systems and architectures, this is no longer a reliable method of determining if a system is Debian GNU/Linux.

5.9 Как в Debian поддерживаются языки, отличные от английского?

- Debian GNU/Linux распространяется с большим количеством клавиатурных раскладок для множества типов клавиатур и с утилитами (в пакете `kbd`), которые позволяют устанавливать, просматривать и редактировать таблицы клавиатурных раскладок.

The installation prompts the user to specify the keyboard to use.

- Nearly all of the software in Debian supports UTF-8 as character set. Legacy character sets, such as ISO-8859-1 or ISO-8859-2, should be considered obsolete.
- Currently, support for German-, Spanish-, French-, Hungarian-, Italian-, Japanese-, Korean-, Dutch-, Polish-, Portuguese-, Russian-, Turkish-, and Chinese-language manual pages is provided through the `manpages-LANG` packages (where *LANG* is the two-letter ISO country code). To access an NLS manual page, the user must set the shell `LC_MESSAGES` variable to the appropriate string.

Например, для чтения на итальянском языке `LC_MESSAGES` нужно установить в `'italian'`. Программа `man` будет искать справочные страницы на итальянском языке в каталоге `/usr/share/man/it/`.

5.10 I have a device that requires non-free firmware. What should I do?

Firmware refers to embedded software which controls electronic devices. Many devices require firmware to operate. Historically, firmware would be built into the device's ROM or Flash memory, but more and more often, a firmware image has to be loaded into the device RAM by a device driver during device initialisation.

Some devices require non-free firmware to work properly. And there could be firmware updates in the future.

In Debian release 11 (bullseye) and older, Debian did not include non-free firmware on official images and live installations. For Debian 12 onwards, all the packaged non-free firmware binaries that Debian can distribute have been moved to a new component in the Debian archive, called `non-free-firmware`. If you're upgrading from an older release of Debian and you need these firmware binaries, make sure that your `sources.list` uses this new component:

```
http://deb.debian.org/debian trixie main non-free-firmware contrib non-free
```

For more information please refer to [Firmware information](https://wiki.debian.org/Firmware) (<https://wiki.debian.org/Firmware>) in the Debian Wiki.

5.11 У меня есть беспроводная сетевая карта, которая не работает в Linux. Что делать?

Купить ту, которая работает :)

В ином случае используйте пакет `ndiswrapper` для того, чтобы использовать драйвер для Windows (если у вас таковой имеется) в системе Linux. Дополнительную информацию см. на [Debian вики-странице `ndiswrapper`](https://wiki.debian.org/NdisWrapper) (<https://wiki.debian.org/NdisWrapper>).

Глава 6

The Debian archives

6.1 Сколько существует дистрибутивов Debian?

Есть три основных дистрибутива: стабильный («stable»), тестируемый («testing») и нестабильный («unstable»). Тестируемый дистрибутив иногда «замораживается» (смотрите Раздел 6.5.1). Кроме этого, есть ещё старый стабильный дистрибутив («oldstable») (т. е. предыдущий стабильный) и экспериментальный дистрибутив («experimental»).

Experimental is used for packages which are still being developed, and with a high risk of breaking your system. It's used by developers who'd like to study and test bleeding edge software. Users shouldn't be using packages from there, because they can be dangerous and harmful even for the most experienced people.

Чтобы определиться с выбором дистрибутива Debian, см. Глава 3.

6.2 Что означают названия вроде etch, lenny и т. д.?

Это всего лишь «кодовые имена». Когда дистрибутив Debian находится в состоянии разработки, ему присваивается не номер версии, а кодовое имя. Эти имена облегчают зеркалирование дистрибутивов Debian (если бы настоящее имя каталога, например unstable, вдруг изменилось на stable, то пришлось бы без реальной необходимости скачивать кучу пакетов заново).

Сейчас stable — это символическая ссылка на trixie (т. е. на Debian GNU/Linux 13), а testing — это символическая ссылка на forky. Это означает, что trixie сейчас является стабильным дистрибутивом, а forky — текущим тестируемым.

unstable — это постоянная символическая ссылка на sid, так как имя sid навсегда закреплено за нестабильным дистрибутивом (смотрите Раздел 6.3).

6.2.1 Какие кодовые имена уже использовались?

Aside trixie and forky, other codenames that have been already used are: buzz for release 1.1, rex for release 1.2, bo for releases 1.3.x, hamm for release 2.0, slink for release 2.1, potato for release 2.2, woody for release 3.0, sarge for release 3.1, etch for release 4.0, lenny for release 5.0, squeeze for release 6.0, wheezy for release 7, jessie for release 8, stretch for release 9, buster for release 10, bullseye for release 11, bookworm for release 12.

6.2.2 Откуда берутся эти кодовые имена?

Уже давно они выбираются из имён героев мультфильма «История игрушек» (Toy Story) компании Pixar.

- *buzz* (Debian 1.1) was the spaceman Buzz Lightyear,
- *rex* (Debian 1.2) was the tyrannosaurus,
- *bo* (Debian 1.3) was Bo Peep, the girl who took care of the sheep,
- *hamm* (Debian 2.0) was the piggy bank,

- *slink* (Debian 2.1) was Slinky Dog, the toy dog,
- *potato* (Debian 2.2) was, of course, Mr. Potato,
- *woody* (Debian 3.0) was the cowboy,
- *sarge* (Debian 3.1) was the sergeant of the Green Plastic Army Men,
- *etch* (Debian 4.0) was the toy whiteboard (Etch-a-Sketch),
- *lenny* (Debian 5.0) was the toy binoculars,
- *squeeze* (Debian 6) was the name of the three-eyed aliens,
- *wheezy* (Debian 7) was the rubber toy penguin with a red bow tie,
- *jessie* (Debian 8) was the yodeling cowgirl,
- *stretch* (Debian 9) was the rubber toy octopus with suckers on her eight long arms.
- *buster* (Debian 10) was Andy's pet dog.
- *bullseye* (Debian 11) was Woody's wooden toyhorse.
- *bookworm* (Debian 12) was a green toy worm with a built-in flashlight who loves reading books.
- *trixie* (Debian 13) was a blue plastic triceratops.
- *sid* — это имя злого соседского мальчишки, который ломает игрушки.

Решение (<https://lists.debian.org/debian-devel/1996/06/msg00515.html>) использовать имена из *Истории игрушек* было **принято** (<https://lists.debian.org/debian-user/1997/04/msg00011.html>) Брюсом Перенсом, который, будучи Лидером Проекта Debian, работал в мультипликационной компании Pixar (эта компания и создала указанные мультфильмы).

6.3 Что такое «sid»?

sid или *unstable* — это место, куда попадает большая часть пакетов при первоначальной загрузке. Он никогда не будет выпущен, так как пакет сначала должен быть включён в *testing*, а позже в *stable*. *sid* содержит пакеты для выпускаемых и невыпускаемых архитектур.

Имя «*sid*» также взято из мультфильма «История Игрушек»: там Sid — это соседский мальчишка, который ломает игрушки :-)

¹

6.4 Что содержится в каталоге *stable*?

- *stable/main/*: В этом каталоге находятся пакеты, официально составляющие последний выпуск системы Debian GNU/Linux.

Эти пакеты соответствуют **критериям Debian по определению Свободного ПО (DFSG)** (https://www.debian.org/social_contract#guidelines) и могут свободно использоваться и распространяться.

¹When the present-day *sid* did not exist, the FTP site organization had one major flaw: there was an assumption that when an architecture is created in the current *unstable*, it will be released when that distribution becomes the new *stable*. For many architectures that isn't the case, with the result that those directories had to be moved at release time. This was impractical because the move would chew up lots of bandwidth.

The archive administrators worked around this problem for several years by placing binaries for unreleased architectures in a special directory called "sid". For those architectures not yet released, the first time they were released there was a link from the current *stable* to *sid*, and from then on they were created inside the *unstable* tree as normal. This layout was somewhat confusing to users.

With the advent of package pools (see Раздел 6.10), binary packages began to be stored in a canonical location in the pool, regardless of the distribution, so releasing a distribution no longer causes large bandwidth consumption on the mirrors (there is, however, a lot of gradual bandwidth consumption throughout the development process).

- `stable/non-free/`: В этом каталоге находятся пакеты, распространение которых ограничено требованиями, указанными в их лицензии.

Например, в лицензиях некоторых пакетов запрещается их коммерческое распространение. Другие пакеты могут распространяться третьими лицами, но только как условно-бесплатное и несвободное ПО. Перед включением таких пакетов в какие бы то ни было дистрибутивы для дальнейшего распространения (например, на CD-диски) требуется изучение и, возможно, обсуждение лицензии каждого из них.

- `stable/contrib/`: В этом каталоге находятся пакеты, которые свободны в полном соответствии с DFSG и сами по себе *распространяются свободно*, но каким-либо образом зависят от *Несвободных* пакетов, доступных только в разделе `non-free`.

6.5 Что содержится в тестируемом дистрибутиве?

Пакеты попадают в каталог «testing» после того, как пройдут некоторое тестирование в `unstable`.

They must be in sync on all architectures where they have been built and mustn't have dependencies that make them uninstallable; they also need to have fewer release-critical bugs than the versions currently in unstable. This way, we hope that "testing" is always close to being a release candidate.

Более подробную информацию как о «testing» вообще, так и о его пакетах можно найти [здесь](https://www.debian.org/devel/testing) (<https://www.debian.org/devel/testing>).

6.5.1 Что происходит с «testing»? Как его «замораживают»?

When the "testing" distribution is mature enough, the release manager starts "freezing" it. The normal propagation delays are increased to ensure that as few new bugs as possible from "unstable" enter "testing".

Спустя какое-то время дистрибутив «testing» становится по-настоящему «замороженным». Это означает, что новые версии пакетов, за исключением лишь тех, что содержат исправления критических ошибок, больше в «testing» не попадают. В таком же состоянии глубокой заморозки дистрибутив может оставаться на время так называемого «тестового периода» перед выпуском.

Когда дистрибутив «testing» «замораживают», дистрибутив «unstable» также становится частично замороженным. Разработчики перестают добавлять очень новое ПО в нестабильный дистрибутив из-за боязни, что замороженное ПО в «testing» потребует небольших обновлений и исправлений критических для выпуска ошибок, которые не дают «testing» стать «stable».

Для дистрибутива «testing» мы ведём учёт как тех ошибок, что не дают отдельным пакетам попасть в следующий выпуск, так и тех, что и вовсе могут задержать выход новой версии дистрибутива. Подробности см. в описании [текущего состояния тестируемого выпуска](https://www.debian.org/releases/testing/) (<https://www.debian.org/releases/testing/>).

Как только количество ошибок понижается до минимально приемлемого значения, замороженный дистрибутив «testing» объявляется стабильным и выпускается с новым номером версии.

Наиболее важно количество *критических для выпуска* ошибок, за которым можно следить на [странице статуса критических для выпуска ошибок](https://bugs.debian.org/release-critical/) (<https://bugs.debian.org/release-critical/>). Общей целью выпуска является [NoRCBugs](https://wiki.debian.org/ReleaseGoals/NoRCBugs) (<https://wiki.debian.org/ReleaseGoals/NoRCBugs>), что означает, что дистрибутив не должен содержать ошибки степени `critical`, `grave` или `serious`. Полный список проблем, считающихся критическими, можно найти в [Документации по политике RC](https://release.debian.org/testing/rc_policy.txt) (https://release.debian.org/testing/rc_policy.txt).

Каждый раз при выпуске новой версии предыдущий дистрибутив «stable» становится устаревшим и перемещается в архив. Подробности см. на странице [архива Debian](https://www.debian.org/distrib/archive) (<https://www.debian.org/distrib/archive>).

6.6 Что содержится в нестабильном дистрибутиве?

В каталоге «unstable» находится «срез» текущей разработки системы. Пользователи вполне могут использовать и тестировать пакеты оттуда, если они осознают степень их готовности. Преимущество использования нестабильного дистрибутива в том, что у вас всегда будут самые передовые достижения программной индустрии GNU/Linux, ну а если какая программа и сломается, у вас будет целых две половинки :-)

В «unstable» тоже есть подкаталоги main, contrib и non-free, содержащие пакеты, разделённые по тем же признакам что и в «stable».

6.7 What are all those directories at the Debian archives?

На каждом сервере-зеркале Debian всё ПО, которое было упаковано для Debian GNU/Linux, распределено по нескольким каталогам.

