

# **Руководство по установке Debian GNU/Linux**

3 ноября 2017 г.

---

## Руководство по установке Debian GNU/Linux

Copyright © 2004 – 2017 Команда разработчиков программы установки Debian

Это руководство является свободным программным обеспечением; вы можете его распространять и/или модифицировать на условиях Стандартной Общественной Лицензии GNU. С текстом соглашения можно ознакомиться здесь: Приложение **F**.

# Оглавление

<b>1</b>	<b>Добро пожаловать в Debian</b>	<b>1</b>
1.1	Что такое Debian?	1
1.2	Что такое GNU/Linux?	2
1.3	Что такое Debian GNU/Linux?	2
1.4	Что такое Debian GNU/kFreeBSD?	3
1.5	Что такое Debian GNU/Hurd?	3
1.6	Получение Debian	3
1.7	Получение последней версии этого документа	4
1.8	Структура этого документа	4
1.9	Помогите нам с документацией!	4
1.10	Об авторских правах и лицензиях на программы	4
<b>2</b>	<b>Системные требования</b>	<b>6</b>
2.1	Поддерживаемое оборудование	6
2.1.1	Поддерживаемые архитектуры	6
2.1.2	Three different ARM ports	7
2.1.3	Variations in ARM CPU designs and support complexity	7
2.1.4	Платформы, поддерживаемые Debian/armhf	7
2.1.5	Платформы, больше не поддерживаемые Debian/armhf	9
2.1.6	Несколько процессоров	9
2.1.7	Graphics Hardware Support	9
2.1.8	Аппаратура для подключения к сети	9
2.1.9	Периферия и другое оборудование	9
2.2	Устройства, которым требуются микропрограммы	9
2.3	Покупка аппаратного обеспечения специально для GNU/Linux	10
2.3.1	Избегайте собственного или закрытого аппаратного обеспечения	10
2.4	Установочные носители	11
2.4.1	CD-ROM/DVD-ROM/BD-ROM	11
2.4.2	Сеть	11
2.4.3	Жёсткий диск	11
2.4.4	Система Un*x или GNU	11
2.4.5	Поддерживаемые системы хранения	11
2.5	Требования к памяти и дисковому пространству	11
<b>3</b>	<b>Перед установкой Debian GNU/Linux</b>	<b>13</b>
3.1	Обзор процесса установки	13
3.2	Сохраните ваши данные в надёжном месте!	14
3.3	Информация, которая понадобится	14
3.3.1	Документация	14
3.3.1.1	Руководство по установке	14
3.3.1.2	Техническая документация	14
3.3.2	Поиск источников технической информации	14
3.3.3	Совместимость аппаратного обеспечения	15
3.3.3.1	Тестирование оборудования на совместимость с помощью Live-системы	16
3.3.4	Настройки сети	16
3.4	Минимальные требования к аппаратному обеспечению	17
3.5	Переразметка для мульти-загрузочных систем	17
3.6	Настройка аппаратного обеспечения и операционной системы перед установкой	18
3.6.1	Микропрограмма ARM	18
3.6.2	Debian-provided U-Boot (system firmware) images	18
3.6.3	Задание MAC-адреса ethernet в U-Boot	18
3.6.4	Проблемы размещения ядра/Initrd/дерева-устройств в U-Boot	19

<b>4</b>	<b>Получение носителя с дистрибутивом</b>	<b>20</b>
4.1	Официальный набор CD/DVD-ROM Debian GNU/Linux	20
4.2	Скачивание файлов с серверов-зеркал Debian	20
4.2.1	Где искать установочные образы	20
4.2.1.1	Мультиплатформенные файлы установки для armhf	20
4.3	Подготовка файлов для загрузки по TFTP	21
4.3.1	Настройка RARP сервера	21
4.3.2	Настройка DHCP сервера	21
4.3.3	Настройка BOOTP сервера	22
4.3.4	Включение TFTP сервера	22
4.3.5	Копирование TFTP образов в каталог TFTP сервера	22
4.4	Автоматическая установка	23
4.4.1	Автоматическая установка с помощью программы установки Debian	23
<b>5</b>	<b>Загрузка системы установки</b>	<b>24</b>
5.1	Запуск программы установки на 32-bit hard-float ARMv7	24
5.1.1	Форматы загрузочных образов	24
5.1.2	Console configuration	24
5.1.3	Загрузка по TFTP	24
5.1.3.1	Загрузка по TFTP из U-Boot	24
5.1.3.2	Pre-built netboot tarball	25
5.1.4	Загрузка с карты памяти USB из U-Boot	26
5.1.5	Using pre-built SD-card images with the installer	26
5.2	Лёгкость управления	26
5.2.1	Пользовательский интерфейс программы установки	27
5.2.2	Встраиваемые устройства	27
5.2.3	Тема с высокой контрастностью	27
5.2.4	Zoom	27
5.2.5	Expert install, rescue mode, automated install	27
5.2.6	Специальные возможности установленной системы	27
5.3	Параметры загрузки	27
5.3.1	Загрузочная консоль	28
5.3.2	Параметры программы установки Debian	28
5.3.3	Использование параметров загрузки для ответов на вопросы	30
5.3.4	Передача параметров модулям ядра	31
5.3.5	Внесение модулей ядра в чёрный список	31
5.4	Диагностика проблем процесса установки	31
5.4.1	Надёжность CD-ROM	31
5.4.1.1	Общие причины	32
5.4.1.2	Как изучать и может быть решить проблему	32
5.4.2	Настройка загрузки	33
5.4.3	Интерпретация загрузочных сообщений ядра	33
5.4.4	Сообщение о проблемах при установке	33
5.4.5	Отправка отчётов об ошибках установки	33
<b>6</b>	<b>Использование программы установки Debian</b>	<b>35</b>
6.1	Как работает программа установки	35
6.1.1	Использование программы установки с графическим интерфейсом	36
6.2	Описание компонент	36
6.3	Применение отдельных компонент	37
6.3.1	Настройка программы установки Debian и оборудования	38
6.3.1.1	Определение доступной памяти / режим с малым потреблением памяти	38
6.3.1.2	Выбор параметров локализации	38
6.3.1.3	Выбор клавиатуры	39
6.3.1.4	Поиск образа ISO программы установки Debian	39
6.3.1.5	Настройка сети	40
6.3.1.5.1	Автоматическая настройка сети	40
6.3.1.5.2	Настройка сети вручную	40
6.3.1.5.3	IPv4 и IPv6	40

6.3.1.6	Настройка времени и часового пояса	40
6.3.2	Добавление пользователей и паролей	41
6.3.2.1	Установка пароля суперпользователя (root)	41
6.3.2.2	Создание учётной записи обычного пользователя	41
6.3.3	Разметка разделов и выбор точек монтирования	41
6.3.3.1	Поддерживаемые параметры разметки	42
6.3.3.2	Направляющая разметка	42
6.3.3.3	Разметка вручную	44
6.3.3.4	Настройка устройств Multidisk (программный RAID)	44
6.3.3.5	Настройка менеджера логических томов (LVM)	47
6.3.3.6	Настройка зашифрованных томов	48
6.3.4	Установка базовой системы	50
6.3.5	Установка дополнительного программного обеспечения	50
6.3.5.1	Настройка apt	50
6.3.5.1.1	Установка с нескольких CD или DVD дисков	51
6.3.5.1.2	Использование сетевого сервера-зеркала пакетов	51
6.3.5.1.3	Выбор сетевого сервера-зеркала пакетов	52
6.3.5.2	Выбор и установка программного обеспечения	52
6.3.6	Установка и настройка системного загрузчика	53
6.3.6.1	Обнаружение операционных систем	53
6.3.6.2	Настройка системы на загрузку с помощью flash-kernel	54
6.3.6.3	Продолжение без системного загрузчика	54
6.3.7	Завершение установки	54
6.3.7.1	Настройка системного времени	54
6.3.7.2	Перезагрузка системы	54
6.3.8	Устранение проблем	54
6.3.8.1	Сохранение протокола установки	54
6.3.8.2	Использование командной строки и просмотр журнальных файлов	55
6.3.9	Установка по сети	55
6.4	Загрузка отсутствующих микропрограмм	56
6.4.1	Подготовка носителя	57
6.4.2	Микропрограмма в установленной системе	57
<b>7</b>	<b>Загрузка только что установленной системы Debian</b>	<b>59</b>
7.1	Момент истины	59
7.2	Монтирование зашифрованных томов	59
7.2.1	Устранение неполадок	60
7.3	Вход в систему	60
<b>8</b>	<b>Следующие шаги и что делать дальше</b>	<b>61</b>
8.1	Выключение системы	61
8.2	Если вы новичок в Unix	61
8.3	Самостоятельное ориентирование в Debian	61
8.3.1	Система управления пакетами Debian	61
8.3.2	Дополнительное программное обеспечение для Debian	62
8.3.3	Управление версиями приложений	62
8.3.4	Управление заданиями cron	62
8.4	Что почитать	62
8.5	Настройка почты	62
8.5.1	Почтовая конфигурация по умолчанию	63
8.5.2	Отправка почты из системы	63
8.5.3	Настройка агента пересылки почты Exim4	63
8.6	Сборка нового ядра	64
8.6.1	Управление образами ядра	65
8.7	Восстановление неработающей системы	65

<b>A</b>	<b>Краткая инструкция по установке</b>	<b>67</b>
A.1	Предварительные сведения	67
A.2	Загрузка программы установки	67
A.2.1	Загрузка с CDROM	67
A.2.2	Загрузка по сети	67
A.2.3	Загрузка с жёсткого диска	67
A.3	Установка	68
A.4	Отправьте нам отчёт об установке	68
A.5	И напоследок...	69
<b>B</b>	<b>Автоматическая установка с помощью списка ответов</b>	<b>70</b>
B.1	Введение	70
B.1.1	Методы использования файла ответов	70
B.1.2	Ограничения	71
B.2	Использование автоматической установки	71
B.2.1	Загрузка файла ответов	71
B.2.2	Использование параметров загрузки для автоматизации	72
B.2.3	Автоматический режим	73
B.2.4	Полезные в автоматизации псевдонимы	74
B.2.5	Использование сервера DHCP для указания файлов автоматической конфигурации	74
B.3	Создание файла ответов	75
B.4	Содержимое файла ответов (для buster)	76
B.4.1	Локализация	76
B.4.2	Настройка сети	77
B.4.3	Сетевая консоль	78
B.4.4	Настройка сервера-зеркала	78
B.4.5	Настройка учётной записи	79
B.4.6	Настройка времени и часового пояса	79
B.4.7	Разметка дисков	80
B.4.7.1	Пример разметки	80
B.4.7.2	Разметка с использованием RAID	81
B.4.7.3	Управление монтированием разделов	82
B.4.8	Установка базовой системы	83
B.4.9	Настройка Apt	83
B.4.10	Выбор пакетов	84
B.4.11	Завершение установки	84
B.4.12	Автоматизация других пакетов	85
B.5	Дополнительные параметры	85
B.5.1	Запуск пользовательских команд во время установки	85
B.5.2	Использование автоматической установки для изменения значений по умолчанию	85
B.5.3	Последовательная загрузка нескольких файлов ответов	86
<b>C</b>	<b>Разметка дисков в Debian</b>	<b>87</b>
C.1	Дисковые разделы Debian и их размеры	87
C.2	Структура каталогов	87
C.3	Рекомендуемые схемы разметки	88
C.4	Имена устройств в Linux	89
C.5	Программы разметки в Debian	89
<b>D</b>	<b>Разное</b>	<b>91</b>
D.1	Устройства Linux	91
D.1.1	Установка мыши	91
D.2	Дисковое пространство, необходимое для задач	92
D.3	Установка Debian GNU/Linux из системы Unix/Linux	93
D.3.1	Начинаем	93
D.3.2	Установка <b>debootstrap</b>	94
D.3.3	Запуск <b>debootstrap</b>	94
D.3.4	Настройка базовой системы	94
D.3.4.1	Создание файлов устройств	95

D.3.4.2	Монтирование разделов . . . . .	95
D.3.4.3	Настройка часового пояса . . . . .	96
D.3.4.4	Настройка сети . . . . .	96
D.3.4.5	Настройка Apt . . . . .	97
D.3.4.6	Настройка раскладки клавиатуры и локалей . . . . .	97
D.3.5	Установка ядра . . . . .	98
D.3.6	Установка системного загрузчика . . . . .	98
D.3.7	Удалённый доступ: установка SSH и настройка доступа . . . . .	98
D.3.8	Заключительные шаги . . . . .	98
D.4	Установка Debian GNU/Linux с использованием PPP через Ethernet (PPPoE) . . . . .	99
<b>E</b>	<b>Общие сведения</b>	<b>100</b>
E.1	Об этом документе . . . . .	100
E.2	Как помочь в разработке этого документа . . . . .	100
E.3	Люди, внёсшие значительный вклад в написание этого руководства . . . . .	100
E.4	Торговые марки . . . . .	101
<b>F</b>	<b>GNU General Public License</b>	<b>102</b>
F.1	Preamble . . . . .	102
F.2	GNU GENERAL PUBLIC LICENSE . . . . .	102
F.3	How to Apply These Terms to Your New Programs . . . . .	105

# Список таблиц

<b>3</b>	<b>Перед установкой Debian GNU/Linux</b>	
3.1	Информация об оборудовании, полезная при установке . . . . .	15
3.2	Рекомендуемые минимальные требования к системе . . . . .	17



### Аннотация

Данный документ содержит инструкции по установке системы Debian GNU/Linux 10 («buster») для архитектуры 32-bit hard-float ARMv7 («armhf»). Кроме того, в нём указаны источники более подробной информации и описаны методы наиболее эффективного применения системы Debian.

#### Внимание



This translation of the installation guide is not up-to-date and currently there is noone actively working on updating it. Keep this in mind when reading it; it may contain outdated or wrong information. Read or double-check the English variant, if in doubt. If you can help us with updating the translation, please contact [debian-boot@lists.debian.org](mailto:debian-boot@lists.debian.org) or the [debian-I10n-xxx mailinglist](#) for this language. Many thanks

# Установка Debian GNU/Linux 10 на armhf

Мы рады, что вы решили попробовать Debian. Мы уверены, что дистрибутив Debian GNU/Linux покажется вам замечательным. Debian GNU/Linux собирает высококачественное свободное ПО со всего света, объединяя его в единое целое. Мы верим, вы признаете, что получившийся результат — это действительно больше, чем сумма частей.

Мы понимаем, что многие из вас хотят установить Debian, не читая этого руководства, и поэтому программа установки Debian была спроектирована с учётом этого. Если прямо сейчас у вас нет времени на чтение всего руководства, прочтите "Краткое руководство по установке", которое покажет вам основы процесса установки, а также укажет на конкретные главы руководства для углублённого изучения конкретных тем или на случай непредвиденных проблем. Приложение **A** содержит это краткое руководство.

Тем не менее, мы надеемся, что у вас найдётся время на чтение большей части этого руководства, так как прочитав его, вы сможете более умело и успешно установить систему.

# Глава 1

## Добро пожаловать в Debian

Эта глава посвящена обзору проекта Debian и Debian GNU/Linux. Если вы уже знаете историю проекта Debian и дистрибутива Debian GNU/Linux, спокойно переходите к следующей главе.

### 1.1 Что такое Debian?

Debian — это организация, полностью состоящая из добровольцев, посвятившая себя разработке свободного программного обеспечения и распространению идеалов сообщества Свободного Программного Обеспечения (Free Software). Проект Debian начался в 1993 году, когда Иэн Мёрдок (Ian Murdock) выпустил открытое приглашение к разработчикам программ внести свой вклад в полноценный программный дистрибутив, основанный на относительно новом ядре Linux. Небольшая группа энтузиастов, на которых повлияла философия GNU, изначально поддерживаемая [Фондом Свободного Программного Обеспечения](#), с годами выросла в организацию, с количеством *разработчиков Debian* 1062.

Разработчики Debian выполняют различные задачи, включая администрирование [Web](#) и [FTP](#) сайтов, графический дизайн, юридический анализ программных лицензий, написание документации и, конечно же, сопровождение пакетов.

В интересах распространения нашей философии и привлечения разработчиков, которые верят в принципы Debian, проект Debian выпустил множество документов, которые подчёркивают наши ценности и рассказывают, что значит быть разработчиком Debian:

- [Общественный Договор Debian](#) — это утверждение обязательств Debian перед Сообществом Свободного Программного Обеспечения. Любой, кто соглашается следовать [Общественному Договору Debian](#) может стать [сопровождающим](#). Любой сопровождающий может добавить новые программы в Debian — при условии, что программы соответствуют нашим критериям свободы, а пакеты следуют нашим стандартам качества.
- [Критерии Debian по определению Свободного Программного Обеспечения \(DFSG\)](#) — это краткие и ясные определения критериев Debian к свободным программам. DFSG является очень влиятельным документом в Движении Свободного Программного Обеспечения и взят за основу [Определения Открытых Исходных Текстов](#).
- [Руководство по политике Debian](#) — это обширная спецификация стандартов качества проекта Debian.

Разработчики Debian также вовлечены и в другие проекты, некоторые из которых созданы специально для Debian, другие выполняются для части или для всего сообщества Linux. Вот несколько примеров:

- Проект [Linux Standard Base \(LSB, Основа Стандарта Linux\)](#) нацелен на стандартизацию базовой системы GNU/Linux, что позволит сторонним разработчикам программного обеспечения и оборудования легко разрабатывать программы и драйверы устройств для Linux в целом, а не для конкретного дистрибутива GNU/Linux.
- [Стандарт Иерархии Файловой Системы \(FHS\)](#) — это попытка стандартизировать организацию файловой системы Linux. Стандарт FHS позволит разработчикам программ сконцентрировать свои усилия на создании программы и не беспокоиться о том, как пакет будет устанавливаться в различные дистрибутивы GNU/Linux.

- Внутренний проект **Debian для детей**, его цель — уверенность в том, что Debian'у есть что предложить нашим младшим пользователям.

Для более общей информации о Debian, смотрите **Debian FAQ**.

## 1.2 Что такое GNU/Linux?

Linux — это операционная система: набор программ, которые позволяют вам работать с вашим компьютером и запускать другие программы.

Операционная система (ОС) состоит из нескольких основных программ, которые нужны вашему компьютеру, чтобы он мог общаться и получать инструкции от пользователей; читать и записывать данные на жёсткие диски, ленты и принтеры; контролировать использование памяти; запускать другие программы. Самая важная часть ОС — ядро. В системе GNU/Linux ядром является Linux. Оставшаяся часть системы состоит из других программ, многие из которых написаны проектом GNU или для проекта GNU. Поскольку одно ядро Linux — это ещё не вся работающая ОС, мы предпочитаем использовать термин «GNU/Linux» для обозначения систем, которые многие люди ошибочно называют «Linux».

Linux сделана по образцу операционной системы Unix. С самого начала Linux разрабатывалась как многозадачная, многопользовательская система. Этим факторов достаточно, чтобы сделать Linux отличной от других широко известных операционных систем. Однако, Linux отличается намного сильнее, чем вы можете себе представить. В отличие от других операционных систем, Linux никому не принадлежит. Значительная часть системы разработана добровольцами бесплатно.

Разработка того, что позже станет GNU/Linux, началась в 1984 году, когда **Фонд Свободного Программного обеспечения** начал разработку свободной Unix-подобной операционной системы, названной GNU.

**Проект GNU** разработал всеобъемлющий набор свободного программного обеспечения для использования в Unix™ и Unix-подобных операционных системах, таких как Linux. Эти программы позволяют пользователям выполнять задачи широкого спектра, от простых (копирование или удаление файлов из системы) до сложных (написание и компилирование программ или выполнение сложного редактирования в различных форматах документов).

Хотя многие рабочие группы и отдельные люди вложили свой труд в создание Linux, наибольшим пока ещё остаётся вклад Фонда Свободного Программного Обеспечения. Помимо разработки большинства используемых инструментов, фонд также создал философию и организовал сообщество (независимых программистов и энтузиастов-пользователей). Благодаря этому, в свою очередь, стало возможным появление отлаженных Linux систем.

**Ядро Linux** впервые появилось в 1991 году, когда финский студент-компьютерщик по имени Линус Торвальдс (Linus Torvalds) анонсировал первую версию замены ядра Minix в группе новостей `comp.os.minix`. Подробнее смотрите на сайте Linux International **страничку истории Linux**.

Линус Торвальдс продолжает координировать работу нескольких сотен разработчиков с помощью нескольких сопровождающих подсистем. Есть **официальный сайт** для ядра Linux. Информацию о списке рассылки `linux-kernel` можно найти в **FAQ списка рассылки linux-kernel**.

Пользователи Linux имеют огромную свободу в выборе программного обеспечения. Например, пользователи Linux могут выбирать из дюжины различных оболочек командной строки и нескольких графических рабочих столов. Этот выбор часто удивляет пользователей других операционных систем, которые и не думали, что командные оболочки или рабочий стол можно заменить.

Linux также менее подвержен сбоям, лучше приспособлен для одновременного выполнения более чем одной программы и более защищён, чем многие другие системы. Благодаря этим преимуществам, Linux быстро завоевывает рынок серверных операционных систем. С недавних пор, Linux начал становиться популярным среди домашних пользователей и в корпоративной среде.

## 1.3 Что такое Debian GNU/Linux?

Комбинация философии и методологии Debian, средств GNU, ядра Linux и других важных программ, составляют уникальный дистрибутив под названием Debian GNU/Linux. Этот дистрибутив собран из большого количества *пакетов* программ. Каждый пакет в дистрибутиве содержит исполняемые файлы, сценарии, документацию, конфигурационную информацию и имеет *сопровождающего*, который отвечает за поддержание пакета в актуальном состоянии, отслеживает сообщения об ошибках (bug reports) и общается с основными авторами программы. Огромная база наших пользователей в сочетании с нашей системой отслеживания сообщений об ошибках гарантируют, что проблемы будут быстро обнаружены и устранены.

Внимание Debian к деталям позволило создать высококачественный, стабильный и расширяемый дистрибутив. Установленные системы могут быть легко настроены на выполнение самых различных задач: от упрощённого межсетевого экрана (firewall), рабочей станции учёного, до высокопроизводительного сетевого сервера.

Debian особенно популярен среди опытных пользователей из-за его технического совершенства и глубокого понимания нужд и ожиданий сообщества Linux. Debian также добавил много новых свойств в Linux, которые теперь повсеместно используются.

Например, Debian был первым дистрибутивом Linux, который имел систему управления пакетами для более лёгкой установки и удаления программ. Также он был первым дистрибутивом Linux, который можно обновить до последней версии без переустановки.

Debian продолжает быть лидером в разработке Linux. Его процесс разработки является примером того, как хорошо может работать модель Открытых Исходников (Open Source) даже для очень сложных задач, таких как построение и сопровождение целой операционной системы.

Наиболее отличительной особенностью Debian от других дистрибутивов Linux является его система управления пакетами. Она даёт администратору системы Debian полный контроль над пакетами, установленными в системе, будь это установка одного пакета или автоматическое обновление всей операционной системы. Также, отдельные пакеты могут быть защищены от обновления. Вы даже можете указать системе управления пакетами программы, которые вы собрали сами и о том, от каких программ они зависят.

Для защиты вашей системы от «троянских коней» и других вредоносных программ, на серверах Debian выполняется проверка того, что поступающие пакеты получены только от зарегистрированных сопровождающих Debian. Также разработчики Debian заботятся о настройке безопасности программ в своих пакетах. Когда случаются проблемы с безопасностью в выпущенном пакете, обычно очень быстро выходят исправления. С помощью простой системы обновлений Debian исправления безопасности можно загружать и устанавливать автоматически через Интернет.

Основным и самым лучшим способом получения поддержки системы Debian GNU/Linux и общения с разработчиками Debian является множество списков рассылки проекта Debian (на момент написания этих строк их было более 283). Самый простой способ подписаться на один из этих списков — посетить [страницу подписки на списки рассылки Debian](#) и заполнить форму.

## 1.4 Что такое Debian GNU/kFreeBSD?

Debian GNU/kFreeBSD — это система Debian GNU с ядром kFreeBSD.

Этот перенос Debian в настоящее время разработан только для архитектур i386 и amd64, хотя возможен перенос и на другие архитектуры.

Заметим, что Debian GNU/kFreeBSD не является системой Linux, и поэтому некоторая информация о Linux к ней неприменима.

Подробности смотрите на [странице переноса Debian GNU/kFreeBSD](#) и в списке рассылки проекта [debian-bsd@lists.debian.org](mailto:debian-bsd@lists.debian.org).

## 1.5 Что такое Debian GNU/Hurd?

Debian GNU/Hurd — это система Debian GNU с GNU Hurd — набором серверов, работающих поверх микроядра GNU Mach.

Проект Hurd пока не завершён и не пригоден для повседневного использования, но работа продолжается. Сейчас Hurd разработан только для архитектуры i386, однако перенос на другие архитектуры будет сделан как только система станет более стабильной.

Заметим, что Debian GNU/Hurd не является системой Linux, и поэтому некоторая информация о Linux к ней неприменима.

Подробности смотрите на [странице Debian GNU/Hurd](#) и в списке рассылки проекта [debian-hurd@lists.debian.org](mailto:debian-hurd@lists.debian.org).

## 1.6 Получение Debian

Информацию о том, как загрузить Debian GNU/Linux из Интернет или у кого можно купить официальные компакт-диски с Debian, смотрите на [странице распространения](#). [Список серверов-зеркал Debian](#) содержит полный перечень официальных зеркал Debian, так что вы можете легко найти ближайший к вам сервер.

После установки Debian может быть легко обновлён. Процедура установки поможет настроить систему так, чтобы вы смогли сделать это обновление после завершения установки, если нужно.

## 1.7 Получение последней версии этого документа

Этот документ постоянно обновляется. На [странице Debian 10](#) вы найдёте самую свежую информацию о последних изменениях в выпуске 10 системы Debian GNU/Linux. Обновлённые версии этого руководства по установке также доступны на [официальных страницах руководства по установке](#).

## 1.8 Структура этого документа

Этот документ может служить руководством для начинающих пользователей Debian. В нём делается несколько предположений о вашем уровне знаний. Однако, мы надеемся, что вы в целом понимаете, как работает аппаратная часть (hardware) вашего компьютера.

Опытные пользователи в этом документе смогут найти ссылки на интересную справочную информацию, например, минимальные размеры установки, детали о поддержке аппаратных средств системой установки Debian и так далее. Мы призываем опытных пользователей хотя бы пробежаться глазами по документу.

Это руководство построено последовательно, оно проведёт вас через процесс установки от начала и до конца. Вот шаги установки Debian GNU/Linux и соответствующие им разделы документа:

1. Определите, соответствует ли ваше аппаратное обеспечение требованиям системы установки (Глава 2).
2. Сделайте резервную копию вашей системы, выполните необходимое планирование и настройку аппаратного обеспечения перед установкой Debian (Глава 3). Если вы подготавливаете мульти-загрузочную систему, то вам, возможно, понадобится выделить место на жёстком диске под раздел Debian.
3. Выбрав способ установки, получите необходимые файлы (Глава 4).
4. The next Глава 5 describes booting into the installation system. This chapter also discusses troubleshooting procedures in case you have problems with this step.
5. Выполните саму установку (Глава 6). Сюда входит: выбор языка установки, настройка модулей драйверов периферийных устройств, настройка сетевых соединений для загрузки оставшихся файлов установки прямо с сервера Debian (если установка производится не с набора CD/DVD), создание разделов на жёстком диске, установка базовой системы, а затем выбор и установка задач (некоторые советы по настройке разделов для системы Debian можно найти в приложении, смотрите Приложение C).
6. Загрузите вашу только что установленную базовую систему (Глава 7).

Глава 8 предназначена для чтения после установки системы. В ней рассказывается о том, где найти дополнительную информации о Unix и Debian, и как заменить ядро.

И наконец, для информации об этом документе и о том как помочь его развитию, смотрите Приложение E.

## 1.9 Помогите нам с документацией!

Мы с радостью примем любого рода помощь, предложения и особенно поправки. Рабочая версия этого документа находится по адресу <http://d-i.alioth.debian.org/manual/>. Там же вы найдёте список всех архитектур, для которых существует этот документ и переводы на другие языки.

Исходный текст этого документа доступен любому желающему. Приложение E содержит информацию о возможном содействии. Если у вас есть предложения, комментарии, конкретные корректировки или сообщения об ошибках, сообщайте нам! Все ошибки желательно заносить в систему отслеживания ошибок на имя пакета `installation-guide`, но сначала проверьте, что об этой ошибке уже не сообщил кто-то другой.

## 1.10 Об авторских правах и лицензиях на программы

Мы уверены, что вы читали некоторые лицензии, на условиях которых распространяется большинство коммерческих программ — обычно в них говорится, что вы можете использовать только одну копию программы на одном компьютере. Лицензия этой системы совсем не такая. Мы предлагаем вам установить копию Debian GNU/Linux на каждый компьютер в вашей школе или на работе. Дайте ваши установочные носители друзьям

и помогите им поставить систему на их компьютеры! Вы даже можете сделать тысячи копий и *продавать* их — хотя и с некоторыми ограничениями. Ваша свобода ставить и использовать систему происходит непосредственно из того, что Debian основан на *свободном программном обеспечении*.

Хотя программы и называются *свободными*, это не означает, что программы не защищены авторскими правами, и это не означает, что CD/DVD, содержащие эти программы, должны распространяться бесплатно. Свободное программное обеспечение, в частности, означает, что лицензии отдельных программ не требуют оплаты за привилегию распространять или использовать эти программы. Свободное программное обеспечение также означает, что кто угодно может не только расширять, адаптировать и изменять программы, но также и распространять результаты своей работы.

#### Замечание



Заметьте, что проект Debian, уступая требованиям пользователей, добавил некоторые пакеты, хотя они и не отвечают нашим критериям свободы. Однако, эти пакеты не являются частью официального дистрибутива, и доступны только из областей **contrib** или **non-free** серверов-зеркал Debian или на неофициальных CD/DVD-ROM; более подробную информацию о расположении и содержании архивов смотрите в разделе «FTP-архивы Debian» в [Debian FAQ](#).

Большинство программ в системе выпущено под *Стандартной Общественной Лицензией GNU*, часто называемой просто «GPL». Лицензия GPL требует от вас сделать *исходный код* программ доступным, если вы распространяете двоичную копию программы; это условие лицензии даёт гарантию того, что любой пользователь сможет изменять программу. Согласно этому положению, в системе Debian доступен исходный код<sup>1</sup> всех таких программ.

Программы в Debian доступны по нескольким соглашениям об авторских правах и программных лицензиях. Сразу после установки в систему, лицензия на пакет сохраняется в файле `/usr/share/doc/имя-пакета/copyright`.

Более полную информацию о лицензиях и о том, как Debian определяет, является ли программа достаточно свободной, чтобы быть включённой в основную (main) часть дистрибутива, смотрите в [критериях Debian для свободного программного обеспечения](#).

Самое важное юридическое замечание — программное обеспечение поставляется *без гарантий*. Программисты, создавшие программу, сделали это на пользу обществу. Не даётся никаких гарантий о пригодности поставляемого программного обеспечения для каких-то определённых целей. Однако, поскольку программное обеспечение свободно, вы вольны модифицировать программу для достижения необходимых вам целей — и наслаждаться выгодами изменений, сделанных другими людьми, которые расширили возможности программы тем же путём.

<sup>1</sup> Информацию о том как найти, распаковать и собрать двоичные программы из исходных кодов Debian, смотрите в [Debian FAQ](#), в разделе «Основы системы управления пакетами Debian».

## Глава 2

# Системные требования

Этот раздел содержит информацию о том, какое оборудование требуется для запуска Debian. Также, вы найдете ссылки на информацию об оборудовании, поддерживаемом GNU и Linux.

### 2.1 Поддерживаемое оборудование

Для Debian не требуется от оборудования сверх того, что требуют ядро Linux или kFreeBSD и утилиты GNU. Таким образом, любая архитектура или платформа, на которую были перенесены ядро Linux или kFreeBSD, libc, gcc и т.д. и на которую перенесён Debian, может работать под Debian. Сверьтесь со страницами переносов <http://www.debian.org/ports/arm/>, какие системы на архитектуре 32-bit hard-float ARMv7 были протестированы с Debian GNU/Linux.

Вместо того, чтобы пытаться описать всё разнообразие аппаратных конфигураций, которое существует на 32-bit hard-float ARMv7, эта глава содержит общую информацию и указания, где можно найти дополнительную информацию.

#### 2.1.1 Поддерживаемые архитектуры

Debian GNU/Linux 10 supports ten major architectures and several variations of each architecture known as «flavors».

Архитектура	Обозначение в Debian	Субархитектура	Вариант
основанные на Intel x86	i386	default x86 machines	default
		Xen PV domains only	xen
AMD64 & Intel 64	amd64		
ARM	armel	Marvell Kirkwood and Orion	marvell
ARM с аппаратным FPU	armhf	multiplatform	armmp
64-битные ARM	arm64		
32bit MIPS (big-endian)	mips	MIPS Malta	4kc-malta
		Cavium Octeon	octeon
64bit MIPS (little-endian)	mips64el	MIPS Malta	5kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
32bit MIPS (little-endian)	mipsel	MIPS Malta	4kc-malta
		Cavium Octeon	octeon
		Loongson 3	loongson-3
Power Systems	ppc64el	IBM POWER8 or newer machines	
64-битный IBM S/390	s390x	IPL с VM-reader и DASD	generic

Этот документ содержит описание установки на архитектуру *32-bit hard-float ARMv7*. Если вы ищете информацию по любой другой архитектуре, поддерживаемой Debian, посмотрите на странице [переносов Debian](#).