Каталог `dists` — это сокращённое название «distributions» (дистрибутивы); это канонический путь доступа к имеющимся на данный момент выпускам Debian (и предварительным выпускам).

В каталоге `pool` содержатся собственно пакеты, см. Раздел 6.10.

Есть несколько вспомогательных каталогов:

/tools/: Утилиты DOS для создания загрузочных дискет, разметки жёсткого диска, сжатия/распаковки файлов и загрузки Linux.

/doc/: Основная документация Debian, например ЧаВо по Debian, инструкции по отправке сообщений об ошибках и т. д.

/indices/: Различные индексные файлы (файл Maintainers и файлы переназначений).

/project/: В основном, материалы для разработчиков и некоторые другие файлы.

6.8 Что находится в каталогах внутри `dists/stable/main`?

Within each of the major directory trees², there are three sets of subdirectories containing index files.

Есть набор подкаталогов `binary-что-то` с индексными файлами для двоичных пакетов для каждой доступной компьютерной архитектуры, например `binary-i386` — для пакетов, собранных для компьютеров Intel x86, или `binary-sparc` — для машин Sun SPARCStation.

Полный список доступных архитектур для каждого выпуска есть на [веб-странице выпуска](https://www.debian.org/releases) (<https://www.debian.org/releases>). Для списка текущего выпуска см. Раздел 4.1.

Индексные файлы в `binary-*` называются Packages(.gz, .bz2) и включают сводку по каждому двоичному пакету, вошедшему в дистрибутив. Реальные файлы двоичных пакетов находятся на самом верхнем уровне дерева в [каталоге pool](#).

Кроме этого, есть подкаталог с именем `source/`, в котором находятся индексные файлы вошедших в дистрибутив пакетов с исходными кодами. Индексный файл называется Sources(.gz, .bz2).

И последнее, но не менее важное: есть набор подкаталогов, предназначенных для индексных файлов системы установки, они называются `debian-installer/binary-архитектура`.

6.9 Где находятся исходные коды?

Исходные коды в Debian есть для каждого пакета. Более того, в лицензиях большинства программ содержатся *требования* распространять вместе с ними и их исходные коды, либо сопровождать их предложением о том, откуда их можно получить.

The source code is distributed in the `pool` directory (see Раздел 6.10) together with all the architecture-specific binary directories. To retrieve the source code without having to be familiar with the structure of the archive, try a command like `apt-get source mypackagename`.

Из-за ограничения в лицензиях исходный код пакетов в разделах "contrib" и "non-free" либо может, либо не может быть доступен, формально эти разделы не являются частью системы Debian. Двоичные файлы, не имеющие исходного кода, могут распространяться лишь в некоторых случаях (напр., см.: `firmware-misc-nonfree`); в других случаях лицензия запрещает распространение собранных двоичных файлов, но разрешает распространение пакетов с исходным кодом, которые могут быть скомпилированы пользователями самостоятельно (см.: `broadcom-sta-dkms`).

²`dists/stable/main, dists/stable/contrib, dists/stable/non-free, and dists/unstable/main/, etc.`

6.10 Что находится в каталоге pool?

Пакеты содержатся в огромном «пуле», структурированном по именам пакетов с исходным кодом. Для большей управляемости пул разбит по разделам («main», «contrib» и «non-free») и по первым буквам имён пакетов с исходным кодом. В этих каталогах содержится по несколько файлов: двоичные пакеты для каждой архитектуры и пакеты с исходным кодом, из которых были собраны двоичные пакеты.

Чтобы выяснить, где находится определённый пакет, можно воспользоваться командой `apt-cache showsrc имя_пакета` и посмотреть на строку «Directory:». Например, пакеты `apache` находятся в `pool/main/a/apache/`.

Также, из-за того, что пакетов, начинающихся с `lib*`, слишком много, они распределены чуть по-другому: например, пакеты `libpaper` находятся в `pool/main/libp/libpaper/`.

³

6.11 Для чего нужен каталог «incoming»?

После того как разработчик закачал пакет, до того, как он будет проверен на подлинность, и ему будет разрешено попасть в архив, он какое-то время хранится в каталоге входящих пакетов «incoming»

Обычно никто не должен устанавливать пакеты из этого каталога. Однако, на случай крайней необходимости, каталог `incoming` доступен по адресу <https://incoming.debian.org/>. Оттуда можно вручную выкачать нужные пакеты, проверить подпись GPG и контрольную сумму MD5 в файлах `.changes` и `.dsc` и установить их.

6.12 Where can I find old packages of previous releases?

Old releases are removed from the main archive and mirrors, which only keep the content of the releases up to "oldstable" (the stable release before the current one). If you are interested in obtaining older versions of packages, go to <https://snapshot.debian.org/>.

The snapshot archive is a wayback machine that allows access to old packages based on dates and version numbers. It consists of all past and current packages the Debian archive provides. It provides a valuable resource for tracking down when regressions were introduced, or for providing a specific environment that a particular application may require to run. The snapshot archive is accessible like any normal apt repository, allowing it to be easily used by all.

6.13 Как создать свой собственный apt-репозиторий?

Если собрали какие-то свои собственные пакеты Debian, которые вы хотели бы установить, используя стандартные инструменты по управлению пакетами Debian, вы можете настроить свой собственный архив пакетов с поддержкой apt. Это также полезно в том случае, если вы хотите поделиться своими пакетами Debian, даже несмотря на то, что они не распространяются Проектом Debian. Инструкции о том, как это сделать приведены в [вики Debian](https://wiki.debian.org/HowToSetupADebianRepository) (<https://wiki.debian.org/HowToSetupADebianRepository>).

³Historically, packages were kept in the subdirectory of `dists` corresponding to which distribution contained them. This turned out to cause various problems, such as large bandwidth consumption on mirrors when major changes were made. This was fixed with the introduction of the package pool.

The `dists` directories are still used for the index files used by programs like `apt`.

Глава 7

Основы системы управления пакетами Debian

В этой главе вкратце рассматривается внутренняя низкоуровневая организация системы управления пакетами Debian. Если вас главным образом интересует вопрос о том, как *использовать* соответствующие утилиты, переходите сразу к Глава 8 или Глава 9.

7.1 Что такое пакет Debian?

Пакет обычно содержит полный комплект файлов, необходимых для реализации определённого набора команд или возможностей. Существует два типа пакетов Debian:

- *Binary packages*, which contain executables, configuration files, man/info pages, copyright information, and other documentation. These packages are distributed in a Debian-specific archive format (see Раздел 7.2); they are usually characterized by having a '.deb' file extension. Binary packages can be unpacked using the Debian utility `dpkg` (possibly via a frontend like `apt`); details are given in its manual page.
- *Source packages*, which consist of a `.dsc` file describing the source package (including the names of the following files), a `.orig.tar.gz` file that contains the original unmodified source in gzip-compressed tar format and usually a `.debian.tar.xz` file that contains the Debian-specific changes to the original source. The utility `dpkg-source` packs and unpacks Debian source archives; details are provided in its manual page. (The program `apt-get` can be used as a frontend for `dpkg-source`.)

Installation of software by the package system uses "dependencies" which are carefully designed by the package maintainers. These dependencies are documented in the `control` file associated with each package. For example, the package containing the GNU C compiler (`gcc`) "depends" on the package `binutils` which includes the linker and assembler. If a user attempts to install `gcc` without having first installed `binutils`, the package management system (`dpkg`) will send an error message that it also needs `binutils`, and stop installing `gcc`. (However, this facility can be overridden by the insistent user, see `dpkg(8)`.) See more in Раздел 7.9 below.

Инструменты управления пакетами Debian могут использоваться для:

- манипулирования и управления пакетами или их частями;
- управления локальными заменами файлов пакета;
- помощи разработчикам в сборке пакетов;
- aid users in the installation of packages which reside on a remote archive site.

7.2 Какой формат у двоичных пакетов Debian?

A Debian "package", or a Debian archive file, contains the executable files, libraries, and documentation associated with a particular program or set of related programs. Normally, a Debian archive file has a filename that ends in `.deb`.

The internals of this Debian binary packages format are described in the `deb(5)` manual page. This internal format is subject to change (between major releases of Debian GNU/Linux), therefore please always use `dpkg-deb(1)` if you need to do lowlevel manipulations on `.deb` files.

7.3 Почему имена файлов пакетов Debian такие длинные?

The Debian binary package file names conform to the following convention: `<DebianPackageName>_<VersionNumber>-<DebianRevisionNumber>_<DebianArchitecture>.deb`

Checking the package name associated with a particular Debian archive file (`.deb` file) can be done in one of these ways:

- inspect the "Packages" file in the directory where it was stored at a Debian archive site. This file contains a stanza describing each package; the first field in each stanza is the formal package name.
- use the command `dpkg --info PPP_VVV-RRR_AAA.deb` (where PPP, VVV, RRR and AAA are the package name, version, revision and architecture of the package in question, respectively). This displays, among other things, the package name corresponding to the archive file being unpacked.

Компонент VVV определяет номер версии, установленный разработчиком программы. Его формат не стандартизован, поэтому номер версии может быть любым, например «19990513» или «1.3.8pre1».

The RRR component is the Debian revision number, and is specified by the Debian developer (or a user who chooses to rebuild the package locally). This number corresponds to the revision level of the Debian package, thus, a new revision level usually signifies changes in the Debian Makefile (`debian/rules`), the Debian control file (`debian/control`), the installation or removal scripts (`debian/p*`), or in the configuration files used with the package.

The AAA component identifies the processor for which the package was built. This is commonly `amd64`, which refers to AMD64, Intel 64 or VIA Nano chips. For other possibilities review Debian's archive directory structure at Раздел 6.7. For details, see the description of "Debian architecture" in the manual page `dpkg-architecture(1)`.

7.4 Зачем нужен файл control?

Содержимое файла `control` подробно рассматривается в разделе 5 «Руководства по политике Debian» (Debian Policy Manual) (см. Раздел 12.1).

Краткий пример файла `control` для пакета Debian `hello` приведён ниже:

```
Package: hello
Version: 2.9-2+deb8u1
Architecture: amd64
Maintainer: Santiago Vila <sanvila@debian.org>
Installed-Size: 145
Depends: libc6 (>= 2.14)
Conflicts: hello-traditional
Breaks: hello-debhelper (<< 2.9)
Replaces: hello-debhelper (<< 2.9), hello-traditional
Section: devel
Priority: optional
Homepage: https://www.gnu.org/software/hello/
Description: example package based on GNU hello
 The GNU hello program produces a familiar, friendly greeting. It
 allows non-programmers to use a classic computer science tool which
 would otherwise be unavailable to them.
```

```
Seriously, though: this is an example of how to do a Debian package.
It is the Debian version of the GNU Project's "hello world" program
(which is itself an example for the GNU Project).
```

Поле `Package` содержит имя пакета. Это имя, по которому инструменты управления пакетами будут его опознавать. Обычно (но не обязательно) оно совпадает с первым компонентом имени файла архива Debian.

Поле `Version` содержит номер версии программы, установленный её разработчиками, и (в последнем компоненте) номер ревизии пакета этой программы в Debian, см. Раздел 7.3.

Поле `Architecture` определяет тип процессора, для которого были скомпилированы двоичные файлы в данном пакете.

Поле `Depends` содержит список пакетов, которые должны быть установлены для успешной установки данного пакета.

`Installed-Size` отражает размер дискового пространства, который будет занят пакетом после установки. Этот параметр может использоваться программами установки для проверки достаточности дискового пространства перед установкой пакета.

The `Section` line gives the "section" where this Debian package is stored at the Debian archive sites.

Поле `Priority` показывает, насколько установка этого пакета важна для системы; некоторые программы, например, `art` или `artitude`, могут сортировать пакеты по категориям (напр., поместить пакет в список необязательных пакетов), см. Раздел 7.7.

В поле `Maintainer` указан адрес электронной почты человека, ответственного за поддержку данного пакета.

В поле `Description` приводится краткое описание функциональности пакета.

Более подробную информацию о всех возможных полях управляющего файла пакета см. в разделе 5 («Управляющие файлы и их поля») «Руководства по политике Debian» (Debian Policy Manual), см. Раздел 12.1.

7.5 Зачем нужен файл conffile?

`Conffile` содержит список файлов настроек (обычно помещаемых в `/etc`), которые при обновлении пакета не будут перезаписываться системой управления пакетами. Это гарантирует, что содержимое файлов настроек будет сохранено, и позволяет обновлять пакеты, не прерывая работу системы.