### 2.1.2 Three different ARM ports

The ARM architecture has evolved over time and modern ARM processors provide features which are not available in older models. Debian therefore provides three ARM ports to give the best support for a very wide range of different machines:

- Debian/armel targets older 32-bit ARM processors without support for a hardware floating point unit (FPU),
- Debian/armhf works only on newer 32-bit ARM processors which implement at least the ARMv7 architecture with version 3 of the ARM vector floating point specification (VFPv3). It makes use of the extended features and performance enhancements available on these models.
- Debian/arm64 works on 64-bit ARM processors which implement at least the ARMv8 architecture.

Technically, all currently available ARM CPUs can be run in either endian mode (big or little), but in practice the vast majority use little-endian mode. All of Debian/arm64, Debian/armhf and Debian/armel support only little-endian systems.

### 2.1.3 Variations in ARM CPU designs and support complexity

ARM systems are much more heterogeneous than those based on the i386/amd64-based PC architecture, so the support situation can be much more complicated.

The ARM architecture is used mainly in so-called «system-on-chip» (SoC) designs. These SoCs are designed by many different companies with vastly varying hardware components even for the very basic functionality required to bring the system up. System firmware interfaces have been increasingly standardised over time, but especially on older hardware firmware/boot interfaces vary a great deal, so on these systems the Linux kernel has to take care of many system-specific low-level issues which would be handled by the mainboard's BIOS in the PC world.

At the beginning of the ARM support in the Linux kernel, the hardware variety resulted in the requirement of having a separate kernel for each ARM system in contrast to the «one-fits-all» kernel for PC systems. As this approach does not scale to a large number of different systems, work was done to allow booting with a single ARM kernel that can run on different ARM systems. Support for newer ARM systems is now implemented in a way that allows the use of such a multiplatform kernel, but for several older systems a separate specific kernel is still required. Because of this, the standard Debian distribution only supports installation on a selected number of such older ARM systems, alongside the newer systems which are supported by the ARM multiplatform kernels (called «armmp») in Debian/armhf.

### 2.1.4 Платформы, поддерживаемые Debian/armhf

Следующие системы работают с Debian/armhf на мультиплатформенном (armmp) ядре:

**Freescale MX53 Quick Start Board (MX53 LOCO Board)** Плата разработчика IMX53QSB на процессоре i.MX53.

**Versatile Express** The Versatile Express is a development board series from ARM consisting of a baseboard which can be equipped with various CPU daughter boards.

**Встраиваемые системы и некоторые платы разработчика на основе Allwinner sunXi** The armmp kernel supports several development boards and embedded systems based on the Allwinner A10 (architecture codename «sun4i»), A10s/A13 (architecture codename «sun5i»), A20 (architecture codename «sun7i»), A31/A31s (architecture codename «sun6i») and A23/A33 (part of the «sun8i» family) SoCs. Full installer support (including provision of ready-made SD card images with the installer) is currently available for the following sunXi-based systems:

- Cubietech Cubieboard 1 + 2 / Cubietruck
- LeMaker Banana Pi and Banana Pro
- LinkSprite pcDuino and pcDuino3
- Olimex A10-Olinuxino-LIME / A20-Olinuxino-LIME / A20-Olinuxino-LIME2 / A20-Olinuxino Micro / A20-SOM-EVB
- Xunlong OrangePi Plus

System support for Allwinner sunXi-based devices is limited to drivers and device-tree information available in the mainline Linux kernel. Vendor-specific kernel trees (such as the Allwinner SDK kernels) and the android-derived linux-sunxi.org kernel 3.4 series are not supported by Debian.

The mainline Linux kernel generally supports serial console, ethernet, SATA, USB and MMC/SD-cards on Allwinner A10, A10s/A13, A20, A23/A33 and A31/A31s SoCs. The level of support for local display (HDMI/VGA/LCD) and audio hardware varies between individual systems. For most systems, the kernel doesn't have native graphics drivers but instead uses the «simplefb» infrastructure in which the bootloader initializes the display and the kernel just re-uses the pre-initialized framebuffer. This generally works reasonably well, although it results in certain limitations (the display resolution cannot be changed on the fly and display powermanagement is not possible).

Onboard flash memory intended to be used as a mass storage device generally exists in two basic variants on sunXi-based systems: raw NAND flash and eMMC flash. Most older sunXi-based boards with onboard flash storage use raw NAND flash for which support is not generally available in the mainline kernel and therefore also not in Debian. A number of newer systems use eMMC flash instead of raw NAND flash. An eMMC flash chip basically appears as a fast, non-removable SD card and is supported in the same way as a regular SD card.

The installer includes basic support for a number of sunXi-based systems not listed above, but it is largely untested on those systems as the Debian project doesn't have access to the corresponding hardware. No pre-built SD card images with the installer are provided for those systems. Development boards with such limited support include:

- Olimex A10s-Olinuxino Micro / A13-Olinuxino / A13-Olinuxino Micro
- Sinovoip BPI-M2 (A31s-based)
- Xunlong Orange Pi (A20-based) / Orange Pi Mini (A20-based)

In addition to the SoCs and systems listed above, the installer has very limited support for the Allwinner H3 SoC and a number of boards based on it. Mainline kernel support for the H3 is still largely work in progress at the time of the Debian 9 release freeze, so the installer only supports serial console, MMC/SD and the USB host controller on H3-based systems. There is no driver for the on-board ethernet port of the H3 yet, so networking is only possible with a USB ethernet adaptor or a USB wifi dongle. Systems based on the H3 for which such very basic installer support is available include:

- FriendlyARM NanoPi NEO
- Xunlong Orange Pi Lite / Orange Pi One / Orange Pi PC / Orange Pi PC Plus / Orange Pi Plus / Orange Pi Plus 2E / Orange Pi 2

**NVIDIA Jetson TK1** The NVIDIA Jetson TK1 is a developer board based on the Tegra K1 chip (also known as Tegra 124). The Tegra K1 features a quad-core 32-bit ARM Cortex-A15 CPU and Kepler GPU (GK20A) with 192 CUDA cores. Other systems based on the Tegra 124 may work, too.

**Seagate Personal Cloud and Seagate NAS** The Seagate Personal Cloud and Seagate NAS are NAS devices based on Marvell's Armada 370 platform. Debian supports the Personal Cloud (SRN21C), Personal Cloud 2-Bay (SRN22C), Seagate NAS 2-Bay (SRPD20) and Seagate NAS 4-Bay (SRPD40).

**SolidRun Cubox-i2eX / Cubox-i4Pro** The Cubox-i series is a set of small, cubical-shaped systems based on the Freescale i.MX6 SoC family. System support for the Cubox-i series is limited to drivers and device-tree information available in the mainline Linux kernel; the Freescale 3.0 kernel series for the Cubox-i is not supported by Debian. Available drivers in the mainline kernel include serial console, ethernet, USB, MMC/SD-card and display support over HDMI (console and X11). In addition to that, the eSATA port on the Cubox-i4Pro is supported.

**Wandboard** The Wandboard Quad, Dual and Solo are development boards based on the Freescale i.MX6 Quad SoC. System support is limited to drivers and device-tree information available in the mainline Linux kernel; the wandboard-specific 3.0 and 3.10 kernel series from wandboard.org are not supported by Debian. The mainline kernel includes driver support for serial console, display via HDMI (console and X11), ethernet, USB, MMC/SD, SATA (Quad only) and analog audio. Support for the other audio options (S/PDIF, HDMI-Audio) and for the onboard WLAN/Bluetooth module is untested or not available in Debian 9.

Generally, the ARM multiplatform support in the Linux kernel allows running `debian-installer` on `armhf` systems not explicitly listed above, as long as the kernel used by `debian-installer` has support for the target system's components and a device-tree file for the target is available. In these cases, the installer can usually provide a working installation, but it may not be able to automatically make the system bootable. Doing that in many cases requires device-specific information.

When using `debian-installer` on such systems, you may have to manually make the system bootable at the end of the installation, e.g. by running the required commands in a shell started from within `debian-installer`.

### 2.1.5 Платформы, больше не поддерживаемые Debian/armhf

**EfikaMX** The EfikaMX platform (Genesi Efika Smartbook and Genesi EfikaMX nettop) was supported in Debian 7 with a platform-specific kernel, but is no longer supported from Debian 8 onwards. The code required to build the formerly used platform-specific kernel has been removed from the upstream Linux kernel source in 2012, so Debian cannot provide newer builds. Using the armmp multiplatform kernel on the EfikaMX platform would require device-tree support for it, which is currently not available.

### 2.1.6 Несколько процессоров

На этой архитектуре поддерживается нескольких процессоров — так называемая «симметричная многопроцессорная обработка (symmetric multi-processing)» или SMP. Стандартное ядро Debian 10 собрано с поддержкой *SMP-alternatives*. Это означает, что ядро определит число процессоров (или процессорных ядер) и автоматически выключит SMP в однопроцессорных системах.

Раньше, несколько процессоров имелось только в высокопроизводительных серверных системах, но в настоящее время так называемые «многоядерные» процессоры встраивают почти по всё. В них содержится один ЦП с двумя и более вычислительными блоками, называемыми «ядрами».

### 2.1.7 Graphics Hardware Support

Debian's support for graphical interfaces is determined by the underlying support found in X.Org's X11 system, and the kernel. Basic framebuffer graphics is provided by the kernel, whilst desktop environments use X11. Whether advanced graphics card features such as 3D-hardware acceleration or hardware-accelerated video are available, depends on the actual graphics hardware used in the system and in some cases on the installation of additional «firmware» images (see Раздел 2.2).

Nearly all ARM machines have the graphics hardware built-in, rather than being on a plug-in card. Some machines do have expansion slots which will take graphics cards, but that is a rarity. Hardware designed to be headless with no graphics at all is quite common. Whilst basic framebuffer video provided by the kernel should work on all devices that have graphics, fast 3D graphics invariably needs binary drivers to work. The situation is changing quickly but at the time of the buster release free drivers for nouveau (Nvidia Tegra K1 SoC) and freedreno (Qualcomm Snapdragon SoCs) are available in the release. Other hardware needs non-free drivers from 3rd parties.

Details on supported graphics hardware and pointing devices can be found at <http://xorg.freedesktop.org/>. Debian 10 ships with X.Org version 7.7.

### 2.1.8 Аппаратура для подключения к сети

Почти любая сетевая плата (NIC), поддерживаемая ядром Linux, должна поддерживаться системой установки; драйверы модулей должны загрузиться автоматически.

На 32-bit hard-float ARMv7 поддерживается большинство встроенных устройств Ethernet и предоставляются модули для дополнительных устройств PCI и USB.

### 2.1.9 Периферия и другое оборудование

Linux поддерживает много разных устройств, таких как мыши, принтеры, сканеры, PCMCIA/CardBus/ExpressCard и USB устройства. Однако, большинство этих устройств не требуется для установки системы.

## 2.2 Устройства, которым требуются микропрограммы

Для того, чтобы устройства начали работать, кроме драйверов в некоторое аппаратное обеспечение также требуется загрузить так называемую *микропрограмму* или *микрокод*. Это требуется почти для всех сетевых карт (особенно для беспроводных сетей), а также для некоторых устройств USB и даже контроллеров жёстких дисков. Основные возможности многих графических карт доступны без дополнительной микропрограммы, но для включения расширенных возможностей требуется установить в систему файл с соответствующей микропрограммой.

Во многих старых устройствах, которым для работы требовалась микропрограмма, данные файла микропрограммы постоянно хранились в EEPROM/чипе Flash на самом устройстве. В настоящее время в новые устройства микропрограмма больше не встраивается; её код загружается в устройство операционной системы при каждом запуске.

В большинстве случаев микропрограмма является несвободной, если исходить из критериев, используемых проектом Debian GNU/Linux, и поэтому не может быть включена в основной дистрибутив или в систему установки. Если сам драйвер устройства включён в дистрибутив и если для Debian GNU/Linux разрешено легально распространять микропрограмму, то а, чаще всего, доступна в виде отдельного пакета в архиве в разделе non-free.

Однако, это не означает, что такое аппаратное обеспечение не может использоваться во время установки. Начиная с Debian GNU/Linux 5.0, `debian-installer` поддерживает загрузку файлов микропрограмм или пакетов с микропрограммами со сменных носителей, таких как USB-накопители. Подробнее о том, как загружать файлы микропрограмм и пакеты во время установки см. Раздел 6.4.

Если `debian-installer` просит файл с микропрограммой, а у вас этого файла нет или вы не хотите его устанавливать в систему из non-free, то вы можете попробовать продолжить установку без загрузки микропрограммы. Иногда драйвер просит дополнительную микропрограмму, так как она может ему понадобиться в определённых ситуациях, но в большинстве систем устройство всё равно будет работать без неё (например, это происходит с некоторыми сетевыми картами, использующими драйвер `tg3`).

## 2.3 Покупка аппаратного обеспечения специально для GNU/Linux

Некоторые продавцы предлагают системы с уже установленным Debian или с другими дистрибутивами GNU/Linux. Может быть, за это вам придётся заплатить чуть больше, но это плата за душевный покой, так как в этом случае можно быть уверенным в том, что данное аппаратное обеспечение хорошо поддерживается GNU/Linux.

Покупаете ли вы машину с установленным Linux или нет, или даже поддержанную систему, всё равно, важно проверить, что аппаратное обеспечение поддерживается ядром Linux. Убедитесь, что ваше аппаратное обеспечение находится в списках по ссылке, приведённой выше. Скажите продавцу (если он есть), что вы покупаете систему для Linux. Поддержите дружественных к Linux продавцов аппаратного обеспечения.

### 2.3.1 Избегайте собственного или закрытого аппаратного обеспечения

Некоторые производители аппаратного обеспечения просто не говорят нам как писать драйверы для своего аппаратного обеспечения. Другие не дают нам доступ к документации без соглашения о неразглашении, которое запрещает нам выпускать исходный код, который является основой свободного ПО. Так как у нас нет доступа рабочей документации на эти устройства, они просто не будут работать под Linux.

Во многих случаях есть стандарты (или, по крайней мере, некоторые стандарты де-факто), описывающие как операционная система и её драйверы устройств работают с определённым классом устройств. Все устройства, которые соответствуют такому (де-факто)стандарту, можно использовать через единственный универсальный драйвер устройства и никакие специальные драйверы для устройств не требуются. Такие классы оборудования (например, USB «Human Interface Devices», т.е. клавиатуры, мыши и т. д., и USB-устройства хранения, такие как флешки USB и устройства чтения карт памяти) очень хорошо работают и практически каждое устройство, продаваемое на рынке, соответствует стандартам.

В других категориях, среди них например, принтеры, это, к сожалению, не так. В то время как есть много принтеров, с которыми можно работать через несколько (де-факто) стандартных языков управления и поэтому их можно заставить работать без проблем в любой операционной системе, есть довольно много моделей со своими закрытыми командами управления, на которые нет документации. Такие принтеры вообще не могут быть использованы в свободных операционных системах, или могут использоваться только через закрытые драйверы без исходного кода, предоставляемые производителями.

Даже если производитель предоставляет драйвер с закрытым исходным кодом для оборудования при покупке устройства, срок эксплуатации устройства ограничен доступностью драйвера. В настоящее время циклы поддержки продуктов стали короткими, и весьма распространена ситуация, когда через короткий промежуток времени после прекращения производства потребительского устройства, обновление драйверов производитель также перестаёт выполнять. Если старый драйвер с закрытым исходным кодом после обновления системы перестал работать, совершенно рабочее устройство становится бесполезным хламом из-за отсутствия поддержки драйвера, и в этом случае ничего нельзя сделать. Поэтому вы не должны покупать закрытое оборудование, независимо от операционной системы, в которой будете его использовать.

Вы можете помочь решить эту проблему, призвав производителей закрытого оборудования выпускать документацию и другие ресурсы, необходимые для написания свободных драйверов для их оборудования.

## 2.4 Установочные носители

Этот раздел поможет определить, какие типы носителей вы можете использовать для установки Debian. Существует целая глава, целиком посвящённая выбору носителей (Глава 4), в которой перечислены преимущества и недостатки каждого типа. Вы можете вернуться к этой странице, когда прочитаете эту главу.

### 2.4.1 CD-ROM/DVD-ROM/BD-ROM

#### Замечание



Whenever you see «CD-ROM» in this manual, it applies to all of CD-ROMs, DVD-ROMs and BD-ROMs, because all these technologies are really the same from the operating system's point of view.

Установка с CD-ROM поддерживается для большинства архитектур.

### 2.4.2 Сеть

Для получения требуемых для установки файлов может быть использована сеть. Будет или нет задействована сеть зависит от выбранного метода и ваших ответов на некоторые вопросы, которые будут заданы во время установки. Система установки поддерживает большинство типов подключения к сети (включая PPPoE, но не ISDN или PPP), и работает по HTTP или FTP. Настройку ISDN и PPP вы сможете выполнить после завершения установки.

You can also *boot* the installation system over the network without needing any local media like CDs/DVDs or USB sticks. If you already have a netboot-infrastructure available (i.e. you are already running DHCP and TFTP services in your network), this allows an easy and fast deployment of a large number of machines. Setting up the necessary infrastructure requires a certain level of technical experience, so this is not recommended for novice users.

Ещё один вариант — бездисковая установка с использованием загрузки по локальной сети и монтированием всех локальных файловых систем через NFS.

### 2.4.3 Жёсткий диск

Загрузка системы установки непосредственно с жёсткого диска — это ещё один вариант, работающий на многих архитектурах. Он требует наличия какой-нибудь другой операционной системы для загрузки программы установки на жёсткий диск. Данный метод рекомендуется использовать только в особых случаях, если недоступны остальные методы.

### 2.4.4 Система Un\*x или GNU

Если вы работаете в другой Unix системе, то можете использовать её для установки Debian GNU/Linux без использования `debian-installer`, описанного в остальной части руководства. Этот тип установки может пригодиться пользователям с каким-то неподдерживаемым оборудованием или на машинах, работу которых нельзя прерывать. Если вы заинтересовались данным способом, перейдите в Раздел D.3. Данный метод установки рекомендуется применять только опытным пользователям, если недоступны остальные методы.

### 2.4.5 Поддерживаемые системы хранения

Программа установки Debian содержит ядро, при сборке которого главным было увеличение количества систем, на которых бы оно запускалось.

## 2.5 Требования к памяти и дисковому пространству

Для выполнения обычной установки у вас должно быть 31МВ оперативной памяти и 780МВ на жёстком диске. Заметим, что это действительно самый минимум, который должен быть. Более реалистичные данные смотрите в Раздел 3.4.

Installation on systems with less memory or disk space available may be possible but is only advised for experienced users.

## Глава 3

# Перед установкой Debian GNU/Linux

В этой главе рассказывается о том, что нужно сделать перед запуском программы установки Debian: резервное копирование данных, сбор информации об оборудовании и поиск необходимой информации.

### 3.1 Обзор процесса установки

Во-первых, немного о переустановках. Случаи, когда приходится полностью переустанавливать Debian очень редки; наиболее часто это делается при механической поломке жёсткого диска.

Для большинства операционных систем при критических сбоях или для обновления до новой версии может требоваться выполнение полной установки. Даже если новая установка не требуется, часто приходится переустанавливать некоторые программы, чтобы они заработали в новой версии ОС.

В случае Debian GNU/Linux у вас намного больше шансов, что ОС будет восстановлена без переустановки. При обновлении до новой версии установка "всего" не требуется никогда; достаточно обновления только нужных частей. Все программы почти всегда совместимы с новыми версиями операционной системы. Если новая версия программы требует более нового программного обеспечения, система управления пакетами Debian обеспечивает автоматическое обнаружение и установку всего необходимого программного обеспечения. Много труда было вложено именно в то, чтобы избежать переустановки, поэтому думайте об этом как о последней причине вашего выбора. Программа установки *не* разрабатывалась для переустановки поверх существующей системы.

Вот путь, который вам предстоит совершить.

1. Сделать резервную копию существующих данных и документов жёсткого диска, на который вы планируете выполнить установку.
2. Перед началом установки собрать информацию о компьютере и всю необходимую документацию.
3. Выделить место под Debian на жёстком диске.
4. Найти и/или скачать программу установки и все нужные для вашей машины специальные драйверы или файлы микропрограмм.
5. Создать загрузочный медианоситель, такой как CDs/DVDs/USB, или организовать инфраструктуру загрузки по сети, с помощью которой можно будет запустить программу установки.
6. Загрузить систему установки.
7. Выбрать язык установки.
8. Активировать, если есть, подключение к сети ethernet.
9. Создать и смонтировать разделы, на которые будет установлен Debian.
10. Присматривать за автоматической загрузкой/установкой/настройкой *базовой системы*.
11. Установить *системный загрузчик*, который сможет запускать Debian GNU/Linux и/или другую существующую систему.
12. Загрузить только что установленную систему в первый раз.

Если у вас возникли проблемы во время установки, хорошо иметь представление о том, какими пакетами это могло быть вызвано и на каком шаге. Представляем основных актёров от программного обеспечения в драме "установка":

Программа установки, `debian-installer` — это главное, о чём писалось данное руководство. Она обнаруживает аппаратное обеспечение и загружает соответствующие драйверы, использует `dhcp-client` для настройки подключения к сети, запускает `debootstrap` для установки пакетов базовой системы и запускает `tasksel` для того, чтобы позволить вам выбрать установку дополнительного ПО. Намного больше актёров играют меньшие роли в этом процессе, а `debian-installer` завершает свою работу после первой загрузки операционной системы.

Для настройки системы под ваши нужды служит программа `tasksel`, которая позволяет выбрать для установки различные готовые группы взаимосвязанных программ под задачу, например, для веб сервера или настольного компьютера.

Одним из важных пунктов установки является выбор будет или нет установлено графическое окружение рабочего стола, состоящее из X Window System и одного из доступных графических окружений рабочего стола. Если вы не выберете задачу «Окружение рабочего стола», то у вас получится относительно простая система, управляемая из командной строки. Установка окружения рабочего стола является необязательной задачей, так как, по сравнению с системой без графики, для неё требуется относительно большое количество дискового пространства, и из-за того, что многие системы Debian GNU/Linux являются серверами, которым не нужен графический интерфейс пользователя для выполнения требуемых задач.

Оконная система X — это полностью отдельная программа, никак не связанная с `debian-installer`, и фактически, намного более сложная. Устранение проблем X Window System не предмет рассмотрения данного руководства.

## 3.2 Сохраните ваши данные в надёжном месте!

Перед тем как начать, убедитесь, что вы сохранили каждый файл из вашей системы. Если вы впервые устанавливаете на компьютер не родную операционную систему, скорее всего вам придётся переразметить диск, чтобы выделить место для Debian GNU/Linux. Всегда при разметке диска вы должны считать, что теряете всё на диске, вне зависимости от используемого для этого инструмента. Программы, используемые в установке, достаточно надёжны; с большинством из них люди работают не первый год; но они достаточно мощны для того, чтобы одно неосторожное движение дорого вам обошлось. Даже после сохранения данных будьте осторожны и думайте над даваемыми ответами и действиями. Две минуты раздумий могут спасти вас от нескольких часов ненужной работы.

Если вы создаёте мульти-загрузочную систему, убедитесь, что у вас под рукой есть носители с дистрибутивами всех операционных систем, присутствующих на диске. Иногда может возникнуть ситуация, когда вам потребуется переустановить системный загрузчик операционной системы, чтобы система могла загружаться или, в худшем случае, полностью переустановить операционную систему и восстановить все файлы из резервных копий.

## 3.3 Информация, которая понадобится

### 3.3.1 Документация

#### 3.3.1.1 Руководство по установке

Документ, который вы сейчас читаете, является разрабатываемой версией руководства по установке Debian следующей версии и доступен в [различных форматах и на разных языках](#).

#### 3.3.1.2 Техническая документация

Часто содержит полезную информацию по настройке и использованию вашего аппаратного обеспечения.

### 3.3.2 Поиск источников технической информации

В большинстве случаев программа установки способна определить используемое оборудование автоматически. Но всё же мы рекомендуем вам перед установкой самостоятельно ознакомиться с имеющимся оборудованием.

Информацию по аппаратному обеспечению можно найти:



- В руководствах, которые были вместе с приобретённым аппаратным обеспечением.
- В настройках BIOS. Вы можете увидеть эти настройки, если нажмёте комбинацию клавиш во время загрузки компьютера. Посмотрите в руководстве нужную комбинацию. Часто это клавиша **Delete** или **F2**, но некоторые производители используют другие клавиши или комбинации. Обычно при включении компьютера появляется сообщение, в котором показано, что нужно нажать, чтобы попасть в BIOS.
- На коробках и ящиках от аппаратного обеспечения.
- С помощью системных команд и инструментов других операционных систем, включая файловый менеджер. Это особенно полезный источник информации об оперативной памяти и жёстких дисках.
- Узнать у вашего системного администратора или провайдера услуг Интернет. Они могут рассказать вам о настройках почты и подключении к сети.

**Таблица 3.1** Информация об оборудовании, полезная при установке

Оборудование	Информация, которая может потребоваться
Жёсткие диски	Количество дисков.
	Их порядок в системе.
	IDE (тн. PATA), SATA или SCSI.
	Доступное дисковое пространство.
	Разделы.
	Разделы, на которых установлены операционные системы.
Сетевые интерфейсы	Тип/модель доступных сетевых интерфейсов.
Принтер	Модель и производитель.
Видеокарта	Тип/модель и производитель.

### 3.3.3 Совместимость аппаратного обеспечения

Многие устройства без проблем работают с Linux. Кроме того, поддержка оборудования в Linux ежедневно улучшается. Однако, Linux всё равно пока не может сравниться по количеству поддерживаемой аппаратуры с другими операционными системами.

Драйверы в Linux, в большинстве случаев, пишутся не для определённого «продукта» или «марки» какого-то производителя, а под определённое оборудование/набор микросхем. Много кажущихся различными продуктов/марок основаны на одном аппаратном проекте; часто бывает, что производитель микросхемы предоставляет так называемый «типовой вариант дизайна» для продуктов, основанных на своих микросхемах, который затем используется различными производителями устройств для выпуска различных продуктов или марок.

Это имеет свои преимущества и недостатки. Преимущество состоит в том, что драйвер для одной микросхемы работает для большого количества различных продуктов от различных производителей, пока их продукт основывается на этой микросхеме. Недостатком является то, что не всегда просто определить, какая микросхема фактически используется в определённом продукте/марке. К сожалению, иногда производители устройства изменяют аппаратную основу продукта, не изменяя название или, по крайней мере, номер версии, поэтому иногда две модели одной марки/продукта, купленных в разное время, могут основываться на различных микросхемах, и поэтому нужно использовать два различных драйвера, или для одного из них драйвер вообще может отсутствовать.

Чтобы выяснить на какой микросхеме основаны устройства USB и PCI/PCI-Express/ExpressCard достаточно найти идентификатор устройства. Все устройства USB/PCI/PCI-Express/ExpressCard имеют так называемый идентификатор «производителя» и «продукта», и их комбинация, обычно, одинакова для продукта, основанного на одной модели микросхемы.

В системах Linux эти идентификаторы можно узнать с помощью команды **lsusb** для устройств USB и с помощью команды **lspci -nn** для устройств for PCI/PCI-Express/ExpressCard. Идентификаторы производителя и продукта, обычно, показываются в виде двух шестнадцатеричных чисел, разделённых двоеточием, например «1d6b:0001».

Пример вывода команды **lsusb**: «Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub», где 1d6b — идентификатор производителя и 0002 — идентификатор продукта.

Пример вывода команды `lspci -nn` для карты Ethernet: «03:00.0 Ethernet controller [0200]: Realtek Semiconductor Co., Ltd. RTL8111/8168B PCI Express Gigabit Ethernet controller [10ec:8168] (rev 06)». Идентификаторы указаны внутри последних квадратных скобок, т. е. здесь 10ec — идентификатор производителя и 8168 — продукта.

Другой пример, графическая карта: «04:00.0 VGA compatible controller [0300]: Advanced Micro Devices [AMD] nee ATI RV710 [Radeon HD 4350] [1002:954f]».

В Windows идентификаторы устройства можно найти в менеджере устройств Windows во вкладке «подробней», где идентификатор производителя начинается с VEN\_, а идентификатор продукта начинается с DEV\_. Чтобы увидеть идентификаторы в Windows 7 вы должны выбрать свойство «Идентификаторы оборудования» в менеджере устройств на вкладке подробностей, так как по умолчанию они не отображаются.

Поиск в Интернет по идентификатору производителя/продукта, «Linux» и «driver» часто даёт информацию о состоянии поддержки драйвером определённой микросхемы. Если поиск идентификатору производителя/продукта не дал желаемых результатов, то может помочь поиск по названию микросхемы, которое часто выдаётся `lsusb` и `lspci` («RTL8111»/«RTL8168B» — в примере с сетевой картой и «RV710» — в примере с графической картой).

### 3.3.3.1 Тестирование оборудования на совместимость с помощью Live-системы

Для некоторых архитектур Debian GNU/Linux доступен в виде так называемой «живой системы». Живая система (live system) — это настроенная, готовая к работе система, предоставляемая в сжатом виде, которая может быть загружена и использована с носителя, недоступного для записи, такого как CD или DVD. По умолчанию при её работе на компьютер не вносятся никаких постоянных изменений. Вы можете менять настройки пользователя и устанавливать дополнительные программы внутрь живой системы, но всё это выполняется только в оперативной памяти, т. е. при выключении компьютера и загрузки живой системы снова все изменения теряются. Если вы хотите проверить, поддерживается ли ваше оборудование Debian GNU/Linux, то самым простым способом сделать это будет попробовать запустить живую систему Debian.

Есть несколько ограничений при использовании живой системы. Прежде всего, так как все изменения живой системы должны быть сохранены в RAM компьютера, она работает только в системах с достаточным количеством RAM, и из-за ограничений памяти нельзя установить дополнительные большие пакеты ПО. Другое ограничение тестирования аппаратной совместимости — живая система состоит только из свободных компонентов, т. е. в ней отсутствуют несвободные файлы микропрограмм. Конечно, эти несвободные пакеты можно установить вручную после запуска системы, но какие именно — это придётся выяснять пользователю, так как в живой системе отсутствует автоматическое определение требуемых микропрограмм (в отличие от `debian-installer`).

Информация о доступных вариантах живых образов Debian приведена на [сайте живых образов Debian](#).

### 3.3.4 Настройки сети

Если компьютер подключён к постоянной сети (т. е. к Ethernet или через аналогичное подключение, но не коммутируемое/PPP соединение), которую обслуживаете не вы, то вы должны попросить у вашего сетевого системного администратора следующую информацию:

- Имя вашей машины (возможно вы можете выбрать его самостоятельно).
- Имя домена.
- IP-адрес компьютера.
- Маска сети.
- IP-адрес шлюза по умолчанию, если в вашей сети *есть* такой шлюз.
- Система в сети, которая используется как DNS (Domain Name Service) сервер.

Если в используемой сети для получения сетевых настроек используется DHCP (Dynamic Host Configuration Protocol), то вам не нужна вся эта информация, так как DHCP-сервер передаст её вашему компьютеру автоматически во время процесса установки.

Если вы подключены к Интернету по DSL или кабельному модему (т. е. через телевизионную кабельную сеть) и используете маршрутизатор (часто его предоставляет уже настроенным телефонный или телевизионный оператор), который осуществляет соединение с сетью, то, обычно, DHCP включён по умолчанию.

Если вы используете беспроводную сеть (WLAN/WiFi), то также нужно выяснить:

- ESSID («имя сети») беспроводной сети.
- Ключ безопасности WEP или WPA/WPA2 для доступа к сети (если применяется).

### 3.4 Минимальные требования к аппаратному обеспечению

После сбора всей информации о компонентах компьютера, проверьте, что ваше аппаратное обеспечение соответствует требованиям желаемого типа установки.

В зависимости от потребностей, у вас может оказаться машина меньшей мощности, чем рекомендовано в таблице ниже. Однако, большинство пользователей рискуют оказаться расстроенными, если проигнорируют эти советы.

**Таблица 3.2** Рекомендуемые минимальные требования к системе

Тип установки	Память (минимум)	Память (рекомендуется)	Жёсткий диск
Без настольных приложений	128 мегабайт	512 мегабайт	2 гигабайта
С настольными приложениями	256 мегабайт	1 гигабайт	10 гигабайт

The actual minimum memory requirements are a lot less than the numbers listed in this table. Depending on the architecture, it is possible to install Debian with as little as 60MB (for amd64). The same goes for the disk space requirements, especially if you pick and choose which applications to install; see Раздел D.2 for additional information on disk space requirements.

It is possible to run a graphical desktop environment on older or low-end systems, but in that case it is recommended to install a window manager that is less resource-hungry than those of the GNOME or KDE Plasma desktop environments; alternatives include `xfce4`, `icewm` and `wmaker`, but there are others to choose from.

Практически невозможно указать требования к диску и памяти для установки серверной конфигурации, так как это в многом зависит от типа задач, выполняемых сервером.

Не забудьте, что в данные размеры не включены размеры файлы, которые обычно лежат на машине: файлы пользователя, почта и данные. Всегда нужно быть более щедрым, когда подсчитываешь место для своих файлов и данных.

Disk space required for the smooth operation of the Debian GNU/Linux system itself is taken into account in these recommended system requirements. Notably, the `/var` partition contains a lot of state information specific to Debian in addition to its regular contents, like logfiles. The `dpkg` files (with information on all installed packages) can easily consume 40MB. Also, `apt` puts downloaded packages here before they are installed. You should usually allocate at least 200MB for `/var`, and a lot more if you install a graphical desktop environment.

### 3.5 Переразметка для мульти-загрузочных систем

Разметка диска — это просто операция деления диска на секции. Каждая выделенная секция не зависит от других. Это, грубо говоря, как построить стены внутри дома; если вы добавите в одну комнату мебель, то это никак не повлияет на другие комнаты.