Чтобы точно определить, какие файлы сохраняются при обновлении, запустите:

```
dpkg --status b'pb''b''ab''b''kb''b''eb''b''tb''
```

и взгляните на строку «`Conffiles:`».

7.6 Зачем нужны сценарии preinst, postinst, prerm и postrm?

Это исполняемые сценарии, автоматически запускаемые до или после установки или удаления пакета. Вместе с файлом `control` эти файлы являются частью «управляющего» раздела архивного файла Debian.

Более подробно:

preinst This script is executed before the package it belongs to is unpacked from its Debian archive (".deb") file. Many 'preinst' scripts stop services for packages which are being upgraded until their installation or upgrade is completed (following the successful execution of the 'postinst' script).

postinst This script typically completes any required configuration of the package `foo` once `foo` has been unpacked from its Debian archive (".deb") file. Many 'postinst' scripts execute any commands necessary to start or restart a service once a new package has been installed or upgraded.

prerm Этот сценарий обычно служит для останова всех связанных с пакетом служб. Он выполняется перед удалением файлов данного пакета.

postrm Этот сценарий обычно служит для изменения ссылок или других файлов, связанных с `foo`, и/или удаления файлов, созданных пакетом. (См. также Раздел 7.8.)

Currently all of the control files can be found in the directory `/var/lib/dpkg/info`. The files relevant to package `foo` begin with the name "foo" and have file extensions of "preinst", "postinst", etc., as appropriate. The file `foo.list` in that directory lists all of the files that were installed with the package `foo`. (Note that the location of these files is a dpkg internal; you should not rely on it.)

7.7 Что такое *Пакет первой необходимости (Essential)*, *Необходимый (Required)*, *Важный (Important)*, *Стандартный (Standard)*, *Необязательный (Optional)* или *Дополнительный (Extra)* пакет?

Для поддержки системы управления пакетами каждому пакету в Debian сопровождающими дистрибутива назначается *приоритет*. Возможные приоритеты:

- **Необходимые (Required)** — пакеты, необходимые для правильного функционирования системы.

Сюда входят все инструменты, необходимые для устранения неполадок в системе. Вам не следует удалять эти пакеты, иначе ваша система может перестать работать, и не исключено, что вы даже не сможете использовать `dpkg` для того, чтобы вернуть всё назад. Функциональность системы, в которой установлены только Необходимые пакеты, не слишком высока, но достаточна для того, чтобы позволить системному администратору загрузить её и установить больше программного обеспечения.

- **Важные (Important)** — пакеты, которые должны быть в любой Unix-системе.

Other packages which the system will not run well or be usable without will be here. This does *NOT* include Emacs or X or TeX or any other large application. These packages only constitute the bare infrastructure.

- **Standard** packages are standard on any Linux system, including a reasonably small but not too limited character-mode system. Tools are included to be able to send e-mail (with mutt) and download files from archive servers.

This is what will be installed by default if users do not select anything else. It does not include many large applications, but it does include the Python interpreter and some server software like OpenSSH (for remote administration) and Exim (for mail delivery, although it can be configured for local delivery only). It also includes some common generic documentation that most users will find helpful.

- **Optional** packages include all those that you might reasonably want to install if you do not know what they are, or that do not have specialized requirements.

Сюда входят X, полный дистрибутив TeX и множество других приложений.

- **Дополнительные (Extra)** — пакеты, либо конфликтующие с другими пакетами, имеющими более высокий приоритет, полезные, скорее всего, только в том случае, когда вы уже знаете, что это такое, либо имеющие специфические требования, из-за которых им нельзя дать приоритет «Необязательный».

Если вы выполните установку Debian по умолчанию, то будут установлены все пакеты с приоритетом **Стандартный** или выше. Если вы выберете какие-то определённые задачи, то также будут установлены и пакеты с более низким приоритетом.

Кроме того, некоторые пакеты классифицированы как **Пакеты первой необходимости (Essential)**, так как они абсолютно необходимы для правильной работы системы. Инструменты управления пакетами не допустят их удаления.

7.8 Что такое виртуальный пакет?

A virtual package is a generic name that applies to any one of a group of packages, all of which provide similar basic functionality. For example, both the `konqueror` and `firefox-esr` programs are web

browsers, and should therefore satisfy any dependency of a program that requires a web browser on a system, in order to work or to be useful. They are therefore both said to provide the "virtual package" called `www-browser`.

Similarly, `exim4` and `sendmail` both provide the functionality of a mail transport agent. They are therefore said to provide the virtual package "mail-transport-agent". If either one is installed, then any program depending on the installation of a `mail-transport-agent` will be satisfied by the presence of this virtual package.

Кроме того, в Debian есть механизм, позволяющий системному администратору в том случае, когда в системе установлено несколько пакетов, предоставляющих определённый виртуальный пакет, выбрать предпочтительный. Для этого служит команда `update-alternatives`, см. Раздел 11.11.

7.9 Что имеется в виду, когда говорят, что пакет *Рекомендует* (Recommends), *Предлагает* (Suggests), *Заменяет* (Replaces), *Ломает* (Breaks) или *Предоставляет* (Provides) другой пакет, *Зависит* (Depends) от него или *Конфликтует* (Conflicts) с ним?

В системе пакетов Debian есть несколько типов «зависимостей» пакетов друг от друга, задуманных для определения (в одной переменной) степени независимости одной программы (например, А) от наличия в данной системе другой (Б).

- Пакет А *зависит* от пакета Б, если Б абсолютно необходим для работы А. В некоторых случаях А не просто зависит от Б, но дополнительно требует определённую версию Б. В этом случае обычно накладывается требование, чтобы версия Б была не ниже заданной.
- Пакет А *рекомендует* пакет Б, если сопровождающий пакета считает, что большинство пользователей не захотят пользоваться А, не имея функциональности, предоставляемой пакетом Б.
- Пакет А *предлагает* пакет Б, если Б содержит файлы, имеющие отношение к функциональности пакета А (и обычно её расширяющие).
- Пакет А *конфликтует* с пакетом Б, когда А не может работать, если установлен пакет Б. Наиболее часто конфликты возникают, когда А содержит усовершенствованные версии файлов, содержащихся в Б. «Конфликтует» часто задаётся вместе с «заменяет».
- Пакет А *заменяет* пакет Б, когда файлы, установленные из пакета Б, удаляются и (в некоторых случаях) замещаются файлами из А.
- Пакет А *ломает* пакет Б, когда нельзя одновременно настроить оба пакета в системе. Система управления пакетами предотвратит установку одного, если в системе уже установлен и настроен другой.
- Пакет А *предоставляет* пакет Б, когда все файлы и функциональность Б имеются в А. Этот механизм позволяет пользователям с ограниченным дисковым пространством получить только ту часть пакета А, которая действительно им нужна.

More detailed information on the use of each of these terms can be found in the Debian Policy manual, section 7.2, "Binary Dependencies", see Раздел 12.1.

7.10 Что значит Пред-зависит (Pre-Depends)?

"Pre-Depends" is a special dependency. In the case of most packages, `dpkg` will unpack the archive file of a package (i.e., its `.deb` file) independently of whether or not the files on which it depends exist on the system. Simplistically, unpacking means that `dpkg` will extract the files from the archive file that were meant to be installed on your file system, and put them in place. If those packages *depend* on

the existence of some other packages on your system, `dpkg` will refuse to complete the installation (by executing its "configure" action) until the other packages are installed.

Однако, некоторые пакеты `dpkg` даже не будет распаковывать, пока не будут разрешены некоторые зависимости. Про такие пакеты говорят, что они имеют «предварительную зависимость» от наличия некоторых других пакетов. Этот механизм предоставляется в Debian для поддержки безопасного перехода систем с формата `a.out` на ELF, когда критична *очерёдность* распаковки пакетов. Существуют и другие варианты больших обновлений, где этот приём также полезен, например для пакетов с приоритетом «необходимый», когда они зависят от LibC.

Опять же, более подробную информацию об этом можно найти в руководстве по политике.

7.11 Что означают слова *неизвестно (unknown)*, *установить (install)*, *удалить (remove)*, *вычистить (purge)*, *зафиксировать (hold)* в строке состояния пакета?

Эти флаги определяют, что пользователь «хочет» сделать с пакетом (что определяется вызовами `dpkg/apt/aptitude`).

Их значения:

- *неизвестно (unknown)* — пользователь никоим образом не отметил, нужен ли ему этот пакет.
- *установить (install)* — пользователь хочет установить или обновить пакет;
- *remove* - the user wants the package removed, but does not want to remove any existing configuration file.
- *вычистить (purge)* — пользователь хочет удалить пакет полностью, включая его файлы настроек;
- *зафиксировать (hold)* — пользователь хочет, чтобы над пакетом не совершалось никаких действий, т. е. он хочет сохранить текущую версию пакета, в каком бы состоянии она ни была.

7.12 Как зафиксировать (hold) пакет?

Есть три способа перевода пакета в зафиксированное состояние: с помощью `dpkg`, `apt` или `aptitude`.

При использовании `dpkg` вам нужно экспортировать список состояний отметки пакетов:

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

Затем отредактировать полученный файл `selections.txt`, заменив строку с именем пакета, который нужно зафиксировать, например `libc6`, с:

```
libc6                                install
```

на:

```
libc6                                hold
```

Сохранить файл и загрузить его в базу данных `dpkg`:

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

С помощью `apt` пакет можно зафиксировать командой

```
apt-mark hold b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

а снять фиксацию с помощью

```
apt-mark unhold b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

При использовании `aptitude` пакет можно зафиксировать командой

```
aptitude hold b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

а снять фиксацию с помощью

```
aptitude unhold b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

7.13 Как установить пакет исходного кода?

Пакеты исходного кода Debian на самом деле нельзя «установить», они просто распаковываются в любой указанный вами каталог для сборки двоичного пакета.

Source packages are distributed on most of the same mirrors where you can obtain the binary packages. If you set up your APT's sources.list(5) to include the appropriate "deb-src" lines, you'll be able to easily download any source package by running

```
apt-get source b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

To help you in actually building the source package, Debian source packages provide the so-called build-dependencies mechanism. This means that the source package maintainer keeps a list of other packages that are required to build their package. To see how this is useful, run

```
apt-get build-dep b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b'' ←
'ab''
```

перед сборкой пакета.

7.14 Как собрать двоичный пакет из пакета исходного кода?

Лучше всего это делать с помощью различных утилит-обёрток. Мы покажем как использовать инструментарий devscripts. Установите этот пакет, если это ещё не сделано.

Сначала добудьте пакет с исходным кодом:

```
apt-get source b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

и перейдите в дерево исходников:

```
cd b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''-*
```

Затем установите необходимые сборочные зависимости (при их наличии):

```
sudo apt-get build-dep b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b'' ←
''ab''
```

После этого создайте отдельную версию своей сборки (для того, чтобы позже не удивляться, когда в Debian тоже выйдет новая версия):

```
dch -l local 'Blah blah blah'
```

И, наконец, соберите пакет:

```
debuild -us -uc
```

Если всё прошло успешно, то вы сможете установить свой пакет, запустив

```
sudo dpkg -i ../*.deb
```

Если вы предпочитаете делать всё вручную и не хотите использовать devscripts, то делайте так:

Для компиляции исходного кода вам понадобятся файлы имя_пакета_*.dsc, имя_пакета_*.tar.gz и имя_пакета_*.debian.tar.xz (учтите, что для некоторых родных пакетов Debian файла .debian.tar.xz нет).

Если у вас есть эти файлы (см. Раздел 7.13) и установлен пакет dpkg-dev, то следующая команда:

```
dpkg-source -x b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''_b ←
''вб''b''еб''b''рб''b''сб''b''иб''b''яб''-b''рб''b''еб''b''вб''b''иб''b''зб'' ←
b''иб''b''яб''_b''аб''b''рб''b''кб''b''иб''b''тб''b''еб''b''кб''b''тб''b''yb'' ←
b''рб''b''аб''.dsc
```

извлечёт пакет в каталог с именем имя_пакета-версия.

If you just want to compile the package, you may cd into the foo-version directory and issue the command

```
dpkg-buildpackage -rfakeroot -b
```

для сборки пакета (обратите внимание, что для этого также понадобится пакет fakeroot), а затем

```
dpkg -i ../b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''_b' ←
''вб''b''еб''b''рб''b''сб''b''иб''b''яб''-b''рб''b''еб''b''вб''b''иб''b''зб''b' ←
''иб''b''яб''_b''аб''b''рб''b''кб''b''иб''b''тб''b''еб''b''кб''b''тб''b''yb'' ←
b''рб''b''аб''.deb
```

для установки только что собранного пакета.