Если у вас на машине уже установлена операционная система, которая занимает весь диск и вы хотите добавить Debian на тот же диск, то вам нужно его переразмечить. Для Debian нужно создать отдельные разделы на жёстком диске. Он не может быть установлен на разделы Windows или Mac OS X. Он может совместно использовать некоторые разделы с другими системами Unix, но в этом руководстве это не описывается. Как минимум, вам нужно выделить место для корневой файловой системы Debian.

Информацию об имеющихся разделах можно получить с помощью инструментов разметки из установленной операционной системы. Любая утилита разметки умеет показывать существующие разделы без их изменения.

Изменение раздела с уже имеющейся файловой системой уничтожит всю информацию на нём. То есть вы всегда должны сделать резервную копию всей информации перед переразметкой. Используя аналогию с домом, вы вероятно захотите вынести всю мебель перед переносом стен или же вы рискуете попортить её.

В некоторых современных операционных системах можно перемещать и изменять размеры определённых существующих разделов, не уничтожая их содержимого. Это позволяет выделить место для дополнительных

разделов, не теряя существующих данных. Хотя это работает вполне хорошо в большинстве случаев, изменение разметки диска, по сути, опасное действие и должно выполняться только после создания полной резервной копии всех данных.

Создание и удаление разделов можно выполнять из `debian-installer`, а также из уже установленной операционной системы. Как показывает опыт, разделы лучше создавать той системой, в которой они будут использоваться, т. е. разделы для Debian GNU/Linux лучше создавать из `debian-installer`, а разделы для другой операционной системы, лучше создавать штатными средствами другой ОС. `debian-installer` может создавать не Linux разделы, и разделы, созданные им, обычно, работают без проблем в других операционных системах, но есть несколько редких случаев, когда это может вызвать проблемы, поэтому, если вы не хотите рисковать, используйте родные инструменты ОС для создания её разделов.

Если вы хотите установить более одной операционной системы на одну машину, то установите все операционные системы до установки Debian. Windows и другие ОС могут уничтожить возможность запуска Debian или посоветовать вам переформатировать все неизвестные для них разделы.

Вы можете устранить последствия таких действий или избежать их, но установка родной системы в первую очередь оградит вас от подобных проблем.

## 3.6 Настройка аппаратного обеспечения и операционной системы перед установкой

В этом разделе описывается аппаратное обеспечение, для которого требуется настройка перед установкой Debian. Обычно, для этого производится проверка и возможно, изменение настроек BIOS или микропрограммы вашей системы. «BIOS» или «микропрограмма» — это программное обеспечение, используемое аппаратным обеспечением; данные компоненты почти всегда выполняются во время загрузки машины (после подачи питания).

### 3.6.1 Микропрограмма ARM

Как уже упоминалось ранее, в системах ARM отсутствует стандартная системная микропрограмма. Даже в случае одинаковой по имени микропрограммы, её поведение может отличаться. В результате в большинство встраиваемых устройств с архитектурой ARM производители, обычно, включают специально изменённые версии микропрограммы и заплатки для устройств. К сожалению, производители часто не отправляют сделанные изменения и расширения разработчикам микропрограммы, и поэтому в новых версиях оригинальной микропрограммы поддержка не появляется.

Как следствие, даже в совсем новых системах часто используется микропрограмма на основе изменённой производителем версии многолетней давности, в то время как основной версией микропрограммы предлагаются дополнительные возможности или в ней было изменено поведение в работе. В дополнении к этому, в изменённых производителями одной версии микропрограммы используются разные наименования наборных устройств, что не позволяет написать рабочую единую инструкцию для систем ARM.

### 3.6.2 Debian-provided U-Boot (system firmware) images

Debian provides U-Boot images for various armhf systems that can load their U-Boot from an SD card at `.../images/u-boot/`. The U-Boot builds are offered in two formats: raw U-Boot components and a ready-made card image that can easily be written onto an SD card. The raw U-Boot components are provided for advanced users; the generally recommended way is to use one of the ready-made SD card images. They are named `<system-type>.sdcimg.gz` and can be written to a card e.g. with

```
zcat <system-type>.sdcimg.gz > /dev/SD_CARD_DEVICE
```

As with all images, please be aware that writing the image to an SD card wipes all previous contents of the card!

If Debian provides a U-Boot image for your system, it is recommended that you use this image instead of the vendor-provided U-Boot, as the version in Debian is usually newer and has more features.

### 3.6.3 Задание MAC-адреса ethernet в U-Boot

Считается, что MAC-адрес каждого интерфейса ethernet должен быть уникальным глобально, и, технически, уникальным внутри своего широковещательного домена ethernet. Для этого производителю, обычно, выделяется блок MAC-адресов из централизованно-управляемого пространства (платно), и он настраивает один из этих адресов на каждом продаваемом устройстве.

Выпуская платы разработчик производитель, иногда, чтобы не платить, не предоставляет глобально-уникальный адрес. В этих случаях пользователи сами задают MAC-адреса для своих систем. Когда MAC-адрес интерфейса не задан, некоторые сетевые драйверы генерируют произвольный MAC-адрес, который может меняться при каждом включении, и если это происходит, доступ к сети возможен даже если пользователь не ввёл адрес вручную, но при этом, например, назначение IP-адресов по DHCP на основе MAC-адреса, очевидно, не работает.

Чтобы избежать пересечения с официально выданными MAC-адресами существует зарезервированное пространство, так называемых «локально-контролируемых» адресов. Оно задаётся значением двух специальных бит в первом байте адреса (хорошее описание в статье википедии «MAC address»). На практике это означает, что любой адрес, начинающийся с шестнадцатеричного са (например, са:ff:ee:12:34:56) можно использовать как локально-контролируемый адрес.

В устройствах с U-Boot в качестве системной микропрограммы, ethernet MAC-адрес хранится в переменной окружения «ethaddr». Его можно увидеть из приглашения U-Boot по команде «printenv ethaddr» и изменить командой «setenv ethaddr са:ff:ee:12:34:56». После установки значения команда «saveenv» делает его постоянным.

### 3.6.4 Проблемы размещения ядра/Initrd/дерева-устройств в U-Boot

В процессе загрузки в некоторых устройствах со старыми версиями U-Boot могут быть проблемы с правильным размещением в памяти ядра Linux, начального ramdisk и файла дерева устройств. В этот случае U-Boot выводит сообщение «Starting kernel ...», и система повисает ничего больше не выводя. Эти проблемы были решены в новых версиях U-Boot, начиная с v2014.07.

Если в системе сначала использовалась версия U-Boot старше v2014.07 и позже она была обновлена на новую, то проблема может остаться. При обновлении u-boot, обычно, не изменяются существующие переменные окружения U-Boot и для исправления требуется установка дополнительной переменной окружения (bootm\_size), что делается U-Boot автоматически только при установке с нуля, при отсутствии данных окружения. Возможно вручную указать bootm\_size со значением по умолчанию нового U-Boot командой «env default bootm\_size; saveenv» из приглашения U-Boot.

Ещё один способ обхода проблем размещения: выполнить команду «setenv fdt\_high ffffffff; setenv initrd\_high 0xffffffff; saveenv» в приглашении U-Boot, полностью выключающую размещение начального ramdisk и файла дерева устройств.

## Глава 4

# Получение носителя с дистрибутивом

### 4.1 Официальный набор CD/DVD-ROM Debian GNU/Linux

Самым лёгким путём установки Debian GNU/Linux является установка с официального набора CD/DVD-ROM Debian (смотрите [страницу продавцов компакт-дисков Debian](#)). Также, вы можете загрузить образы CD/DVD-ROM с сервера-зеркала Debian и сделать из них набор самостоятельно, если у вас есть высокоскоростное подключение к сети и устройство записи CD/DVD (подробные инструкции смотрите на странице [Debian GNU/Linux на компакт-дисках](#) и [Debian CD FAQ](#)). Если у вас есть официальный набор компакт-дисков Debian и машина способна загружаться с привода CD-ROM, то вы можете сразу перейти к [Глава 5](#). Много усилий было потрачено на то, чтобы расположить самые используемые файлы на первый CD и DVD-диск, поэтому для установки базового рабочего стола достаточно только первого DVD или — в некоторых случаях — даже первого CD.

При сегодняшних объёмах, на CD-диск можно не так много уместить, например, не каждое окружение рабочего стола можно установить имея только первый CD; для установке с CD некоторых окружений рабочего стола требуется или подключение к сети, или дополнительные диски.

Если машина не поддерживает загрузку с компакт-диска, а вы имеете набор CD, то для загрузки программы установки можно воспользоваться другим методом, например, загрузить ядро по сети, или вручную с CD. Для этого на компакт-диске есть все файлы; структура архива Debian в сети и каталогов CD идентичны. Поэтому, когда ниже встречаются пути к файлам в архиве, которые требуются для определённого вида загрузки, поищите эти файлы в тех же каталогах и подкаталогах на ваших CD.

После загрузки программы установки, можно будет загрузить все остальные файлы с компакт-дисков.

Если у вас нет набора CD, то вам потребуется загрузить системные файлы программы установки и сохранить их на подключённый к сети компьютер и использовать их для загрузки программы установки.

### 4.2 Скачивание файлов с серверов-зеркал Debian

Ближайший (и поэтому, вероятно, самый быстрый) к вам сервер-зеркало можно найти в [списке серверов-зеркал Debian](#).

При скачивании файлов с сервера-зеркала Debian по протоколу FTP проверьте, что загружаете файлы в *двоичном*, а не текстовом или автоматическом режиме.

#### 4.2.1 Где искать установочные образы

Установочные образы на сервере-зеркале Debian находятся в каталоге [debian/dists/buster/main/installer-armhf/current/images/](#) — в файле [MANIFEST](#) описано для чего предназначен каждый образ.

##### 4.2.1.1 Мультиплатформенные файлы установки для armhf

К файлам установки для систем, поддерживаемых мультиплатформенным ядром armhf (смотрите Раздел [2.1.4](#)) относится стандартный образ ядра Linux, стандартный образ начального ramdisk Linux и специальный для каждой системы двоичный файл дерева устройств. Образ ядра и начального ramdisk для загрузки по tftp можно получить с [.../images/netboot/](#), двоичный файл дерева устройств можно получить с [.../images/device-tree/](#). Архив tar для создания загрузочного носителя USB с программой установки можно получить с [.../images/hd-media/](#).

U-boot images for various armhf platforms are available at [.../images/u-boot/](#).

## 4.3 Подготовка файлов для загрузки по TFTP

Если ваша машина подключена к локальной сети, то вы можете загрузить её по сети с другой машины через TFTP. Для этого на удалённую машину в определённое место вам нужно поместить загрузочные файлы и настроить поддержку загрузки вашей машины.

Вам нужно настроить TFTP сервер, а если машин много, то DHCP сервер или RARP сервер.

Reverse Address Resolution Protocol (RARP) — это один из путей указать клиенту какой адрес IP для него предназначен. Другой способ — использовать BOOTP протокол. BOOTP — это IP протокол, который информирует компьютер о его IP-адресе и где в сети получить загрузочный образ. DHCP (Dynamic Host Configuration Protocol) более гибок и обратно совместим с BOOTP. Некоторые системы могут быть настроены только через DHCP.

Trivial File Transfer Protocol (TFTP) используется для загрузки загрузочного образа на клиентскую машину. Теоретически, можно использовать любой сервер на любой платформе, которая реализует эти протоколы. В примерах этого раздела мы используем команды из SunOS 4.x, SunOS 5.x (так называемый Solaris) и GNU/Linux.

### 4.3.1 Настройка RARP сервера

Для настройки RARP вам нужно знать Ethernet адрес (так называемый MAC адрес) клиентского компьютера, на который будет выполняться установка. Если вы не знаете адреса, то можете загрузиться в «Rescue» режиме (например, со дискеты восстановления) и выполнить команду `/ip addr show dev eth0`.

В системе с RARP сервером использующей ядро Linux или Solaris/SunOS вы должны использовать программу `rarpd`. Вам нужно проверить, что аппаратный адрес Ethernet клиента содержится в базе данных «ethers» (или в файле `/etc/ethers` или через NIS/NIS+) и в базе данных «hosts». Затем, вам нужно запустить демона RARP. В большинстве Linux систем и SunOS 5 (Solaris 2) выполните команду (от суперпользователя): `/usr/sbin/rarpd -a`; в других Linux системах `/usr/etc/rarpd -a` или `/usr/etc/rarpd -a` в SunOS 4 (Solaris 1).

### 4.3.2 Настройка DHCP сервера

Одним из свободных DHCP серверов является ISC `dhcpd`. В Debian GNU/Linux он доступен из пакета `isc-dhcp-server`. Вот пример его конфигурационного файла (обычно `/etc/dhcp/dhcpd.conf`):

```
option domain-name "example.com";
option domain-name-servers ns1.example.com;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 7200;
server-name "servername";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.200 192.168.1.253;
    option routers 192.168.1.1;
}

host clientname {
    filename "/tftpboot.img";
    server-name "servername";
    next-server servername;
    hardware ethernet 01:23:45:67:89:AB;
    fixed-address 192.168.1.90;
}
```

В этом примере определён единственный сервер `servername`, который работает в качестве DHCP, TFTP серверов и шлюза сети. Вам почти наверняка нужно изменить опцию `domain-name`, а также имя сервера и аппаратный адрес клиента. Опция `filename` должна содержать имя файла, который нужно получить по TFTP.

После редактирования конфигурационного файла для `dhcpd`, перезагрузите сервер командой `/etc/init.d/isc-dhcp-server restart`.

### 4.3.3 Настройка BOOTP сервера

Для GNU/Linux есть два BOOTP сервера. Первый — СМУ **bootpd**. Второй, на самом деле являющийся сервером DHCP — ISC **dhcpd**. В Debian GNU/Linux они находятся в пакетах **bootp** и **isc-dhcp-server** соответственно.

Чтобы использовать СМУ **bootpd**, во-первых, вы должны раскомментировать (или добавить) соответствующую строку в `/etc/inetd.conf`. Для этого в Debian GNU/Linux вы можете запустить **update-inetd --enable bootps**, затем `/etc/init.d/inetd reload`. Если BOOTP сервер работает не под Debian, то строка выглядит так:

```
bootps dgram udp wait root /usr/sbin/bootpd bootpd -i -t 120
```

Теперь вы должны создать файл `/etc/bootptab`. Внутри он напоминает хорошо знакомый и загадочный формат старых добрых BSD файлов `printcap`, `termcap` и `disktab`. Подробности смотрите в справочной странице `bootptab`. Для СМУ **bootpd** вам нужно знать аппаратный адрес (MAC) клиента. Вот пример `/etc/bootptab`:

```
client:\
  hd=/tftpboot:\
  bf=tftpboot.img:\
  ip=192.168.1.90:\
  sm=255.255.255.0:\
  sa=192.168.1.1:\
  ha=0123456789AB:
```

Вам нужно изменить, по крайней мере, параметр «`ha`», который содержит аппаратный адрес клиента. Параметр «`bf`» содержит файл, который клиент должен получить по TFTP; подробности смотрите в Раздел 4.3.5.

Напротив, настройка BOOTP в ISC **dhcpd** очень проста, так как здесь клиенты BOOTP считаются одним из вариантов клиентов DHCP. Некоторые архитектуры требуют сложной настройки для загрузки клиентов по BOOTP. Если у вас один из таких случаев, прочитайте раздел Раздел 4.3.2. Если нет, то достаточно просто добавить директиву **allow bootp** в блок настройки подсети, содержащей клиента, в `/etc/dhcp/dhcpd.conf` и перезапустить **dhcpd** командой `/etc/init.d/isc-dhcp-server restart`.

### 4.3.4 Включение TFTP сервера

Чтобы сервер TFTP заработал, во-первых нужно проверить что включена **tftpd**.

Программа `tftpd-hpa` может быть запущена двумя способами. Она может запускаться по требованию службой `inetd`, или может быть настроена для работы как независимая служба. Выбор метода происходит при при установке пакета, и его можно изменить через перенастройку пакета.

#### Замечание



Исторически, TFTP-серверы используют каталог `/tftpboot` для хранения образов. Однако, пакеты Debian GNU/Linux могут использовать другие каталоги, чтобы соответствовать [Filesystem Hierarchy Standard](#). Например, `tftpd-hpa` по умолчанию использует `/srv/tftp`. Вам может потребоваться изменить примеры конфигурации для соответствия.

По умолчанию, все альтернативные программы **in.tftpd**, доступные в Debian, протоколируют запросы TFTP в системный журнал. Некоторые из них имеют параметр `-v`, позволяющий включить более подробный протокол. Рекомендуется проверить эти сообщения в журнале в случае возникновения проблем с загрузкой; это хорошая отправная точка при поиске причин ошибок.

### 4.3.5 Копирование TFTP образов в каталог TFTP сервера

Далее, поместите нужный загрузочный образ TFTP из Раздел 4.2.1 в каталог загрузочных образов **tftpd**. Вы можете сделать ссылку на этот файл для файла, который **tftpd** будет передавать для загрузки определённому клиенту. К сожалению, имя файла зависит от клиента TFTP и никак не стандартизовано.