7.15 Как самому создать пакет Debian?

For a more detailed description on this, read the New Maintainers' Guide, available in the maint-guide package or at <https://www.debian.org/doc/devel-manuals#maint-guide>, or the Guide for Debian Maintainers, available in the debmake-doc package or at <https://www.debian.org/doc/devel-manuals#debmake-doc>.

Глава 8

Инструменты управления пакетами Debian

8.1 Какие программы для управления пакетами имеются в Debian?

В Debian для управления пакетами имеется множество средств, от программ с графическими или текстовыми интерфейсами, до низкоуровневых утилит установки пакетов. Корректная работа всех доступных инструментов зависит от низкоуровневых утилит, и все они представлены здесь в порядке уменьшения уровня сложности.

Важно понимать, что высокоуровневые инструменты управления пакетами, такие как **aptitude** или **synaptic**, для управления пакетами используют **apt**, который, в свою очередь, использует **dpkg** для управления пакетами системы.

See [Chapter 2. Debian package management](https://www.debian.org/doc/manuals/debian-reference/ch02.en.html) (<https://www.debian.org/doc/manuals/debian-reference/ch02.en.html>) of the [Debian reference](https://www.debian.org/doc/manuals/debian-reference/) (<https://www.debian.org/doc/manuals/debian-reference/>) for more information about the Debian package management utilities. This document is available in various languages and formats, see [the Debian Reference entry in the DDP Users' Manuals overview](https://www.debian.org/doc/user-manuals#quick-reference) (<https://www.debian.org/doc/user-manuals#quick-reference>).

8.1.1 dpkg

Это основная программа управления пакетами. **dpkg** может вызываться с многими параметрами. Наиболее часто используемые из них:

- Показать список всех параметров: `dpkg --help`
- Показать управляющий файл (и другую информацию) для указанного пакета: `dpkg --info foo_VVV-RRR.deb`
- Установить пакет на жёсткий диск (т. е. распаковать и настроить): `dpkg --install foo_VVV-RRR.deb`
- Распаковать архив Debian на жёсткий диск (но не настраивать): `dpkg --unpack foo_VVV-RRR.deb`. Учтите, что в результате данной операции пакет *не обязан* быть в рабочем состоянии; для правильной работы может потребоваться внесение изменений в некоторые файлы. Данная команда удаляет любую ранее установленную версию программы и запускает сценарий `preinst` указанного пакета (см. Раздел 7.6).
- Настроить пакет, который был распакован ранее: `dpkg --configure foo`. Кроме всего прочего, эта команда запускает сценарий `postinst` указанного пакета (см. Раздел 7.6). Она также обновляет файлы, перечисленные в `conffiles`. Обратите внимание, что в качестве аргумента для параметра `configure` указывается имя пакета (т. е. `foo`), а *не* имя файла-архива Debian (т. е. `foo_VVV-RRR.deb`).
- Распаковать файл с именем "blurf" (или группу файлов с именем "blurf*") из архива Debian: `dpkg --fsys-tarfile foo_VVV-RRR.deb | tar -xf - 'blurf*'`
- Удалить пакет (но не его файлы настроек): `dpkg --remove foo`

- Удалить пакет (вместе с файлами настроек): `dpkg --purge foo`
- Вывести состояние установки пакетов, содержащих в имени строку (или регулярное выражение) «foo*»: `dpkg --list 'foo*'`

8.1.2 APT

APT is the *Advanced Package Tool*, an advanced interface to the Debian packaging system which provides the `apt-get` program. It provides commandline tools for searching and managing packages, and for querying information about them, as well as low-level access to all features of the `libapt-pkg` library. For more information, see the User's Guide in `/usr/share/doc/apt-doc/guide.html/index.html` (you will have to install the `apt-doc` package).

Starting with Debian Jessie, some frequently used `apt-get` and `apt-cache` commands have an equivalent via the new `apt` binary. This means some popular commands like `apt-get update`, `apt-get install`, `apt-get remove`, `apt-cache search`, or `apt-cache show` now can also be called simply via `apt`, say `apt update`, `apt install`, `apt remove`, `apt search`, or `apt show`. The following is an overview of the old and new commands:

```
apt-get update           -> apt update
apt-get upgrade         -> apt upgrade
apt-get dist-upgrade    -> apt full-upgrade
apt-get install package -> apt install package
apt-get remove package  -> apt remove package
apt-get autoremove      -> apt autoremove
apt-cache search string -> apt search string
apt-cache policy package -> apt list -a package
apt-cache show package  -> apt show package
apt-cache showpkg package -> apt show -a package
```

Инструмент `apt` совмещает функциональность `apt-get` и `apt-cache`, а также по умолчанию использует красивый цветной формат вывода, что очень удобно. Для использования в сценариях или для продвинутого использования предпочтительнее использовать `apt-get` (а иногда он просто необходим).

`apt-get` provides a simple way to retrieve and install packages from multiple sources using the command line. Unlike `dpkg`, `apt-get` does not understand `.deb` files, it works with the packages proper name and can only install `.deb` archives from a source specified in `/etc/apt/sources.list`. `apt-get` will call `dpkg` directly after downloading the `.deb` archives¹ from the configured sources.

Часто используемые команды `apt-get`:

- To update the list of packages known by your system, you can run:

```
apt update
```

(вы должны регулярно запускать эту команду для обновления списка пакетов)

- Установить пакет `foo` и все его зависимости:

```
apt install foo
```

- Удалить пакет из системы:

```
apt remove foo
```

- Удалить из системы пакет и все его файлы настроек:

```
apt source b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

- To list all packages for which newer versions are available, run:

```
apt dist-upgrade
```

¹Notice that there are ports that make this tool available with other package management systems, like Red Hat package manager, also known as `rpm`

- Обновить все пакеты в системе (без установки дополнительных пакетов или удаления пакетов):

```
apt upgrade
```

- Обновить все установленные в системе пакеты с установкой или удалением дополнительных пакетов, если это потребуется для обновления какого-то пакета:

```
apt full-upgrade
```

(Команда `upgrade` оставит старую установленную версию пакета, если для разрешения новых зависимостей при обновлении потребуется установка дополнительных пакетов. Команда `full-upgrade` менее консервативна.)

Note that you must be logged in as root to perform any commands that modify packages.

Note that **apt-get** now also installs recommended packages as default, and thanks to its robustness it's the preferred program for package management from console to perform system installation and major system upgrades.

В комплект инструментов `apt` входит также программа для обработки запросов по списку пакетов **apt-cache**. Её можно использовать для поиска пакетов, имеющих определённую функциональность, с помощью простых текстовых запросов или регулярных выражений, а также для получения списка зависимостей из системы управления пакетами. Часто используемые команды **apt-cache**:

- Найти пакеты, содержащие в своём описании *слово*:

```
apt search <i>b''cb''b''pb''b''ob''b''vb''b''ob''</i>
```

- Показать подробную информацию о пакете:

```
apt show <i>b''pb''b''ab''b''xb''b''eb''b''tb''</i>
```

- Показать зависимости пакета:

```
apt-cache depends <i>b''pb''b''ab''b''xb''b''eb''b''tb''</i>
```

- To print detailed information on the versions available for a package and the packages that reverse-depend on it:

```
apt-cache showpkg <i>b''pb''b''ab''b''xb''b''eb''b''tb''</i>
```

For more information, install the `apt` package and read `apt(8)`, `apt-get(8)`, `sources.list(5)` and install the `apt-doc` package and read `/usr/share/doc/apt-doc/guide.html/index.html`.

8.1.3 aptitude

aptitude — это менеджер пакетов для систем Debian GNU/Linux, он предоставляет интерфейс к инфраструктуре управления пакетами `apt`. **aptitude** представляет собой текстовый интерфейс на основе библиотеки `curses`. Действия можно выполнять как из визуального интерфейса, так и из командной строки.

aptitude может использоваться для лёгкого и быстрого выполнения задач по управлению пакетами. Она позволяет пользователю просматривать список пакетов и выполнять такие задачи по управлению пакетами как установка, обновление и удаление.

Помимо функциональности **apt-get**, **aptitude** имеет много других дополнительных возможностей:

- обеспечивает лёгкий доступ ко всем версиям пакета;
- позволяет легко отслеживать устаревшее ПО, заноса его в «список устаревших пакетов и пакетов, созданных локально»;

- включает достаточно мощную систему поиска и ограничения отображаемых пакетов. Пользователи, знакомые с **mutt**, освоятся быстро, так как синтаксис регулярных выражений был навеян этой программой;
- можно использовать для установки заранее сформированных наборов для определённых задач. Подробности см. в Раздел 8.1.5;
- **aptitude** in full screen mode has **su** functionality embedded and can be run by a normal user. It will call **su** (and ask for the root password, if any) when you really need administrative privileges.

С **aptitude** можно работать, используя визуальный интерфейс (просто запустить `aptitude`), или непосредственно из командной строки. Используемый синтаксис командной строки очень похож на синтаксис **apt-get**. Например, для установки пакета `foo`, можно выполнить `aptitude install foo`.

Note that **aptitude** is the preferred program for daily package management from the console.

For more information, read the manual page `aptitude(8)` and install the `aptitude-doc` package.

8.1.4 synaptic

synaptic is a graphical package manager. It enables you to install, upgrade and remove software packages in a user friendly way. Along with most of the features offered by **aptitude**, it also has a feature for editing the list of used repositories, and supports browsing all available documentation related to a package. See the [Synaptic Website](https://www.nongnu.org/synaptic/) (<https://www.nongnu.org/synaptic/>) for more information.

8.1.5 tasksel

Иногда бывает трудно найти подходящий комплект пакетов для выполнения определённой задачи. Разработчики Debian определили задачи, представляющие собой наборы из нескольких пакетов Debian, предназначенных для определённой деятельности. Задачи можно устанавливать с помощью программы **tasksel** или **aptitude**.

Typically, the Debian installer will automatically install the task associated with a standard system and a desktop environment. The specific desktop environment installed will depend on the CD/DVD media used, most commonly it will be the GNOME desktop (`gnome-desktop` task). Also, depending on your selections throughout the installation process, tasks might be automatically installed in your system. For example, if you selected a language other than English, the task associated with it will be installed automatically too.

8.1.6 Другие инструменты управления пакетами

8.1.6.1 dpkg-deb

This program manipulates Debian archive (`.deb`) files. Some common uses are:

- Вывести список допустимых параметров: `dpkg-deb --help`
- Определить, какие файлы содержатся в файле-архиве Debian: `dpkg-deb --contents foo_VVV-RRR.deb`
- Извлечь файлы из указанного архива Debian в определённый пользователем каталог: `dpkg-deb --extract foo_VVV-RRR.deb tmp` извлечёт все файлы из `foo_VVV-RRR.deb` в каталог `tmp/`. Это удобно для просмотра содержимого пакета в отдельном каталоге без его установки в основное дерево каталогов.
- Извлечь из пакета файлы с управляющей информацией: `dpkg-deb --control foo_VVV-RRR.deb tmp`.

Учтите, что любые пакеты, просто распакованные командой `dpkg-deb --extract`, будут установлены некорректно, для установки следует использовать `dpkg --install`.

More information is given in the manual page `dpkg-deb(1)`.

8.2 Говорят, что Debian способен обновить работающую программу; как это делается?

Ядро (файловая система) в системах Debian GNU/Linux поддерживает замену файлов, даже когда они используются.

Мы также предоставляем программу **start-stop-daemon**, которая используется для запуска служб при загрузке компьютера или их останова при изменении уровня выполнения (например, при переключении из многопользовательского в однопользовательский или для выключения компьютера). Эта же программа используется сценариями установки при установке нового пакета со службой, для остановки работающей службы и её перезапуска при необходимости.

8.3 Как узнать, какие пакеты установлены в системе Debian?

Чтобы получить список всех установленных пакетов, выполните команду

```
dpkg --list
```

Эта команда выведет для каждого пакета однострочную сводку, включающую два символа состояния (объясняемые в заголовке), имя пакета, *установленную* версию и краткое описание.

To learn the status of packages whose names match any pattern beginning with "foo", run the command:

```
dpkg --list 'foo*'
```

Чтобы получить более подробный отчёт о состоянии определённого пакета, выполните команду:

```
dpkg --status b''иб''b''мб''b''яб''_b''пб''b''аб''b''кб''b''еб''b''тб''b''аб''
```

8.4 How do I display the files of an installed package?

Чтобы вывести список файлов установленного пакета `foo`, выполните команду

```
dpkg --getfiles foo
```

Учтите, что файлы, созданные сценариями установки, не отображаются.

8.5 Как определить пакет, которому принадлежит определённый файл?

Чтобы определить, в каком пакете содержится файл с именем `foo`, выполните одну из следующих команд:

- `dpkg --search foo`

Эта команда ищет `foo` в установленных пакетах. (В настоящий момент это эквивалентно поиску всех файлов с расширением `.list` в каталоге `/var/lib/dpkg/info/` и выводу имён всех пакетов, которые содержат заданное имя файла, и отклонений.)

Более быстрая альтернатива этому — программа **dlocate**.

```
dlocate -S foo
```

- `zgrep foo Contents-ARCH.gz`

This searches for files which contain the substring `foo` in their full path names. The files `Contents-ARCH.gz` (where ARCH represents the wanted architecture) reside in the major package directories (`main`, `non-free`, `contrib`) at a Debian archive site (i.e. under `/debian/dists/trixie`). A `Contents` file refers only to the packages in the subdirectory tree where it resides. Therefore, a user might have to search more than one `Contents` files to find the package containing the file `foo`.

Преимущество этого метода над `dpkg --search` состоит в том, что будут найдены файлы в тех пакетах, которые могут быть не установлены в вашей системе.

- `apt-file search foo`

If you install the `apt-file` package, similar to the above, it searches files which contain the substring or regular expression `foo` in their full path names. The advantage over the example above is that there is no need to retrieve the `Contents-ARCH.gz` files as it will do this automatically for all the sources defined in `/etc/apt/sources.list` when you run (as root) `apt-file update`.

8.6 Why is "foo-data" not removed when I uninstall "foo"? How do I make sure old unused library-packages get purged?

Some packages are split in program ("foo") and data ("foo-data") (or in "foo" and "foo-doc"). This is true for many games, multimedia applications and dictionaries in Debian and has been introduced since some users might want to access the raw data without installing the program or because the program can be run without the data itself, making "foo-data" optional.

Подобное относится и к библиотекам: обычно они устанавливаются, так как пакеты приложений зависят от них. Когда пакет приложения вычищается, пакет библиотеки может остаться в системе. Или когда пакет приложения больше не зависит, скажем, от `libdb4.2`, а зависит от `libdb4.3`, то пакет `libdb4.2` может остаться в системе при обновлении пакета приложения.

In these cases, "foo-data" doesn't depend on "foo", so when you remove the "foo" package it will not get automatically removed by most package management tools. The same holds true for the library packages. This is necessary to avoid circular dependencies. However, if you use `apt-get` (see [Раздел 8.1.2](#)) or `aptitude` (see [Раздел 8.1.3](#)) as your package management tool, they will track automatically installed packages and give the possibility to remove them, when no packages making use of them remain in your system.

Глава 9

Поддержание обновлений системы Debian

One of Debian’s goals is to provide a consistent upgrade path and a secure upgrade process. We always do our best to make upgrading to new releases a smooth procedure. In case there’s some important note to add to the upgrade process, the packages will alert the user, and often provide a solution to a possible problem.

You should also read the Release Notes document that describes the details of specific upgrades. It is available on the Debian website at <https://www.debian.org/releases/stable/releasenotes> and is also shipped on the Debian CDs, DVDs and Blu-ray discs.

9.1 Как поддерживать систему Debian в актуальном состоянии?

One could simply visit a Debian archive site, then peruse the directories until one finds the desired file, and then fetch it, and finally install it using `dpkg`. Note that `dpkg` will install upgrade files in place, even on a running system. Sometimes, a revised package will require the installation of a newly revised version of another package, in which case the installation will fail until/unless the other package is installed.

Многие люди сочтут предлагаемый способ отнимающим много времени, т. к. Debian развивается слишком быстро — обычно каждую неделю обновляется где-то дюжина и более пакетов. Это число увеличивается по мере приближения в новому основному выпуску. Чтобы справиться с такой лавиной, многие люди предпочитают использовать более автоматизированный способ. Для этих целей существует несколько различных пакетов.

9.1.1 aptitude

aptitude является рекомендуемым менеджером пакетов для систем Debian GNU/Linux и описывается в Раздел 8.1.3.

До того как вы сможете использовать **aptitude** для выполнения обновления, вам следует отредактировать файл `/etc/apt/sources.list`. Если вы хотите выполнить обновление до последней стабильной версии Debian, вам, вероятно, потребуется использовать нечто подобное в качестве источника пакетов:

```
http://deb.debian.org/debian stable main contrib
```

The mirror <https://deb.debian.org/> is backed by a content-delivery network and requests to it will be directed to the closest instance to you. If you have a faster Debian mirror close to you, you can replace `deb.debian.org` with that one. See the mirror list at <https://www.debian.org/mirror/list> for more information.

Or you can use the redirector service `httpredir.debian.org` which aims to solve the problem of choosing a Debian mirror. It uses the geographic location of the user and other information to choose the best mirror that can serve the files. To take advantage of it use a source like this one:

```
http://httpredir.debian.org/debian stable main contrib
```

More details on this can be found in the `sources.list(5)` manual page.
To update your system from the command line, run

```
aptitude update
```

а затем

```
aptitude full-upgrade
```

После того, как вы ответите на возможные вопросы, ваша система будет обновлена.

Заметьте, что **aptitude** не является рекомендованным инструментом для выполнения обновлений с одного выпуска Debian GNU/Linux на другой. Для выполнения таких обновлений вам следует ознакомиться с **Информацией о выпуске** (<https://www.debian.org/releases/stable/releasenotes>). В этом документе описываются рекомендованные шаги для выполнения обновлений с предыдущих выпусков, а также проблемы, которые вам следует решить до выполнения обновления.

For details, see the manual page `aptitude(8)`, and the file `/usr/share/aptitude/README`.

9.1.2 apt-get и apt-cdrom

Альтернативной **aptitude** является **apt-get**, инструмент командной строки на основе APT (он был описан в Раздел 8.1.2).

apt-get, инструмент командной строки на основе APT для работы с пакетами предоставляет простой и безопасный способ установки и обновления пакетов.

Для использования **apt-get**, отредактируйте файл `/etc/apt/sources.list` так же как и для Раздел 9.1.1.

Затем выполните

```
apt-get update
```

а затем

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

Answer any questions that might come up, and your system will be upgraded. See also the `apt-get(8)` manual page, as well as Раздел 8.1.2.

Если для установки пакетов вы хотите использовать компакт-диски, DVD или диски BD, то можете воспользоваться программой **apt-cdrom**. Подробнее об этом написано в информации о выпуске, в разделе «Добавление оптического носителя в качестве источника APT».

Учтите, что когда вы получаете и устанавливаете пакеты, эти файлы так и остаются лежать в одном из подкаталогов `/var`. Чтобы не истратить всё свободное место раздела, не забывайте удалять лишние файлы с помощью команд `apt-get clean` и `apt-get autoclean`, или перемещать их в другое место (подсказка: воспользуйтесь `apt-move`).

9.2 Нужно ли для обновления пакета переходить в однопользовательский режим?

Нет. Пакеты можно обновлять сразу, даже на работающей системе. Если во время обновления пакета требуется остановка процесса, то на этот случай в Debian есть программа `start-stop-daemon`, которая останавливает, а затем перезапускает работавший ранее процесс.

9.3 Нужно ли хранить на диске все файлы-архивы .deb?

Нет. Если вы скачали файлы на диск, то после установки пакетов вы можете удалить их из системы, например с помощью команды `aptitude clean`.

9.4 How can I keep a log of the packages I added to the system? I'd like to know when upgrades and removals have occurred and on which packages!

Параметром `--log` можно заставить **dpkg** вести журнал изменений и выполняемых действий. Протоколируются и все вызовы **dpkg** (например

```
2005-12-30 18:10:33 install hello 1.3.18 2.1.1-4
```

), и их результаты (например

```
2005-12-30 18:10:35 status installed hello 2.1.1-4
```

) If you'd like to log all your **dpkg** invocations (even those done using frontends like **aptitude**), you could add

```
log /var/log/dpkg.log
```

to your `/etc/dpkg/dpkg.cfg`. Be sure the created logfile gets rotated periodically. If you're using **logrotate**, this can be achieved by creating a file `/etc/logrotate.d/dpkg` with the following lines

```
/var/log/dpkg {
    missingok
    notifempty
}
```

More details on **dpkg** logging can be found in the `dpkg(1)` manual page.

aptitude записывает те действия, которые собирается выполнить по установке, удалению или обновлению пакетов, в файл `/var/log/aptitude`. Учтите, что *результаты* этих действий в этом файле не сохраняются!

Another way to record your actions is to run your package management session within the `script(1)` program.

9.5 Возможно ли автоматическое обновление системы?

Да. Для этого можно использовать **cron-apt**; эта программа обновляет систему через равные промежутки времени с помощью задания `cron`. По умолчанию она только обновляет список пакетов и скачивает новые пакеты без запуска их установки.

Замечание: **НЕ** рекомендуется выполнять автоматическое обновление пакетов в *тестовых* или *нестабильных* системах, так как это может привести к неожиданному поведению и удалению пакетов без уведомления.

9.6 Есть несколько машин, как скачивать для них обновления только один раз?

Если в вашей сети более одной машины Debian, то для поддержания систем Debian в актуальном состоянии полезно использовать **apt-cacher**.

apt-cacher сокращает требования к пропускной способности для зеркал Debian, ограничивая частоту обновлений файлов `Packages`, `Releases` и `Sources`, и получает любой файл лишь однократно, независимо от фактических запросов его у прокси. **apt-cacher** автоматически создаёт HTTP-зеркало Debian на основе запросов, проходящих через прокси.

Естественно, те же самые преимущества можно получить с помощью стандартного кэширующего прокси, если настроить все системы на его использование.

Глава 10

Debian и ядро

10.1 Можно ли установить и собрать ядро без некоторых специфичных для Debian настроек?

Да.

There's only one common catch: the Debian C libraries are built with the most recent *stable* releases of the **kernel** headers. If you happen to need to compile a program with kernel headers newer than the ones from the stable branch, then you should either upgrade the package containing the headers (`linux-libc-dev`), or use the new headers from an unpacked tree of the newer kernel. That is, if the kernel sources are in `/usr/src/linux`, then you should add `-I/usr/src/linux/include/` to your command line when compiling.

10.2 Какие инструменты для самостоятельной сборки ядра есть в Debian?

Users who wish to (or must) build a custom kernel are encouraged to use the Debian package target included with recent versions of the kernel build system. After configuring the kernel, simply run the following command:

```
make deb-pkg
```

The new kernel package will be created in the directory one level above the kernel source tree, and it may be installed using `dpkg -i`.

Пользователи должны отдельно скачивать последнюю версию исходного кода ядра (или нужную им версию ядра) со своего любимого Linux-архива, если у них нет пакета `linux-source-версия` (где *версия* означает версию ядра).

10.3 What special provisions does Debian provide to deal with modules?

A configuration file containing modules to be manually loaded at boot time is kept at `/etc/modules`. However, editing this file is rarely needed.

Other module configuration is kept in the `/etc/modprobe.d/` directory. More information about the format of those files can be found in the `modprobe.conf(5)` manual page.

10.4 Можно ли безопасно удалить пакет со старой версией ядра и как это сделать?

Да. Сценарий `linux-image-mm.prerm` проверяет, не пытаетесь ли вы удалить работающее ядро. Поэтому вы можете удалить ненужные пакеты с образами ядра командой:

```
dpkg --purge linux-image-NNN
```

(*NNN*, конечно же, нужно заменить на правильный номер версии и редакции ядра)

10.5 Where can I get more information about Linux packages for Debian?

Further information is maintained in the [Debian Linux Kernel Handbook](https://kernel-team.pages.debian.net/kernel-handbook/) (<https://kernel-team.pages.debian.net/kernel-handbook/>).

Глава 11

Пользовательская настройка системы Debian GNU/Linux

11.1 Как убедиться, что все программы используют одинаковый формат бумаги?

Установите пакет `libpaper1`, и вас попросят указать формат бумаги по умолчанию для всей системы. Настройка будет сохранена в файле `/etc/papersize`.

Users can override the paper size setting using the `PAPERSIZE` environment variable. For details, see the manual page `papersize(5)`.

11.2 Как обеспечить доступ к файлам устройств без нарушения защиты системы?

Многие файлы устройств в каталоге `/dev` принадлежат предопределённым группам. Например, `/dev/sr0` принадлежит группе `cdrom`.

Если вы хотите предоставить конкретному пользователю доступ к устройству, то включите пользователя в группу, которой принадлежит это устройство, т. е. сделайте:

```
adduser b''pb''b''ob''b''lb''b''ьb''b''эb''b''ob''b''vb''b''ab''b''tb''b''eb''b' ←  
'lb''b''ьb'' b''gb''b''pb''b''yb''b''pb''b''ab''
```

Благодаря этому, вам не нужно изменять права доступа на сами файлы устройств.

Если вы выполняли это из оболочки командной строки или графического окружения этого пользователя, то вам нужно выйти из системы и войти снова, чтобы стать членом этой группы. Чтобы посмотреть, в какие группы вы входите, введите команду `groups`.

Заметим, что при использовании `udev` если вы изменяете права доступа к оборудованию, то для некоторых устройств они могут измениться при запуске системы; если это происходит с каком-то важным для вас оборудованием, то вам следует изменить соответствующие правила в `/etc/udev`.

11.3 Как в Debian правильно сменить консольный шрифт при запуске системы?

Для этой задачи существуют пакет `kbd`, отредактируйте файл `/etc/kbd/config`.

11.4 Как настроить параметры по умолчанию для приложений X11?

В Debian файлы ресурсов программ X11 устанавливаются в каталог `/etc/X11/app-defaults/`. Если вы хотите глобально изменить настройки приложений X, то отредактируйте файлы настроек

из этого каталога. Эти файлы считаются настройчными, так что их содержимое будет сохраняться при обновлениях системы.

11.5 How does a Debian system boot?

Like all Unices, Debian boots up by executing the program `init`. Like most Linux distributions, a default Debian system uses `systemd` as the implementation of `init`. Traditional System-V style `init` and other methods are also supported.¹

To control the order in which services are started, traditional System-V style Unix systems use *runlevels*. These are replaced by *targets* under `systemd`. To display the default target to which `systemd` will bring the system, run the command

```
systemctl get-default
```

During boot-up, `systemd` starts the services or other targets listed in the default target file `/lib/systemd/system/default.target`. The files for these services and targets are installed and the service is *enabled* during Debian package installation. If you specifically wish not to start a service during boot-up, instead of removing the corresponding package, you can run the command

```
systemctl disable service.service
```

using the name of the service file installed in `/lib/systemd/system` (usually based on the name of the package).

The *service file* `/lib/systemd/system/rc-local.service` provides an easy way to run customized scripts in the file `/etc/rc.local` after boot-up, similar to what's offered on Debian systems running System-V style `init`. Beware: this script will fail if it tries to interact with the console such as asking for a user password or trying to clear the screen.

You can check the status of any service by the command

```
service package status
```

. To start or stop a service, run

```
service package start
```

and

```
service package stop
```

. The `service` command works with any `init` system supported on a Debian system, not just with `systemd`. If you however prefer to use the same command on any `systemd`-supported Linux system, for checking the status run

```
systemctl status package.service
```

to get the same information.

For more information on `systemd` for Debian, see <https://wiki.debian.org/systemd>.

11.6 And how about Debian and traditional System V init?

Debian supports booting using traditional System V `init`, via the `sysvinit-core` package. The configuration file for System V `init` (which is `/etc/inittab`) specifies that the first script to be executed should

¹In 2014, Debian changed its default `init` system from System V `init` to `systemd`. Debian 8 "jessie" in April 2015 was the first release to ship with `systemd` as default `init`. Four [decisions](https://www.debian.org/devel/tech-ctte#status) (<https://www.debian.org/devel/tech-ctte#status>) of the Debian Technical Committee were involved: [Bug #727708](https://lists.debian.org/20140211193904.GX24404@rzlab.ucr.edu) (<https://lists.debian.org/20140211193904.GX24404@rzlab.ucr.edu>) 2014-02-11: "The committee decided that the default `init` system for Linux architectures in jessie should be `systemd`." [Bug #746715](https://lists.debian.org/20140801023630.GF12356@teltox.donarmstrong.com) (<https://lists.debian.org/20140801023630.GF12356@teltox.donarmstrong.com>) 2014-08-01: "The technical committee expects maintainers to continue to support the multiple available `init` systems", and merge reasonable contributions. [Bug #746578](https://lists.debian.org/20141116001628.G032192@teltox.donarmstrong.com) (<https://lists.debian.org/20141116001628.G032192@teltox.donarmstrong.com>) 2014-11-15: "The committee decided that `systemd-shim` should be the first listed alternative dependency of `libpam-systemd` instead of `systemd-sysv`." This decision made it easier to keep running a non-`systemd` Debian system. [Bug #762194](https://lists.debian.org/21592.61064.527547.410074@chiark.greenend.org.uk) (<https://lists.debian.org/21592.61064.527547.410074@chiark.greenend.org.uk>) 2017-11-04: "On automatic `init` system switching on upgrade"

be `/etc/init.d/rcS`. This script runs all of the scripts in `/etc/rcS.d/` by forking subprocesses to perform initialization such as to check and to mount file systems, to load modules, to start the network services, to set the clock, and to perform other initialization.

По окончании процесса загрузки `init` выполняет все сценарии запуска в каталоге, соответствующем установленному по умолчанию уровню запуска (этот уровень задаётся записью `id` в `/etc/inittab`). Как и в большинстве Unix-ов, совместимых с System V, в Linux имеется 7 уровней запуска:

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

В системах Debian `id = 2`, что означает, что по умолчанию будет установлен уровень запуска '2' (вход в многопользовательский режим), и будут запущены сценарии из каталога `/etc/rc2.d/`.

Debian использует порядок загрузки в соответствии с зависимостями при помощи `insserv`, в каждом сценарии в `/etc/init.d/` используются заголовки LSB, а также параллельную одновременную загрузку при помощи `startpar`, что ускоряет процесс загрузки системы.

Сценарии в любом каталоге, `/etc/rcN.d/` являются лишь символическими ссылками на сценарии в `/etc/init.d/`. Тем не менее, имена файлов в каждом из каталогов `/etc/rcN.d/` выбираются таким образом, что они обозначают способ запуска сценариев в `/etc/init.d/`. Например, до выхода на любой уровень запуска запускаются все сценарии, имена которых начинаются с 'K'; эти сценарии завершают работу служб. После этого запускаются все сценарии, имена которых начинаются с 'S'; эти сценарии запускают работу служб. Число после букв 'K' или 'S' обозначает порядок запуска сценария. Сценарии, имена которых содержат меньшее число, запускаются раньше.

Для поддержки этого метода все сценарии в каталоге `/etc/init.d/` принимают один из параметров: «start», «stop», «reload», «restart» или «force-reload» и затем выполняют задачу, определяемую этим параметром. Эти сценарии можно использовать даже после окончания процесса загрузки системы для управления различными процессами.

Например, при задании параметра «reload» команда

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

отправляет службе `sendmail` сигнал для выполнения повторного чтения файла настройки.

Заметьте, что `invoke-rc.d` не следует использовать для вызова сценариев `/etc/init.d/`, нужно использовать `service`.

11.7 And are there yet other ways of booting a Debian system?

If you do like System V `init`, but don't like the `/etc/rc?.d/*` links, you could install the `file-rc` package. That will convert the links into one single configuration file `/etc/runlevel.conf` instead.

If you like neither System V nor `systemd`, you might like `openrc` or `runit` or `daemontools`.

11.8 Как система управления пакетами работает с пакетами, содержащими настроечные файлы для других пакетов?

Предположим, что какие-то пользователи захотели настроить новый сервер, установив группу пакетов Debian и собственный пакет, состоящий из настроечных файлов. Вообще, это не очень хорошо, поскольку `dpkg` не будет знать о файлах настроек, если они находятся в отдельном пакете, и может затереть эти файлы при обновлении одного из пакетов исходной «группы».

Вместо этого создайте собственный пакет, который правит настроечные файлы интересующей «группы» пакетов. Впоследствии `dpkg` и другие утилиты системы управления пакетами увидят, что файлы были отредактированы локальным «администратором», и не станут их трогать при обновлении указанных пакетов.

11.9 Как заменить программу, установленную из какого-либо пакета, на свою собственную?

Предположим, что системный администратор или пользователь хотят использовать программу «login-local» вместо «login» из пакета Debian login.

НЕ надо:

- перезаписывать /bin/login программой login-local.

Система управления пакетами не будет знать об этих изменениях и при последующих обновлениях или установках просто заменит изменённый вами /bin/login новым из пакета login (или любого другого пакета, предоставляющего /bin/login).

Вместо этого

- выполните:

```
dpkg-divert --divert /bin/login.debian /bin/login
```

чтобы заставить Debian при последующих обновлениях пакета login записывать файл /bin/login в /bin/login.debian.

- затем выполните:

```
cp login-local /bin/login
```

чтобы перенести вашу локальную программу в нужное место.

Запустите `dpkg-divert --list`, чтобы узнать, какие замены (отклонения) уже имеются в вашей системе.

Details are given in the manual page `dpkg-divert(8)`.

11.10 Как включить самостоятельно собранный пакет в список доступных пакетов, известных системе управления пакетами?

Выполните команду:

```
dpkg-scanpackages b''Kb''b''Ab''b''Tb''_b''Дb''b''Bb''b''Ob''b''Иb''b''Чb''b' ←
'Nb''_b''Фb''b''Ab''b''Йb''b''Лb''b''Ob''b''Bb'' b''Фb''b''Ab''b''Йb''b''Лb'' ←
_OVERRIDE [b''Пb''b''Pb''b''Eb''b''Фb''b''Иb''b''Kb''b''Cb''_b''Пb''b''Уb''b' ←
'Tb''b''Иb'''] > my_Packages
```

где:

- **КАТ_ДВОИЧН_ФАЙЛОВ** — это каталог, где находятся файлы-архивы Debian (обычно имеющие расширение «.deb»).
- **OVERRIDE_FILE** is a file that is edited by the distribution maintainers and is usually stored on a Debian archive at `indices/override.main.gz` for the Debian packages in the "main" distribution. You can ignore this for local packages.
- **ПРЕФИКС_ПУТИ** — это *необязательная* строка, которая может предшествовать именам файлов в `my_Packages`.

После создания файла `my_Packages`, сообщите системе управления пакетами о новых пакетах командой:

```
dpkg --merge-avail my_Packages
```

If you are using APT, you can add the local repository to your `sources.list(5)` file, too.

11.11 Некоторым пользователям нравится `mawk`, другим `gawk`; некоторым `vim`, другим `elvis`; некоторым `trn`, другим `tin`. Как осуществляется поддержка предпочтений в Debian?

Бывают случаи, когда в двух пакетах имеются две разные версии какой-либо программы, в целом обеспечивающих одну и ту же функциональность. Одни пользователи могут предпочитать одну из этих программ другой в силу привычки или более привлекательного интерфейса. У других же пользователей этой же системы может быть иное мнение.

Для того, чтобы системные администраторы (или пользователи) могли выбирать для себя из нескольких пакетов с одинаковой функциональностью один предпочтительный, соблюдая в то же время зависимости без необходимости указания конкретного пакета, в Debian используется система «виртуальных» пакетов.

Например, в системе может существовать две различные версии программ чтения новостей. Пакет сервера новостей может 'рекомендовать', чтобы в системе была установлена *некая* программа для чтения новостей, но какая именно — `tin` или `trn`, предлагается выбрать пользователю. Решается это созданием виртуального пакета `news-reader`, который предоставляется пакетами `tin` и `trn`. *Какая* именно из этих программ будет вызываться, определяется символьной ссылкой файла виртуального пакета `/etc/alternatives/news-reader` на выбранный файл, например, на `/usr/bin/trn`.

Простого создания символьной ссылки для полного решения задачи использования альтернативной программы недостаточно; обычно необходимо также заменить страницы руководства и, возможно, вспомогательные программы. Для обеспечения того, чтобы все эти файлы, относящиеся к указанному пакету, выбирались в системе по умолчанию, используется сценарий на Perl `update-alternatives`.

Например, чтобы проверить, какие программы предоставляют «`x-window-manager`», выполните:

```
update-alternatives --display x-window-manager
```

Если вы хотите изменить программу:

```
update-alternatives --config x-window-manager
```

И следуйте инструкциям на экране (просто укажите номер желаемой программы из списка).

Если пакет почему-то не зарегистрировал себя как оконный менеджер (если это ошибка, сообщите о ней), или если вы используете менеджер из каталога `/usr/local`, то на экране вы не увидите предпочитаемой вами программы. Ссылку можно обновить с помощью указания дополнительных параметров для команды, например:

```
update-alternatives --install /usr/bin/x-window-manager \
x-window-manager /usr/local/bin/wmaker-cvs 50
```

Первый аргумент параметра «`--install`» представляет собой символьную ссылку на `/etc/alternatives/NAME`, где `NAME` задаётся вторым аргументом. Третий аргумент указывает программу, на которую должна указывать ссылка `/etc/alternatives/NAME`, а четвёртый задаёт приоритет (чем больше значение, тем больше вероятность автоматического выбора данной альтернативы).

Чтобы удалить добавленный вами вариант, просто выполните:

```
update-alternatives --remove x-window-manager /usr/local/bin/wmaker-cvs
```


Глава 12

Поддержка Debian GNU/Linux

12.1 Какая ещё документация существует по системе Debian?

- Инструкции по установке текущего выпуска: <https://www.debian.org/releases/stable/installmanual>.
- The Debian GNU/Linux reference covers many aspects of system administration through shell-command examples. Basic tutorials, tips, and other information are provided for many different topics ranging from system administration to programming.
Он распространяется в пакете `debian-reference` и доступен по адресу <https://www.debian.org/doc/user-manuals#quick-reference>.
- Руководство по политике Debian описывает требования, предъявляемые к дистрибутиву, т. е. структуру и содержимое архива Debian, некоторые вопросы дизайна операционной системы. Также в нём описаны технические требования, которым должен удовлетворять каждый пакет, чтобы он мог быть включён в дистрибутив, и краткое техническое описание двоичных пакетов и пакетов исходных кодов Debian.
Оно распространяется в пакете `debian-policy` и доступно по адресу <https://www.debian.org/doc/devel-manuals#policy>.
- Documentation developed by the Debian Documentation Project. It is available at <https://www.debian.org/doc> and includes user guides, administration guides and security guides for the Debian GNU/Linux operating system.
- Документация установленных пакетов Debian: большинство пакетов содержит файлы, которые распаковываются в каталог `/usr/share/doc/ПАКЕТ`.
- Документация проекта Linux: из Debian-пакета `doc-linux` устанавливаются все самые новые версии HOWTO и mini-HOWTO [проекта документации для Linux](http://www.tldp.org/) (<http://www.tldp.org/>).
- Страницы «man» в стиле Unix: большинство команд имеют справочные страницы, написанные в духе самых первых man-файлов Unix. Например, чтобы посмотреть справочную страницу команды `ls`, запустите `man ls`. Запуск `man man` выдаст подробную информацию по поиску и просмотру справочных страниц.
Для новых пользователей Debian заметим, что man-страницы многих основных системных команд недоступны, пока не установлены следующие пакеты:
 - `man-db`, который содержит саму программу `man`, и другие программы работы с справочными страницами.
 - `manpages`, который содержит системные справочные страницы. (См. Раздел 5.9).
- Страницы «info» в стиле GNU: пользовательская документация многих команд, особенно утилит GNU, доступна не в man-страницах, а в info-файлах, которые можно читать с помощью GNU программы `info`, набрав `M-x info` в GNU Emacs, или с помощью другой программы просмотра страниц Info.

Its main advantage over the original "man" pages is that it is a hypertext system. It does *not* require the WWW, however; `info` can be run from a plain text console. It was designed by Richard Stallman and preceded the WWW.

Note that you may access a lot of documentation on your system by using a WWW browser, through `dwww`, `dhelp` or `doccentral` commands, found in respective packages, or by using `yelp`.

12.2 Есть ли какие-нибудь онлайн-ресурсы для обсуждения Debian?

Да. Фактически основным способом поддержки пользователей в Debian является электронная почта. Мы расскажем об этом и приведём список других полезных ресурсов. Дополнительные ресурсы представлены на странице [поддержки Debian](https://www.debian.org/support) (<https://www.debian.org/support>).

12.2.1 Списки рассылки

Имеется большое количество [связанных с Debian списков рассылки](https://www.debian.org/MailingLists/) (<https://www.debian.org/MailingLists/>).

В системе с установленным пакетом `doc-debian` все имена списков рассылки можно найти в файле `/usr/share/doc/debian/mailling-lists.txt`.

Списки рассылки Debian именуются следующим образом: `debian-тема-рассылки`. Например: `debian-announce`, `debian-user`, `debian-news`. Чтобы подписаться на какой-нибудь список `debian-тема-рассылки`, отправьте письмо на адрес `debian-тема-рассылки-request@lists.debian.org` со словом «subscribe» в теме письма. Проверьте, что добавили `-request` к адресу электронной почты при подписке или отписке. Иначе ваше письмо попадёт в сам список, что сбивает с толку или раздражает, в зависимости от вашей точки зрения.

Вы можете подписаться на список рассылки, используя [веб-форму](https://www.debian.org/MailingLists/subscribe) (<https://www.debian.org/MailingLists/subscribe>). Также вы можете отписаться от списка рассылки, используя [веб-форму](https://www.debian.org/MailingLists/unsubscribe) (<https://www.debian.org/MailingLists/unsubscribe>).

Если у вас возникли проблемы, вот адрес электронной почты менеджера списков рассылки: listmaster@lists.debian.org.

Списки рассылки являются публичными форумами. Все сообщения, отправленные в списки рассылки, копируются в публичный архив, любой человек (даже если он не является подписчиком) может просмотреть этот архив или выполнить по нему поиск. Никогда не отправляйте какие-либо конфиденциальные или нелицензионные материалы в списки рассылки. В число таких материалов входят адреса электронной почты. Известно, что спамеры злоупотребляют адресами электронной почты, опубликованными в наших списках рассылки. Дополнительную информацию см. в [Политике приватности для списков рассылки](https://www.debian.org/MailingLists/#disclaimer) (<https://www.debian.org/MailingLists/#disclaimer>).

Archives of the Debian mailing lists are available via WWW at <https://lists.debian.org/> which can be searched at [Debian Mailing Lists Archives search](https://lists.debian.org/search.html) (<https://lists.debian.org/search.html>).

12.2.1.1 Как вести себя в списках рассылки?

При использовании списков рассылки Debian, пожалуйста, соблюдайте следующие правила:

- Не посылайте спам. Прочтите [об отношении к рекламе в списках рассылки Debian](https://www.debian.org/MailingLists/#ads) (<https://www.debian.org/MailingLists/#ads>).
- Не болтайте попусту. Люди добровольно работают над Debian, тратя своё время, энергию и деньги.
- Избегайте непристойностей; помимо всего прочего, некоторые люди получают списки рассылки с помощью пакетной радиосвязи, где ругательства запрещены.
- Make sure that you are using the proper list. *Never* post your (un)subscription requests to the mailing list itself.¹
- См. Раздел [12.5](#) с замечаниями по отправке сообщений об ошибках.

¹Use the `debian-list-subject-REQUEST@lists.debian.org` address for that.

12.2.2 Web forum

Debian User Forums (<http://forums.debian.net/>) provides web forums on which you can submit questions about Debian and have them answered by other users. (It is not an official part of the Debian project.)

12.2.3 Вики

Решения общих проблем, практически руководства, советы и другую документацию можно найти в постоянно обновляемой **вики Debian** (<https://wiki.debian.org/>).

12.2.4 Сопровождающие

Пользователи могут задавать вопросы по определённым пакетам сопровождающим по электронной почте. Например, чтобы написать сопровождающему пакета `xuz`, отправьте письмо на адрес `xuz@packages.debian.org`.

12.2.5 Группы новостей Usenet

Пользователи могут отправлять вопросы, не относящиеся к Debian, в одну из групп Linux USENET `comp.os.linux.*` или `linux.*`. Через веб можно найти несколько списков новостных групп Linux Usenet и другие ресурсы, например, на сайте **Linux Online** (<https://www.linux.org/docs/usenet.html>) и **LinuxJournal** (<http://www.linuxjournal.com/helpdesk.php>).

12.3 Как быстро найти информацию по Debian GNU/Linux?

Существуют различные поисковые системы, с помощью которых можно получить информацию по Debian:

- **Поиск по Debian WWW** (<https://search.debian.org/>).
- **Debian Mailing Lists Archives search** (<https://lists.debian.org/search.html>).
For example, to find out what experiences people have had with finding drivers for NVIDIA graphic cards under Debian, try searching the phrase `NVIDIA Debian driver`. This will show you all the posts that contain these strings, i.e. those where people discussed these topics.
- Any of the common web spidering engines, such as **DuckDuckGo** (<https://duckduckgo.com/>) or **Google** (<https://www.google.com/>), as long as you use the right search terms.
For example, searching on the string `"evince"` gives a more detailed explanation of this package than the brief description field in its control file.

12.4 Существуют ли журналы известных ошибок?

Отчёты о неразрешённых (и закрытых) проблемах общедоступны. Проект Debian дал обещание, что "мы не будем скрывать проблемы" в **Общественном договоре Debian** (https://www.debian.org/social_contract).

Дистрибутив Debian GNU/Linux имеет систему отслеживания ошибок (BTS), в файлах которой подробно описываются ошибки, присланные пользователями и разработчиками. Каждой ошибке присваивается номер, и она хранится в файле, пока не будет помечена как исправленная.

Копия информации об ошибках доступна по адресу: <https://www.debian.org/Bugs/>.

Почтовый сервер предоставляет доступ к базе данных системы отслеживания ошибок через электронную почту. Для получения инструкций пошлите письмо по адресу `request@bugs.debian.org` со словом `<help>` в теле сообщения.

12.5 Как сообщить об ошибке в Debian?

Если вы обнаружили ошибку в Debian, прочтите, пожалуйста, инструкции по отправке ошибок в Debian. Эти инструкции можно найти в следующих местах:

- Через WWW. Копию инструкций можно посмотреть здесь <https://www.debian.org/Bugs/Reporting>.
- В любой системе Debian с установленным пакетом `doc-debian`. Инструкции находятся в файле `/usr/share/doc/debian/bug-reporting.txt`.

Вы можете воспользоваться пакетом `reportbug`, который проведёт вас через процесс заполнения сообщения и отправит почту на соответствующий адрес с дополнительными подробностями о вашей системе, которые будут добавлены автоматически. Он также покажет список уже имеющихся ошибок в пакете, и если о вашей ошибке уже кто-то сообщил ранее, вы можете отправить дополнительную информацию по данной проблеме.

Дождитесь автоматического подтверждения о получении вашего сообщения об ошибке. Также, автоматически ей будет присвоен контрольный номер, сделана запись в журнале об ошибках и письмо будет переслано в список рассылки `debian-bugs-dist`.

Глава 13

Помощь проекту Debian

Donations (<https://www.debian.org/donations>) of time (to develop new packages, maintain existing packages, or provide user support), resources (to mirror the package and WWW archives), and money (to pay for new testbeds as well as hardware for the archives) can help the project. See also **How can you help Debian?** (<https://www.debian.org/intro/help>).

13.1 How can I become a Debian member/Debian developer?

The development of Debian is open to all, and new users with the right skills and/or the willingness to learn are needed to maintain existing packages which have been "orphaned" by their previous maintainers, to develop new packages, to write documentation, to do translation work, to help with the Debian website, to provide user support, etc.

Volunteers can participate in the project as Debian contributor, Debian maintainer, or as a Debian developer (with or without uploading privileges):

- Debian contributors participate in many tasks, such as: writing documentation, maintaining the website, translating, supporting users, reviewing bug reports, or participating in different teams. It is an unofficial role and it is usually the starting point for many people.
- Debian maintainers maintain one or more specific packages and can upload these packages directly to the archive without requiring another Debian developer (or maintainer) to sponsor them.
- Debian developers is the full membership role in Debian. A developer can participate in Debian elections, can log into most of the systems Debian manages, and can upload any package to the archive. This role is provided after a strict and thorough process.

The website provides more information on **how to join the project** (<https://www.debian.org/devel/join/>). The description of becoming a Debian maintainer can be found at the **New Member's Corner** (<https://www.debian.org/devel/join/newmaint>) at the Debian web site.

Debian uses Salsa (salsa.debian.org (<https://salsa.debian.org/>)), a GitLab instance, for collaborative development where anyone can sign up and send Merge Requests.