## 4.4 Автоматическая установка

Для установки на несколько компьютеров можно воспользоваться полностью автоматической установкой. Необходимые пакеты Debian: `fai-quickstart` (используется сервером установки) и сама программа установки Debian. Подробная информация представлена на [домашней странице FAI](#).

### 4.4.1 Автоматическая установка с помощью программы установки Debian

Программа установки Debian поддерживает автоматическую установку через файлы автоматической установки. Файл автоматической установки может быть загружен по сети или со сменного носителя, и используется для ответов на вопросы, задаваемые во время процесса установки.

Полную документацию по автоматической установке и рабочий пример, который вы можете отредактировать, находится в Приложении В.

## Глава 5

# Загрузка системы установки

### 5.1 Запуск программы установки на 32-bit hard-float ARMv7

#### 5.1.1 Форматы загрузочных образов

В системах на базе ARM в большинстве случаев используется один из двух форматов загрузочных образов: а) стандартные ядра Linux в формате zImage («vmlinuz») в вместе со стандартными начальными Linux ramdisk («initrd.gz») или б) ядра в формате uImage («uImage») вместе с соответствующими начальными ramdisk («uInitrd»).

uImage/uInitrd are image formats designed for the U-Boot firmware that is used on many ARM-based systems (mostly 32-bit ones). Older U-Boot versions can only boot files in uImage/uInitrd format, so these are often used on older armel systems. Newer U-Boot versions can - besides booting uImages/uInitrds - also boot standard Linux kernels and ramdisk images, but the command syntax to do that is slightly different from that for booting uImages.

For systems using a multiplatform kernel, besides kernel and initial ramdisk a so-called device-tree file (or device-tree blob, «dtb») is needed. It is specific to each supported system and contains a description of the particular hardware. The dtb should be supplied on the device by the firmware, but in practice a newer one often needs to be loaded.

#### 5.1.2 Console configuration

The netboot tarball (Раздел 5.1.3.2), and the installer SD-card images (Раздел 5.1.5) use the (platform-specific) default console that is defined by U-Boot in the «console» variable. In most cases that is a serial console, so on those platforms you by default need a serial console cable to use the installer.

On platforms which also support a video console, you can modify the U-Boot «console» variable accordingly if you would like the installer to start on the video console.

#### 5.1.3 Загрузка по TFTP

Загрузка по сети требует наличия сетевого подключения и сетевого загрузочного сервера TFTP (и, вероятно, также серверов DHCP, RARP или BOOTP для автоматической настройки по сети).

Настройка сервера для обеспечения загрузки из сети описана в Раздел 4.3.

##### 5.1.3.1 Загрузка по TFTP из U-Boot

Загрузка по сети в системах, использующих микропрограмму U-Boot, состоит из трёх шагов: а) настройка сети, б) загрузка образов (ядро/начальный ramdisk/dtb) в память и в) исполнение ранее загруженного кода.

Во-первых, нужно настроить сеть, или автоматически по DHCP командой

```
setenv autoload no
dhcp
```

или вручную указав настройки в переменных окружения

```
setenv ipaddr <ip address of the client>
setenv netmask <netmask>
setenv serverip <ip address of the tftp server>
setenv dnsip <ip address of the nameserver>
setenv gatewayip <ip address of the default gateway>
```

Если хотите, то можно сделать настройки постоянными с помощью команды

```
saveenv
```

После этого нужно загрузить образы (ядро/начальный ramdisk/dtb) в память. Это выполняется с помощью команды `tftpboot`, которой указывается адрес по которому требуется сохранить образ в память. К сожалению, в разных устройствах карты памяти сильно различаются, поэтому нет общего правила какие адреса следует использовать.

On some systems, U-Boot predefines a set of environment variables with suitable load addresses: `kernel_addr_r`, `ramdisk_addr_r` and `fdt_addr_r`. You can check whether they are defined by running

```
printenv kernel_addr_r ramdisk_addr_r fdt_addr_r
```

If they are not defined, you have to check your system's documentation for appropriate values and set them manually. For systems based on Allwinner SunXi SOCs (e.g. the Allwinner A10, architecture name «sun4i» or the Allwinner A20, architecture name «sun7i»), you can e.g. use the following values:

```
setenv kernel_addr_r 0x46000000
setenv fdt_addr_r 0x47000000
setenv ramdisk_addr_r 0x48000000
```

После определения адресов загрузки вы можете загрузить образы в память с ранее настроенного сервера `tftp` с помощью команд

```
tftpboot ${kernel_addr_r} <имя файла образа ядра>
tftpboot ${fdt_addr_r} <имя файла dtb>
tftpboot ${ramdisk_addr_r} <имя файла начального ramdisk>
```

Третьим шагом является настройка командной строки ядра и запуск загруженного кода. U-boot передаёт содержимое переменной окружения «bootargs» ядру как командную строку, поэтому все параметры ядра и программы установки — консольное устройство (смотрите Раздел 5.3.1) или ответы на вопросы о настройке (смотрите Раздел 5.3.2 and Приложение B) — можно указать с помощью команды вида

```
setenv bootargs console=ttyS0,115200 rootwait panic=10
```

Точная команда для выполнения ранее загруженного кода зависит от используемого формата образа. Для `uImage/uInitrd`, команда будет

```
bootm ${kernel_addr_r} ${ramdisk_addr_r} ${fdt_addr_r}
```

, а для родных образов Linux

```
bootz ${kernel_addr_r} ${ramdisk_addr_r}:${filesize} ${fdt_addr_r}
```

Замечание: При загрузке стандартных образов linux важно загрузить образ начального ramdisk после ядра и dtb так как U-Boot устанавливает переменную `filesize` равной размеру последнего загруженного файла, а для правильной работы команды `bootz` требуется размер образа ramdisk. При загрузке специального ядра устройства, то есть ядра без дерева устройств, просто не указывайте параметр `${fdt_addr_r}`.

### 5.1.3.2 Pre-built netboot tarball

Debian provides a pre-built tarball ([.../images/netboot/netboot.tar.gz](#)) that can simply be unpacked on your tftp server and contains all files necessary for netbooting. It also includes a boot script that automates all steps to load the installer. Modern U-Boot versions contain a tftp autoboot feature that becomes active if there is no bootable local storage device (MMC/SD, USB, IDE/SATA/SCSI) and then loads this boot script from the tftp server. Prerequisite for using this feature is that you have a dhcp server in your network which provides the client with the address of the tftp server.

If you would like to trigger the tftp autoboot feature from the U-Boot commandline, you can use the following command:

```
run bootcmd_dhcp
```

To manually load the bootscript provided by the tarball, you can alternatively issue the following commands at the U-Boot prompt:

```
setenv autoload no
dhcp
tftpboot ${scriptaddr} /debian-installer/armhf/tftpboot.scr
source ${scriptaddr}
```

### 5.1.4 Загрузка с карты памяти USB из U-Boot

Многие современные версии U-Boot поддерживают USB и позволяют выполнять загрузку накопителей USB, например карт памяти USB (флешек). К сожалению, точные шаги для выполнения данной процедуры для разных устройств различаются.

В u-Boot v2014.10 появился общий механизм обработки командной строки и autoboot. Это позволяет собирать единые загрузочные образы, которые работают на любой системе, поддерживающей данный механизм. `debian-installer` поддерживает установку с носителей USB для таких систем, но, к сожалению, пока не все платформы переделаны под этот новый механизм.

Чтобы создать загрузочный носитель USB для установки Debian распакуйте tar-архив `hd-media` (смотрите Раздел 4.2.1) на носитель USB с файловой системой, поддерживаемой версией U-Boot вашего устройства. Современные версии U-Boot, обычно, работают с FAT16 / FAT32 / ext2 / ext3 / ext4. После этого скопируйте на носитель образ ISO первого установочного CD или DVD Debian.

Механизм autoboot в современных версиях U-Boot работает также как BIOS ПК, т. е. по списку возможных загрузочных устройств ищется корректный загрузочный образ и запускается первый найденный. Если операционная система не установлена, то подключение носителя USB и включение питания должно привести к запуску программы установки. Также вы можете запустить процесс загрузки с USB в любой момент из приглашения U-Boot с помощью команды `«run usb_boot»`.

Единственная проблема, которая может возникнуть при загрузке с USB при использовании последовательной консоли — несовпадение скоростей передачи консоли. Если переменная консоли определена в U-Boot, то загрузочный сценарий `debian-installer` автоматически передаёт её ядру для настройки первичной консоли устройства и, если уместно, скорость консоли. К сожалению, на разных платформах используется разная переменная консоли — у некоторых она содержит скорость (`«console=ttyS0,115200»`), а у других имеется только устройство (`«console=ttyS0»`). В последнем случае из-за несовпадения скоростей консоли по умолчанию у U-Boot и ядра вывод данных искажается. Современные версии U-Boot часто используют скорость 115200 бод, а ядро по-прежнему работает на традиционных 9600 бод. Если такое произошло, то вы должны задать переменную консоли вручную с правильной скоростью и затем запустить программу установки с помощью команды `«run usb_boot»`.

### 5.1.5 Using pre-built SD-card images with the installer

For a number of systems, Debian provides SD card images that contain both U-Boot and the `debian-installer`. These images are provided in two variants - one for downloading the software packages over the network (available at [.../images/netboot/SD-card-images/](#)) and one for offline installations using a Debian CD/DVD (available at [.../images/hd-media/SD-card-images/](#)). To save space and network bandwidth, the images consist of two parts - a system-dependent part named `«firmware.<system-type>.img.gz»`, and a system-independent part named `«partition.img.gz»`.

To create a complete image from the two parts on Linux systems, you can use `zcat` as follows:

```
zcat firmware.<system-type>.img.gz partition.img.gz > complete_image.img
```

On Windows systems, you have to first decompress the two parts separately, which can be done e.g. by using 7-Zip, and then concatenate the decompressed parts together by running the command

```
copy /b firmware.<system-type>.img + partition.img complete_image.img
```

in a Windows CMD.exe window.

Write the resulting image onto an SD card, e.g. by running the following command on a Linux system:

```
cat complete_image.img > /dev/SD_CARD_DEVICE
```

After plugging the SD card into the target system and powering the system up, the installer is loaded from the SD card. If you use the `hd-media` variant for offline installations, you must provide the installer with access to the first Debian CD/DVD on a separate medium, which can e.g. be a CD/DVD ISO image on a USB stick.

When you come to the partitioning step in the installer (see Раздел 6.3.3), you can delete or replace any previous partitions on the card. Once the installer is started, it runs completely in the system's main memory and does not need to access the SD card anymore, so you can use the full card for installing Debian. The easiest way to create a proper partition layout on the SD card is to let the installer automatically create one for you (see Раздел 6.3.3.2).

## 5.2 Лёгкость управления

Some users may need specific support because of e.g. some visual impairment. Most accessibility features have to be enabled manually. Some boot parameters can be appended to enable accessibility features. Note that on most

architectures the boot loader interprets your keyboard as a QWERTY keyboard.

### 5.2.1 Пользовательский интерфейс программы установки

Программа установки Debian поддерживает несколько пользовательских интерфейсов для задания вопросов с различной степенью удобства управления: в частности, в **text** используется простой текст, в **newt** используются текстовые псевдографические окна. Выбор можно сделать в приглашении к загрузке, смотрите документацию на **DEBIAN\_FRONTEND** в Раздел 5.3.2.

### 5.2.2 Встраиваемые устройства

Некоторые устройства облегчения управления представляют собой платы, которые устанавливаются внутрь компьютера и читают текст напрямую из видеопамяти. Чтобы они смогли работать, нужно выключить использование фрейм-буфера с помощью загрузочного параметра **fb=false**. Однако, это сократит количество доступных языков установки.

### 5.2.3 Тема с высокой контрастностью

Для пользователей с ослабленным зрением в программе установки предусмотрена высоко-контрастная цветовая тема, которая делает её более читаемой. Чтобы её использовать, добавьте загрузочный параметр **theme=dark**.

### 5.2.4 Zoom

For users with low vision, the graphical installer has a very basic zoom support: the Control+ and Control- shortcuts increase and decrease the font size.

### 5.2.5 Expert install, rescue mode, automated install

Expert, Rescue, and Automated installation choices are also available with accessibility support. To access them, one has to first enter the «Advanced options» submenu from the boot menu by typing **a**. When using a BIOS system (the boot menu will have beeped only once), this has to be followed by **Enter**. Then, to enable speech synthesis, **s** can optionally be pressed (followed again by **Enter** when using a BIOS system). From there, various shortcuts can be used: **x** for expert installation, **r** for rescue mode, or **a** for automated installation. When using a BIOS system, they must be followed by **Enter**.

The automated install choice allows to install Debian completely automatically by using preseeding, whose source can be entered after accessibility features get started. Preseeding itself is documented in Приложение B.

### 5.2.6 Специальные возможности установленной системы

Документация по специальным возможностям установленной системы доступна на [вики-странице специальных возможностей Debian](#).

## 5.3 Параметры загрузки

Параметры загрузки — это параметры ядра Linux, которые обычно используются для того, чтобы указать присутствующие периферийные устройства. В большинстве случаев ядро может определить периферийные устройства автоматически. Однако, иногда вы должны немного помочь ядру.

Если вы в первый раз загружаете систему, попробуйте параметры загрузки по умолчанию (то есть не пытайтесь их изменять) и посмотрите, корректно ли всё работает. Скорее всего, так оно и будет. Если нет, вы сможете позже перезагрузиться и поискать специальные параметры, которые помогут системе определить аппаратуру вашего компьютера.

Информацию о многих параметрах загрузки можно найти в [Linux BootPrompt HOWTO](#), где можно найти советы и по нераспознанному оборудованию. В этом разделе содержатся только краткие сведения о наиболее важных параметрах. Некоторые распространённые ошибки включены ниже в Раздел 5.4.

### 5.3.1 Загрузочная консоль

Если вы загружаетесь через консоль на последовательном порту, обычно ядро обнаруживает это автоматически. Если к компьютеру подключены видеокарта (фрейм-буфер) и клавиатура, а вы хотите загрузиться через консоль на последовательном порту, вам надо передать ядру параметр **console=устройство**, где *устройство* — имя устройства последовательного порта, обычно `ttys0`.

Вам может потребоваться указать параметры последовательного порта, такие как скорость и чётность, например **console=ttys0, 9600n8**; другие возможные скорости: 57600 или 115200. Проверьте, что указали этот параметр после «---», так как он копируется в настройку системного загрузчика (bootloader) устанавливаемой системы (если это поддерживается программой установки для системного загрузчика).

In order to ensure the terminal type used by the installer matches your terminal emulator, the parameter **TERM=type** can be added. Note that the installer only supports the following terminal types: `linux`, `bterm`, `ansi`, `vt102` and `dumb`. The default for serial console in `debian-installer` is `vt102`. If you are using an IPMI console, or a virtualization tool which does not provide conversion into such terminals types itself, e.g. QEMU/KVM, you can start it inside a **screen** session. That will indeed perform translation into the `screen` terminal type, which is very close to `vt102`.

### 5.3.2 Параметры программы установки Debian

The installation system recognizes a few additional boot parameters<sup>1</sup> which may be useful.

Некоторые параметры имеют «сокращённую форму записи», что помогает избежать ограничений параметров командной строки ядра и облегчает ввод параметров. Если у параметра есть сокращённая форма, то она даётся в скобках после (обычной) длинной формы. В примерах руководства также используется сокращённая форма.

**debconf/priority (priority)** Этот параметр задаёт низший приоритет отображаемых сообщений.

Значение по умолчанию равно **priority=high**. Это означает, что будут показаны сообщения с высоким и с критическим приоритетами, а сообщения со средним и низким приоритетами будут пропущены. В случае возникновения проблем, программа установки подстраивает значение приоритета.

Если вы добавите **priority=medium** в качестве параметра загрузки, то увидите меню программы установки и получите больший контроль над установкой. При использовании **priority=low** показываются все сообщения (эквивалентно методу загрузки *expert*). Если указать **priority=critical**, то система установки будет показывать только критические сообщения и попыбует всё сделать правильно самостоятельно.

**DEBIAN\_FRONTEND** Этот параметр загрузки контролирует тип пользовательского интерфейса, используемого программой установки. Возможные на текущий момент значения параметра:

- **DEBIAN\_FRONTEND=noninteractive**
- **DEBIAN\_FRONTEND=text**
- **DEBIAN\_FRONTEND=newt**
- **DEBIAN\_FRONTEND=gtk**

По умолчанию выбран тип **DEBIAN\_FRONTEND=newt**. Для установки через консоль на последовательном порту наиболее предпочтителен **DEBIAN\_FRONTEND=text**. Некоторые специализированные установочные носители могут предлагать ограниченный список интерфейсов, но **newt** и **text** доступны в большинстве из них. На архитектурах, для которых есть поддержка, используется графический интерфейс программы установки **gtk**.

**BOOT\_DEBUG** Присвоение этому загрузочному параметру значения 2 приводит к более подробному протоколированию процесса загрузки программы установки. Значение 3 приводит к запуску интерпретатора командной строки в наиболее важных точках процесса загрузки. (Завершение работы с интерпретатором приводит к продолжению процесса загрузки.)

**BOOT\_DEBUG=0** Значение по умолчанию.

**BOOT\_DEBUG=1** Подробней чем обычно.

**BOOT\_DEBUG=2** Много отладочной информации.

<sup>1</sup> With current kernels (2.6.9 or newer) you can use 32 command line options and 32 environment options. If these numbers are exceeded, the kernel will panic. Also there is a limit of 255 characters for the whole kernel command line, everything above this limit may be silently truncated.

- BOOT\_DEBUG=3** В различных точках процесса загрузки будет вызван интерпретатор командной строки, чтобы разрешить детальную отладку. При выходе из интерпретатора загрузка продолжится.
- INSTALL\_MEDIA\_DEV** Значением этого параметра является путь к устройству, с которого загружается программа установки Debian. Например, **INSTALL\_MEDIA\_DEV=/dev/floppy/0**
- Загрузочная дискета в поисках корневой дискеты сканирует все дискеты и USB устройства хранения. Этим параметром можно заставить её искать корневую дискету только на избранном устройстве.
- log\_host, log\_port** Используется для указания программе установки отправлять сообщения протокола кроме локального файла ещё и удалённой службе syslog на указанный узел и порт. Если не задано, то портом по умолчанию является стандартный 514-й порт syslog.
- lowmem** Используется для указания программе установки выставить для lowmem больший уровень, чем это рассчитывается по умолчанию исходя из количества доступной памяти. Возможные значения 1 и 2. Смотрите также Раздел [6.3.1.1](#).
- noshell** Указывает программе установки не запускать интерактивную оболочку на tty2 и tty3. Полезно при автоматизированной установке, где ограничена физическая безопасность.
- debian-installer/framebuffer (fb)** Чтобы предложить установку на нескольких языках на некоторых архитектурах используется фрейм-буфер ядра. Если работа через фрейм-буфер вызывает проблемы на вашей системе, можно выключить его параметром **fb=false**. Симптомы проблемы — сообщения об ошибках bterm или bogl, пустой экран или замирание на несколько минут в начале установки.
- debian-installer/theme (theme)** Темой определяется вид пользовательского интерфейса программы установки (раскраска, иконки и т.д.). У каждого интерфейса есть свои темы. В настоящий момент для интерфейсов newt и gtk есть только одна тема «dark», которая была специально разработана для людей с ослабленным зрением. Для установки темы загрузитесь с параметром **theme=dark**.
- netcfg/disable\_autoconfig** По умолчанию, `debian-installer` автоматически пытается настроить сеть через IPv6 autoconfiguration и DHCP. Если получение адреса прошло успешно, у вас не будет шанса пересмотреть и изменить полученные настройки. Вы сможете настроить сеть вручную только в случае ошибок при получении параметров автоматически.
- Если в вашей локальной сети есть маршрутизатор IPv6 или DHCP сервер, но вы не хотите их использовать, потому что они, например, даёт неправильные ответы, то можете указать параметр **netcfg/disable\_autoconfig** чтобы отключить настройку сети (v4 и v6) и ввести информацию вручную.
- hw-detect/start\_pcmcia** Установите в **false**, чтобы избежать запуска служб PCMCIA, если это вызывает проблемы. Особенно полезно для некоторых ноутбуков.
- disk-detect/dmraid/enable (dmraid)** Установите в **true**, чтобы включить поддержку Serial ATA RAID (также называемого ATA RAID, BIOS RAID или недо-RAID) дисков в программе установки. Заметим, что эта поддержка является экспериментальной. Дополнительную информацию можно найти на страницах [вики Debian Installer](#).
- preseed/url (url)** Задайте url для загрузки файла с настройками и использования автоматической установки. Смотрите Раздел [4.4](#).
- preseed/file (file)** Укажите путь к файлу для загрузки настроек автоматической установки. Смотрите Раздел [4.4](#).
- preseed/interactive** Установка в **true** включает отображение всех вопросов, даже для которых есть автоматизированный ответ. Может оказаться полезным при отладке или тестировании файла ответов. Заметим, что это не действует на параметры, указанные в приглашении к загрузке, для работы с ними используется специальный синтаксис. Смотрите Раздел [B.5.2](#).
- auto-install/enable (auto)** Отложить вопросы, которые обычно задаются перед автоматической установкой, которые можно задать после настройки сети. Подробнее об автоматизации установки смотрите в Раздел [B.2.3](#).
- finish-install/keep-consoles** При установке с консоли на последовательном порту обычные виртуальные консоли (VT1-VT6) выключаются в `/etc/inittab`. Если установить в значение **true**, то этого не случится.

**cdrom-detect/eject** По умолчанию перед перезагрузкой, `debian-installer` автоматически извлекает компакт-диск, который использовался при установке. Это делать необязательно, если система не загружается с компакт-диска автоматически. В некоторых случаях это даже нежелательно, например, если устройство чтения не может самостоятельно загрузить компакт-диск и пользователь не может сделать это вручную. Многие устройства со слотовой загрузкой, `slim-line` и `caddy` не способны загружать носитель автоматически.

Чтобы запретить автоматическое извлечение задайте значение **false**, но проверьте, что система не загружается автоматически с компакт-диска после начальной установки.

**base-installer/install-recommends (recommends)** Если назначить данному параметру значение **false**, то это отключит в системе управления пакетами установку «Рекомендуемых» пакетов, как во время установки так и в установленной системе.

Заметим, что этот параметр позволяет получить более отточенную систему, но это также может привести к отсутствию некоторых возможностей, которые вы рассчитывали получить. Для получения дополнительных возможностей вы можете установить некоторые рекомендуемые пакеты вручную. Поэтому этот параметр должен использоваться только очень опытными пользователями.

**debian-installer/allow\_unauthenticated** По умолчанию программа установки требует, чтобы репозитории аутентифицировались с помощью известного ключа `gpg`. Установка в **true** выключает данную аутентификацию. **Предупреждение: это небезопасно, рекомендуется не делать этого.**

**rescue/enable** Установка в **true** активирует режим восстановления, вместо обычного режима установки. Смотрите Раздел 8.7.

### 5.3.3 Использование параметров загрузки для ответов на вопросы

За некоторым исключением, в приглашении к загрузке можно задать любые значения ответов на вопросы установки, хотя это полезно только в специфических случаях. Описание как это делать можно найти в Раздел 5.2.2. Некоторые специальные примеры даны ниже.

**debian-installer/language (language), debian-installer/country (country), debian-installer/locale (locale)** Есть два способа указать язык, страну и локаль, используемую при установке и в установленной системе.

Первый и самый простой — передать только параметр `locale`. Для языка и страны будет использовано вычисленное значение. Например, вы можете использовать `locale=de_CH` для выбора немецкого языка и Швейцарии в качестве страны (в установленной системе по умолчанию будет использована локаль `de_CH.UTF-8`). Этот способ имеет ограничения по комбинациям языка, страны и локали.

Второй, более гибкий способ — отдельно указать язык и страну. В этом случае `locale` можно задать дополнительно для выбора специфичной локали по умолчанию для установленной системы. Пример: `language=en country=DE locale=en_GB.UTF-8`.

**anna/choose\_modules (modules)** Может использоваться для автоматической загрузки компонент программы установки, которые не загружаются по умолчанию. В качестве примеров дополнительных полезных компонент, можно привести `openssh-client-udeb` (для того, чтобы во время установки можно было использовать команду `scp`) и `ppp-udeb` (настройки PPPoE).

**netcfg/disable\_autoconfig** Установите в **true**, если хотите выключить IPv6 autoconfiguration и DHCP и задать настройки сети статически.

**mirror/protocol (protocol)** По умолчанию, программа установки использует протокол `http` для загрузки файлов с серверов-зеркал Debian, и изменить его на `ftp` невозможно при установке с обычным приоритетом. Установив этот параметр равным **ftp**, вы можете заставить программу установки использовать этот протокол. Заметим, что вы не сможете выбрать сервер-зеркало `ftp` из списка, вам придётся вводить имя вручную.

**tasksel:tasksel/first (tasks)** Может быть использована для выбора задач, которых нет в списке выбора в программе установки, например задача `kde-desktop`. Подробнее смотрите в Раздел 6.3.5.2.



### 5.3.4 Передача параметров модулям ядра

Если драйверы встроены в ядро, вы можете передать им параметры как это описано в документации к ядру. Если же драйверы собраны как модули и так как модули ядра загружаются немного по-другому чем при загрузке установленной системы, невозможно передать параметры модулям обычным способом. Вместо этого вам нужно использовать специальный синтаксис, распознаваемый программой установки, которая сохраняет эти параметры в нужные конфигурационные файлы, для того чтобы использовать их при загрузке модуля. Параметры также будут автоматически включены в конфигурацию установленной системы.

Заметим, что теперь редко требуется указывать параметры модулям. В большинстве случаев ядро способно само определить имеющееся оборудование в системе и настроить его правильно. Однако иногда всё ещё может потребоваться установка параметров вручную.

Синтаксис установки параметров модулей:

```
имя_модуля.имя_параметра=значение
```

Если требуется передать несколько параметров одному или разным модулям, то выражение повторяют. Например, чтобы указать старым сетевым картам 3Com использовать BNC (коаксиальный) разъём и IRQ 10, вы должны написать:

```
3c509.xcvr=3 3c509.irq=10
```

### 5.3.5 Внесение модулей ядра в чёрный список

Иногда бывает необходимо внести модуль в чёрный список, чтобы предотвратить его автоматическую загрузку ядром и `udev`. Одной из причин для этого может быть то, что этот модуль неправильно работает с вашим оборудованием. Также, иногда ядро имеет два разных драйвера для одного устройства. Это может приводить к неправильной работе устройства при конфликте драйверов, или если сперва загружается неправильный драйвер.

Чтобы внести модуль в чёрный список используйте следующий синтаксис: **`module_name.blacklist=yes`**. Это поместит модуль в чёрный список в файле `/etc/modprobe.d/blacklist.local` на время установки и в установленной системе.

Заметим, что модуль всё ещё может загружаться самой системой установки. Чтобы этого избежать, запустите установку в экспертном режиме и снимите выделение с модуля в списке модулей, отображаемого во время шага обнаружения оборудования.

## 5.4 Диагностика проблем процесса установки

### 5.4.1 Надёжность CD-ROM

Иногда, особенно в старых приводах CD-ROM, программа установки может не загрузиться с CD-ROM. Программа установки также может (даже после успешной загрузки с CD-ROM) не распознать CD-ROM или выдавать ошибки при чтении во время установки.

Такие проблемы могут возникать по разным причинам. Мы можем описать только общие из них и предложить общие решения. Остальное зависит от вас.

Есть несколько простых вещей, которые вы можете попробовать сделать в первую очередь.

- Если с CD-ROM не получается загрузиться, проверьте, что он вставлен правильно и что диск не грязный.
- Если программа установки не смогла определить CD-ROM, попробуйте выполнить пункт меню Поиск и монтирование CD-ROM второй раз. Таким способом решаются некоторые проблемы с DMA на очень старых приводах CD-ROM.

Если это не сработало, попробуйте советы данные ниже. Хотя большинство, но не все, описываемые предложения работают и для CD-ROM и для DVD, для простоты мы будем использовать термин CD-ROM.

Если вы не можете заставить программу установки работать с CD-ROM, попробуйте, если возможно, другой метод установки.

### 5.4.1.1 Общие причины

- Некоторые старые приводы CD-ROM не поддерживают чтение с дисков, которые были записаны на высоких скоростях в современных устройствах записи CD.
- Некоторые очень старые приводы CD-ROM работают неправильно, если включён «прямой доступ к памяти» (DMA).

### 5.4.1.2 Как изучать и может быть решить проблему

Если с CD-ROM не удаётся загрузиться, попробуйте способы указанные ниже.

- Проверьте, что в BIOS включена загрузка с CD-ROM (возможно, очень старые машины этого не умеют), и что привод CD-ROM поддерживает используемый тип носителя.
- Если вы скачивали образ iso, проверьте, что md5sum этого образа совпадает с суммой указанной в файле MD5SUMS, который должен лежать там же, где вы брали образ.

```
$ md5sum debian-testing-i386-netinst.iso
a20391b12f7ff22ef705cee4059c6b92  debian-testing-i386-netinst.iso
```

Далее проверьте, что md5sum записанного CD-ROM также совпадает. Следующая команда должна сработать. Здесь считается сумма прочитанных с CD-ROM байт, составляющих образ.

```
$ dd if=/dev/cdrom | \
> head -c 'stat --format=%s debian-testing-i386-netinst.iso' | \
> md5sum
a20391b12f7ff22ef705cee4059c6b92 -
262668+0 records in
262668+0 records out
134486016 bytes (134 MB) copied, 97.474 seconds, 1.4 MB/s
```

Если после успешной загрузки программы установки не определяется CD-ROM, то иногда помогает просто попробовать ещё раз. Если у вас более одного привода CD-ROM, попробуйте загрузиться с другого CD-ROM. Если это не помогает или если CD-ROM определяется, но при чтении возникают ошибки, попробуйте выполнить предложения, данные ниже. Для этого потребуются азы работы в Linux. Перед тем как выполнить любую команду, сначала вы должны переключиться во вторую виртуальную консоль (VT2) и активировать оболочку командной строки.

- Переключитесь в VT4 или посмотрите файл `/var/log/syslog` (в качестве редактора используется **nano**), чтобы найти сообщения о специфических ошибках. После этого также проверьте вывод команды **dmesg**.
- В выводе **dmesg** проверьте, что привод CD-ROM был найден. Вы должны увидеть что-то типа этого (строки могут быть в другой последовательности):

```
ata1.00: ATAPI: MATSHITADVD-RAM UJ-822S, 1.61, max UDMA/33
ata1.00: configured for UDMA/33
scsi 0:0:0:0: CD-ROM MATSHITA DVD-RAM UJ-822S 1.61 PQ: 0 ANSI: 5
sr0: scsi3-mmc drive: 24x/24x writer dvd-ram cd/rw xa/form2 cdda tray
cdrom: Uniform CD-ROM driver Revision: 3.20
```

Если вы не нашли подобных строк, это случай, когда контроллер, к которому подключён CD-ROM, не распознан или не поддерживается ядром вообще. Если вы знаете, какой драйвер требуется для привода, то можете попытаться загрузить его вручную с помощью команды **modprobe**.

- Проверьте, что есть соответствующий файл устройства для привода CD-ROM в каталоге `/dev/`. Для примера выше, должен существовать файл `/dev/sr0`. Также может быть `/dev/cdroms/cdrom0`.
- С помощью команды **mount** проверьте, что CD-ROM уже смонтировать; если нет, попытайтесь смонтировать вручную:

```
$ mount /dev/hdc /cdrom
```

Проверьте, не появилось ли ошибок после этой команды.

- Проверьте, включён ли DMA:

```
$ cd /proc/ide/hdc
$ grep using_dma settings
using_dma      1      0      1      rw
```

«1» означает включён. Если включён, попробуйте выключить его:

```
$ echo -n "using_dma:0" >settings
```

Удостоверьтесь, что находитесь в каталоге для устройства, соответствующего приводу CD-ROM.

- Если возникли проблемы во время установки, попробуйте проверить целостность CD-ROM с помощью пункта внизу главного меню программы установки. Этим также можно провести общую проверку надёжности чтения CD-ROM.

### 5.4.2 Настройка загрузки

Если у вас возникли проблемы и ядро падает во время загрузки, не распознает имеющуюся периферию или неправильно распознаются диски, сперва проверьте параметры загрузки, как это обсуждается в Раздел 5.3.

В некоторых случаях, неисправность может быть вызвана отсутствием микропрограммы устройства (смотрите Раздел 2.2 и Раздел 6.4).

### 5.4.3 Интерпретация загрузочных сообщений ядра

Во время загрузки вы можете увидеть много сообщений вида `can't find что-то` или `что-то not present, can't initialize что-то` или даже `this driver release depends on чего-то`. Большинство этих сообщений безвредны. Вы видите их потому что устанавливаемое ядро собрано так, чтобы запускаться на компьютерах с самыми различными периферийными устройствами. Очевидно, ни один компьютер не может содержать все доступные устройства, так что операционная система может немного пожаловаться во время поиска периферии, которой у вас нет. Также, вы можете заметить, что система иногда замирает. Это случается из-за ожидания ответа от устройства, которого нет в вашей системе. Если вы посчитаете, что загрузка системы занимает недопустимо долгое время, вы можете позже создать собственное ядро (смотрите Раздел 8.6).

### 5.4.4 Сообщение о проблемах при установке

Если вы прошли фазу начальной загрузки, но не смогли завершить установку, полезно войти в меню Сохранение файлов журнала с отладочной информацией. Из этого меню можно скопировать системные журналы ошибок и информацию о конфигурации из программы установки на дискету или загрузить через веб браузер. Эта информация поможет понять, что было не так и как это исправить. Если вы отправляете отчёт об ошибках, то можете приложить эту информацию к отчёту.

Другие сообщения, имеющие отношение к установке, можно найти в `/var/log/` во время установки, и в `/var/log/installer/` после того, как компьютер загрузит установленную систему.