13.2 Как помочь проекту Debian своими ресурсами?

Поскольку Проект стремится к тому, чтобы сделать значительную часть ПО легко доступной по всему миру, нам нужны зеркала. Желательно, но не абсолютно необходимо зеркалировать весь архив. Посетите страницу **Размер зеркала Debian** (<https://www.debian.org/mirror/size>) для получения информации о требованиях к дисковому пространству.

Большинство действий по зеркалированию полностью автоматизированы и выполняются без какого-либо вмешательства человека. Однако при случайных сбоях или изменениях в системе может потребоваться вмешательство человека.

Если у вас есть высокоскоростной доступ к Интернет, ресурсы для зеркала всего дистрибутива или его части, и вы можете выделить своё время, или найти кого-то, кто может регулярно сопровождать систему, свяжитесь, пожалуйста, с нами по адресу debian-admin@lists.debian.org.

13.3 Как оказать финансовую помощь проекту Debian?

Пожертвования спонсоров позволяют Debian иметь машины, а также другое оборудование, организовывать конференции и спринты разработки, а также многое другое. Для получения дополнительной информации обратитесь к странице [Пожертвования Debian](https://www.debian.org/donations) (<https://www.debian.org/donations>). Кроме того, на этой странице приведены различные методы того, как можно осуществить пожертвование.

Можно делать индивидуальные пожертвования организациям, которые очень важны для разработки Проекта Debian. Основной организацией является Software in the Public Interest, зарегистрированная в США, но есть и другие.

13.3.1 Software in the Public Interest

Software in the Public Interest (SPI) является некоммерческой в соответствии с IRS 501(c)(3) организацией, расположенной в США. Целью организации является разработка и распространение свободного ПО.

Она мотивирует программистов использовать Стандартную общественную лицензию GNU и другие лицензии, которые позволяют свободное распространение и использование ПО, а также мотивирует разработчиков оборудования распространять документацию, которая позволит написать драйверы устройств для создаваемого ими оборудования.

SPI выступает фискальным спонсором множества свободных проектов и проектов с открытым исходным кодом. Проект Debian является ассоциированным проектом с момента создания этой организации.

С SPI можно войти в контакт через: <https://www.spi-inc.org/>.

13.3.2 Другие организации

There are a number of organizations created in different countries that hold assets in trust for Debian. The [donations page](https://www.debian.org/donations) (<https://www.debian.org/donations>) lists the trusted organizations individuals can donate to. At the time of this writing there are two of them: the [Debian France Association](https://france.debian.net/) (<https://france.debian.net/>) (in France), and debian.ch (<https://debian.ch/>) (Switzerland and the Principality of Liechtenstein). Additional affiliate organizations in other countries are listed in [Organizations](https://wiki.debian.org/Teams/Auditor/Organizations) (<https://wiki.debian.org/Teams/Auditor/Organizations>) page in the Debian Wiki.

Глава 14

Распространение Debian GNU/Linux в коммерческом продукте

14.1 Можно ли штамповать CD-диски Debian и продавать их?

Запросто! Вам не нужно разрешение на распространение всего, что мы *выпустили*, так что можете штамповать свои CD-диски, как только окончится этап бета-тестирования. Вы не должны ничего нам платить. Конечно, все производители CD-дисков должны соблюдать условия лицензий программ Debian. Многие программы, например, лицензированы под GPL, которая требует от вас распространять их исходный код.

Также мы публикуем список производителей CD-дисков, жертвующих свои деньги, ПО и время для проекта Debian, и мы будем советовать пользователям приобретать продукцию этих производителей в качестве компенсации этих пожертвований.

14.2 Можно ли укомплектовать Debian несвободным ПО?

Да. В то время как все основные компоненты Debian являются свободным ПО, для тех программ, которые не распространяются свободно, мы предусмотрели каталог non-free.

Производители CD-дисков *могут* распространять программы, которые помещены в этот каталог, в зависимости от положений их лицензии или собственных пожеланий авторов этих пакетов программ. Кроме того, производители CD-дисков могут помещать на эти же CD несвободные программы, которые они получают из других источников. В этом нет ничего нового: многие производители распространяют свободные и несвободные программы на одном CD-диске. Мы же, конечно, стремимся к тому, чтобы авторы программ выпускали свои продукты как свободное ПО.

14.3 Я делаю специальный дистрибутив Linux для «вертикального маркетинга». Могу я использовать Debian GNU/Linux как основу этой системы и добавлять мои собственные приложения поверх него?

Yes. Debian-derived distributions are being created both in close cooperation with the Debian project itself and by external parties. One can use the [Debian Pure Blends](https://www.debian.org/blends/) (<https://www.debian.org/blends/>) framework to work together with Debian; [DebianEdu/Skolelinux](https://wiki.debian.org/DebianEdu/) (<https://wiki.debian.org/DebianEdu/>) is one such project.

There are several other Debian-derived distributions already on the market, such as grml, LMDE (Linux Mint Debian Edition), Knoppix and Ubuntu, that are targeted at a different kind of audience than the original Debian GNU/Linux is, but use most of our components in their product.

Debian также предлагает механизм, который позволяет разработчикам и системным администраторам устанавливать локальные версии определённых файлов таким образом, что они не будут перезаписываться при обновлении пакетов. См. Раздел [11.9](#).

14.4 Можно ли создать «пакет» Debian из коммерческой программы, чтобы её можно было без проблем установить в любой системе Debian?

Да, это возможно. Средство управления пакетами — свободное программное обеспечение; пакеты могут содержать как свободное, так и несвободное ПО, и любой из них легко установить.

Глава 15

Изменения, ожидаемые в следующем выпуске Debian

Для каждого нового выпуска Проект Debian делает попытку сконцентрироваться на нескольких задачах. Они называются "Задачами выпуска", эти задачи описаны на странице <https://wiki.debian.org/ReleaseGoals/>. Заметьте, что следующие разделы могут быть не совсем актуальны, для получения дополнительной и более свежей информации о статусе этих задач обращайтесь к вики.

15.1 Улучшение безопасности системы

Целью Проекта Debian является обеспечение гарантии, что любая установленная система является отказоустойчивой и защищённой от атак. Для достижения этой цели имеется несколько путей:

- Улучшение безопасности программ путём их компиляции со **специальными сборочными флагами безопасности** (<https://wiki.debian.org/ReleaseGoals/SecurityHardeningBuildFlags>), что позволяет использовать различные методы защиты от известных проблем безопасности,
- Улучшение системных настроек по умолчанию, чтобы они были менее подвержены атакам (и локальным, и удалённым),
- Включать опции по обеспечению безопасности, содержащиеся в новых версиях ядра.

All of these are done in an ongoing basis. For the first item, a set of security hardening build flags that try to prevent known attacks such as stack smashing, predictable locations of values in memory, etc. is used. The target is to cover at least all packages that are part of the basic installation as well as packages that had to be updated through a Security Advisory since 2006. As of this writing, around 400 packages have been modified since this effort was first started. All the issues are **tracked in the BTS** (<https://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=goal-hardening;users=hardening-discuss@lists.alioth.debian.org>).

15.2 Расширенная поддержка не-англоговорящих пользователей

В Debian уже имеется очень хорошая поддержка не-англоговорящих пользователей, см. Раздел 5.9.

Мы надеемся найти людей, которые помогут с обеспечением поддержки большего количества языков и переведут программы и документацию. Многие программы и связанная с Debian документация уже поддерживают интернационализацию, поэтому нам требуются переводчики. Тем не менее, некоторые программы всё ещё нужно подготовить (интернационализировать) для такой работы.

Переводческий проект GNU <ftp://ftp.gnu.org/pub/gnu/ABOUT-NLS> работает над интернационализацией программ GNU и различных проектов, у таких окружений рабочего стола как GNOME и KDE имеются свои команды переводчиков. Цель Проекта Debian не заключается в замене или повторении работы, проделанной этими проектами, в действительности Debian только

выигрывает от работы, проделанной переводчиками в рамках этих проектов. Тем не менее, всё ещё существует множество программ, которые не затрагиваются этими проектами, и работа над переводами которых осуществляется в рамках Debian.

Предыдущие выпуски Debian концентрировались на таких темах как:

- Поддержка интернационализации во всех пакетах, использующих debconf: пакеты, использующие управление настройками Debian, должны позволять перевод всех видимых пользователем сообщений во время настройки пакета.
- Поддержка интернационализации описаний пакетов: обновить программы управления пакетами так, чтобы они использовали перевод описаний пакетов.
- Использовать кодировку UTF-8 для файлов debian/changelog и debian/control. Это позволит, например, правильно отображать в файлах журналов имена людей из азиатских стран.
- Поддержка интернационализации в программе установки Debian, включая полную поддержку некоторых языков, для которых требуется использовать графического интерфейса.

15.3 Улучшения в программе установки Debian

Большая работа была проделана над программой установки Debian, что привело к значительным улучшениям. Хотелось бы упомянуть два из них.

Запуск программы установки из Microsoft Windows: теперь возможно запустить программу установки прямо из Microsoft Windows без необходимости изменения настроек BIOS. Достаточно вставить CD-ROM, DVD-ROM или USB-накопитель, и запустится программа автозапуска, предлагающая пошаговый процесс запуска программы установки Debian.

15.4 Ещё больше архитектур

Complete Debian system on other architectures. Notice that even though some architectures are dropped for a given release, there still might be a way to install and upgrade using the latest sid.

15.5 Больше ядер

В дополнение к Debian GNU/Hurd, Debian также переносится на ядра BSD, а именно FreeBSD (<https://www.debian.org/ports/kfreebsd-gnu/>). Этот перенос работает на AMD64 («kfreebsd-amd64») и обычном Intel («kfreebsd-i386»).

Глава 16

Общая информация о ЧаВо

16.1 Авторы

Первая редакция этих ЧаВо (FAQ) была написана и сопровождалась Дж.Х.М. Дассеном (J.H.M. Dassen (Ray)) и Чаком Стикельманом (Chuck Stickelman). Сюзан Г. Клейман (Susan G. Kleinmann) и Свен Рудольф (Sven Rudolph) переписали Debian GNU/Linux FAQ. Затем ЧаВо стал сопровождать Сантьяго Вила (Santiago Vila), а позже Иосип Родин (Josip Rodin). Текущим сопровождающим является Хавьер Фернандес-Сангвино (Javier Fernandez-Sanguino).

Откуда бралась информация:

- Анонс о выпуске Debian-1.1, автор Брюс Перенс (<https://perens.com/>),
- ЧаВо по Linux, написанные Яном Джексоном (<https://www.chiark.greenend.org.uk/~ijackson/>),
- Архивы списков рассылки Debian (<https://lists.debian.org/>),
- Руководство программиста dpkg и Руководство по политике Debian (см. Раздел 12.1),
- от многих разработчиков, добровольцев и бета-тестеров и
- чудных воспоминаний его авторов. :-)
- Камараджу Кусуманчи, ЧаВо по выбору выпуска Debian (http://people.cornell.edu/pages/kk288/debian_choosing_distribution.html), который любезно выпустил этот документ под лицензией GPL, чтобы я смог включить его в качестве новой главы в данное руководство (см. Глава 3).

Авторы хотели бы поблагодарить всех, кто сделал возможным выход этого документа в свет. Никаких гарантий. Все торговые марки являются собственностью их владельцев.

16.2 Обратная связь

Всегда приветствуются комментарии и дополнения к данному документу. Отправляйте сообщения электронной почты на адрес doc-debian@packages.debian.org, либо шлите отчёты об ошибках уровня wishlist в пакете `debian-faq` (<https://bugs.debian.org/debian-faq>).

16.3 Где взять

Последнюю версию этого документа можно посмотреть на WWW странице Debian по адресу <https://www.debian.org/doc/FAQ/>.

It is also available for download in plain text, HTML, and PDF formats at <https://www.debian.org/doc/user-manuals#faq>. Also, there are several translations there.

Этот документ доступен в пакете `debian-faq`. Переводы доступны в пакетах `debian-faq-ru`, `debian-faq-fr` и других.

The original XML files used to create this document are also available in `debian-faq`'s source package, or on Salsa (salsa.debian.org), Debian's GitLab instance for collaborative development, at: <https://salsa.debian.org/ddp-team/debian-faq.git>. You can browse the repository at <https://salsa.debian.org/ddp-team/debian-faq>.

16.4 Формат документа

This document was written using the DocBook XML DTD. This 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, or GNU info.