### 5.4.5 Отправка отчётов об ошибках установки

Если у вас возникли проблемы, отправьте отчёт об установке. Также, мы приветствуем отправку отчётов даже в случае успешной установки, чтобы получить побольше информации о разных аппаратных конфигурациях.

Заметим, что отчёт об установке будет опубликован в системе отслеживания ошибок Debian (BTS) и переслан в открытый список рассылки. Проверьте, что вы используете адрес электронной почты, который не жалко опубликовывать.

If you have a working Debian system, the easiest way to send an installation report is to install the `installation-report` and `reportbug` packages (`apt install installation-report reportbug`), configure `reportbug` as explained in Раздел 8.5.2, and run the command `reportbug installation-reports`.

Или же используйте этот шаблон для заполнения отчёта об установке и отправьте отчёт как сообщение об ошибке псевдо-пакету `installation-reports` на адрес [submit@bugs.debian.org](mailto:submit@bugs.debian.org).

```
Package: installation-reports
```

```
Boot method: <Как запускалась программа установки? с CD? с дискет? из сети?>
```

```
Image version: <Указать полный url, где брался образ>
Date: <Дата и время установки>

Machine: <Описание машины (например, IBM Thinkpad R32)>
Processor:
Memory:
Partitions: < вывод результата команды df -Tl; лучше таблицу разделов в ←
             необработанном (raw) виде>

Вывод результата команд lspci -knn и lspci -nn:

Base System Installation Checklist:
[O] = OK, [E] = Ошибка (описать подробности ниже), [ ] = не пробовал

Initial boot:           [ ]
Detect network card:    [ ]
Configure network:      [ ]
Detect CD:              [ ]
Load installer modules: [ ]
Detect hard drives:     [ ]
Partition hard drives:  [ ]
Install base system:    [ ]
Clock/timezone setup:   [ ]
User/password setup:    [ ]
Install tasks:          [ ]
Install boot loader:    [ ]
Overall install:        [ ]

Comments/Problems:

<Описание установки, комментарии, предложения возникшие в процессе.>
```

В отчёте об ошибках опишите, в чём состоит проблема, включая последние увиденные сообщения ядра в случае подвисания ядра. Опишите шаги, которые вы делали, чтобы привести систему в проблемное состояние.

## Глава 6

# Использование программы установки Debian

### 6.1 Как работает программа установки

Программа установки Debian состоит из нескольких специализированных компонент, предназначенных для выполнения определённых задач установки. Каждая компонента выполняет свою задачу, задавая пользователю вопросы, если это необходимо для выполнения работы. Каждому вопросу назначен приоритет, и можно задать приоритет задаваемых вопросов при старте программы установки.

Когда выполняется установка по умолчанию, задаются только самые важные (высокоприоритетные) вопросы. Результатом будет полностью автоматизированный процесс установки с небольшим вмешательством пользователя. Компоненты автоматически запускаются в определённой последовательности; какая компонента будет запущена, зависит, в основном, от выбранного метода установки и от имеющегося оборудования. Программа установки будет использовать значения по умолчанию для вопросов, которые не задавались.

При возникновении проблемы, пользователь увидит экран с ошибкой и, возможно, меню программы установки предложит выбрать какое-нибудь альтернативное действие. Если же проблем нет, пользователь никогда не увидит меню программы установки, а просто будет отвечать на вопросы каждой компоненты. Извещения о серьёзных ошибках установлен приоритет «критический», так что пользователь всегда будет оповещён.

Некоторые параметры по умолчанию, используемые программой установки, можно изменить передав как значения параметров загрузки при старте `debian-installer`. Если, например, вы хотите настраивать сеть вручную (по умолчанию используется IPv6 autoconfiguration и DHCP, если получится), то можете добавить параметр загрузки `netcfg/disable_autoconfig=true`. Возможные параметры можно посмотреть в Раздел 5.3.2.

Опытным пользователям, возможно, будет более удобен интерфейс управления через меню, где каждый шаг контролируется пользователем, а не выполняется автоматически программой установки. Для использования программы установки в ручном режиме управляемом через меню, добавьте параметр загрузки `priority=medium`.

Если ваше оборудование требует передачи параметров модулям ядра при их установке, нужно запустить программу установки в режиме «эксперта». Это можно сделать либо введя команду `expert` для запуска программы установки, либо передав параметр загрузки `priority=low`. Режим эксперта даёт вам полный контроль над `debian-installer`.

Для данной архитектуры программа установки поддерживает только текстовый пользовательский интерфейс. Графический интерфейс пока недоступен.

В текстовом окружении невозможно управление с помощью мыши. Вот кнопки, с помощью которых вы можете перемещаться внутри различных диалогов. Клавиша **Tab** или **правая** стрелка перемещают указатель «вперёд», а Shift-Tab или **левая** стрелка — «назад» по отображаемым кнопкам или полям ввода. Клавишами стрелок **вверх** и **вниз** выбираются различные пункты в прокручиваемом списке, а также прокручивается сам список. Кроме этого, в длинных списках вы можете нажать на клавишу с буквой, чтобы список прокрутился прямо к пункту, начинающемуся на эту букву, а также использовать клавиши **Pg-Up** и **Pg-Down** для прокручивания списка в секциях. Клавиша **пробел** служит для выбора пунктов типа флажок. Для подтверждения выбора используйте клавишу **Enter**.

По некоторым окнам есть дополнительная справочная информация. Если она доступна, то об этом будет сообщено в нижней строке экрана надписью, что справочная информация доступна по нажатию клавиши **F1**.

Сообщения об ошибках и протокол выполнения выводятся на четвёртой консоли. Вы можете получить доступ к этой консоли, нажав Левый Alt-F4 (удерживайте клавишу **Alt** нажатой во время нажатия функциональной клавиши **F4**); чтобы вернуться к основному процессу установки нажмите Левый Alt-F1.

Также эти сообщения можно найти в файле `/var/log/syslog`. После установки этот файл копируется в файл `/var/log/installer/syslog` на новой системе. Остальные сообщения об установке можно найти в каталоге `/var/log/` во время установки и в `/var/log/installer/` после того, как компьютер перезагрузится в установленную систему.

### 6.1.1 Использование программы установки с графическим интерфейсом

Графическая версия программы установки работает практически также как и текстовая, и поэтому это руководство можно использовать для выполнения процесса установки.

Если вы предпочитаете использовать клавиатуру вместо мыши, есть две вещи, которые необходимо знать. Чтобы раскрыть свёрнутый список (например, выбор стран и континентов), вы можете использовать клавиши **+** и **-**. Если в вопросе можно выбрать более одного значения (например, выбор задач), во-первых, вам нужно перейти на кнопку `Continue` после окончания выбора; нажимая `enter` для переключения выбора, не активировать `Continue`.

Если есть дополнительная справочная информация, то появляется кнопка Справка. Справочная информация доступна по нажатию кнопки или клавиши **F1**.

Чтобы перейти на другую консоль, вам также нужно использовать клавишу **Ctrl**, как и в X Window System. Например, чтобы перейти на VT2 (первая оболочка командной строки для отладки), воспользуйтесь: `Ctrl`-левый `Alt`-`F2`. Сама программа установки с графическим интерфейсом работает на VT5, так что вы можете использовать левый `Alt`-`F5` для обратного переключения.

## 6.2 Описание компонент

Ниже представлен список компонент программы установки с кратким описанием назначения каждой компоненты. Детали об использовании каждой компоненты можно найти в Раздел 6.3.

**main-menu** Показывает список компонент пользователю во время работы программы установки и запускает компоненту, когда она выбрана. Вопросам `main-menu` установлен средний (`medium`) приоритет, так что если вы изменили приоритет на высокий или критический (по умолчанию высокий), вы не увидите меню. С другой стороны, если возникает ошибка, которая требует вмешательства пользователя, приоритет вопросов может быть временно понижен, чтобы позволить решить проблему, и в этом случае меню может появиться.

Вы можете добраться до главного меню, многократно нажав кнопку `Go Back`, чтобы пройти весь путь обратно от выполняемой в данный момент компоненты.

**localechooser** Позволяет пользователю указать параметры локализации процесса установки и устанавливаемой системы: язык, страну и локали. Программа установки будет отображать сообщения на выбранном языке, если есть перевод на этот язык. Если перевода сообщения нет, то будет показано сообщение на английском языке.

**console-setup** Показывает список клавиатур, из которого пользователь может выбрать подходящую модель.

**hw-detect** Автоматически определяет большую часть аппаратуры системы, включая сетевые карты, приводы дисков и PCMCIA.

**cdrom-detect** Ищет и монтирует компакт-диск с программой установки Debian.

**netcfg** Настраивает сетевые подключения компьютера, чтобы он мог получить доступ к Интернет.

**iso-scan** Ищет ISO-образы (файлы `.iso files`) на жёстких дисках.

**choose-mirror** Предоставляет список серверов-зеркал Debian. Пользователь может выбрать источник, из которого будут устанавливаться пакеты.

**cdrom-checker** Проверяет целостность данных на компакт-диске. Таким образом пользователь может убедиться, что установочный компакт-диск не повреждён.

- lowmem** Пытается распознать системы с малым объёмом памяти и затем выполняет различные действия, чтобы удалить ненужные части `debian-installer` из памяти (ценой потери некоторых возможностей).
- anna** Anna Не Настоящий АРТ. Устанавливает пакеты, которые были получены с выбранного сервера-зеркала или компакт-диска.
- user-setup** Настраивает пароль суперпользователя и добавляет обычного пользователя.
- clock-setup** Обновляет системные часы и определяет установлены часы по Гринвичу (UTC) или нет.
- tzsetup** Выбирает часовой пояс, основываясь на местоположении указанном ранее.
- partman** Позволяет пользователю разметить подключённые к системе диски, создать файловые системы на выбранных разделах и назначить им точки монтирования. Также обладает интересными возможностями, например, может работать в полностью автоматическом режиме и поддерживает LVM. Это предпочтительная утилита для работы с разделами в Debian.
- partitioner** Позволяет пользователю разметить диски, подключённые к системе. Программа разметки будет соответствовать архитектуре вашего компьютера.
- partconf** Отображает список разделов и создаёт файловые системы на выбранных разделах, согласно пользовательским указаниям.
- lvmcfg** Помогает пользователю с настройкой LVM (менеджера логических томов).
- mdcfg** Позволяет пользователю установить программный RAID (Redundant Array of Inexpensive Disks). Программный RAID обычно лучше, чем дешёвые IDE (псевдо-аппаратные) RAID-контроллеры в новых материнских платах.
- base-installer** Устанавливает самый минимальный набор пакетов, который позволит компьютеру работать под Debian GNU/Linux после перезагрузки.
- apt-setup** Настраивает apt, по большей части автоматически, основываясь на том, с какого носителя была запущена программа установки.
- pkgsel** Использует `tasksel` для выбора и установки дополнительного ПО.
- os-prober** Определяет уже установленные на компьютере операционные системы и передаёт эту информацию в `bootloader-installer`, который может предложить вам добавить найденные операционные системы в стартовое меню системного загрузчика. Таким образом пользователь может легко выбрать операционную систему для запуска в момент загрузки.
- bootloader-installer** Устанавливает на жёсткий диск программу системный загрузчик, которая необходима для того, чтобы компьютер запустился под Linux без использования дискеты или компакт-диска. Многие системные загрузчики позволяют пользователю выбрать загрузку альтернативной операционной системы при старте компьютера.
- shell** Позволяет пользователю вызвать интерпретатор командной строки из меню или на второй консоли.
- save-logs** Предоставляет пользователю возможность записать информацию на дискету, через сеть, на жёсткий диск или другой носитель в случае возникновения проблем, чтобы позже отправить разработчикам Debian точный отчёт о проблеме в программе установки.

### 6.3 Применение отдельных компонент

В этом разделе мы подробно опишем каждую компоненту программы установки. Компоненты собраны в понятные пользователю группы и расположены в порядке использования во время установки. Обратите внимание, что в зависимости от метода установки и вашей аппаратуры некоторые модули могут не использоваться.

### 6.3.1 Настройка программы установки Debian и оборудования

Давайте представим, что программа установки Debian загрузилась и вы видите её первый экран. В этот момент возможности `debian-installer` всё ещё немного ограничены. Она многого не знает о вашем оборудовании, предпочитаемом языке или даже о задачах, которые она должна выполнить. Не беспокойтесь. Поскольку `debian-installer` немножко умная, она может автоматически определить ваше оборудование, найти свои остальные компоненты и обновить себя до очень умелой системы установки. Однако, вы должны помочь `debian-installer` некоторой информацией, которую она не сможет определить автоматически (например, выбрать предпочитаемый вами язык, раскладку клавиатуры или необходимый сетевой сервер-зеркало).

Заметим, что `debian-installer` выполняет *определение оборудования* несколько раз во время установки. На первом этапе определяется оборудование, необходимое для загрузки компонент программы установки (т.е. ваш CD-ROM или сетевая карта). Поскольку не все драйверы могут быть доступны во время этого первого запуска, определение оборудования надо будет повторить позже по ходу установки.

Во время определения оборудования `debian-installer` проверяет необходимость загрузки микропрограмм в драйверы устройств. Если микропрограмма требуется, но недоступна, выводится диалоговое окно, которое позволяет загрузить отсутствующую микропрограмму со сменного носителя. Подробнее см. Раздел 6.4.

#### 6.3.1.1 Определение доступной памяти / режим с малым потреблением памяти

Одну из первых вещей, которую делает `debian-installer`, это проверка размера доступной памяти. Если размер ограничен, то эта компонента делает некоторые изменения процесса установки, которые, будем надеяться, позволят установить Debian GNU/Linux на вашу машину.

Первое, что делается для уменьшения потребления памяти программой установки — это выключение переводов в интерфейсе, то есть установка будет производиться на английском языке. Конечно, вы сможете выполнить локализацию системы после завершения установки.

Если этого окажется недостаточно, программа установки дальше станет уменьшать потребление памяти загружая только компоненты, которые необходимы для выполнения базовой установки. Это уменьшает функциональность системы установки. Вам будет предоставлена возможность загрузить дополнительные компоненты вручную, но вы должны понимать, что каждая выбранная вами компонента использует память, и это может привести к неудачному завершению установки.

При работе программы установки в режиме с малым потреблением памяти рекомендуется создать относительно большой раздел подкачки (64–128МБ). Раздел подкачки будет использован в качестве виртуальной памяти, и это увеличит количество доступной памяти системы. Программа установки активирует раздел подкачки сразу как только это станет возможно в процессе установки. Заметим, что интенсивное использование раздела подкачки уменьшит производительность системы и может привести к большой нагрузке на диск.

Несмотря на все старания, система всё равно может повиснуть, могут произойти неожиданные ошибки или процессы могут быть завершены ядром из-за нехватки памяти в системе (вы увидите сообщения «Out of memory» на VT4 и в `syslog`).

Например, были отчёты об установке, что не удавалось создать большую файловую систему в режиме с малым потреблением памяти, если было недостаточно пространства подкачки. Если увеличение размера подкачки не помогает, попробуйте вместо неё создать файловую систему `ext2` (которая является обязательной компонентой программы установки). После установки возможно изменить раздел `ext2` на `ext3`.

Возможно заставить программу установки использовать более высокий уровень для `lowmem` чем значение, рассчитываемое из количества доступной оперативной памяти, с помощью загрузочного параметра «`lowmem`» как это описано в Раздел 5.3.2.

#### 6.3.1.2 Выбор параметров локализации

В большинстве случаев сначала вас попросят указать параметры локализации, которые будут использованы в программе установки и в устанавливаемой системе. В параметры локализации входят язык, расположение и локали.

Выбранный язык будет использован в процессе установки, предлагая переводы различных диалогов если возможно. Если перевод для выбранного языка недоступен, по умолчанию используется английский язык.

Указанное значение географического расположения (в большинстве случаев — страны) будет использовано позднее в процессе установки для выбора правильного часового пояса и подходящего для страны сервера-зеркала Debian. Значения страны и языка будут использованы вместе для настройки локали вашей системы и выборе правильной раскладки клавиатуры.



Сначала вас попросят выбрать предпочитаемый язык. Названия языков написаны на английском(слева) и на самом языке (справа); названия справа также показаны в правильном написании для этого языка. Список отсортирован по английским названиям. Вверху списка есть дополнительный пункт, который вместо языка позволяет выбрать «С» локаль. При выборе «С» локаль установка будет производиться на английском; установленная система не будет иметь поддержку локализации, так как пакет `locales` в данном случае установлен не будет.

Далее вас попросят указать географическое расположение. Если вы выбрали язык, на котором говорят более чем в одной стране<sup>1</sup>, вам будет показан список только этих стран. Чтобы указать страну не из списка, выберите другая (последний пункт). Далее вам будет предложен список континентов; при выборе определённого континента будет показан список стран, которые на нём расположены.

Если с языком связана только одна страна, то будет показан список стран континента или региона, на котором находится страна, и по умолчанию будет выбрана эта страна. Нажмите Вернуться, для выбора стран другого континента.

#### Замечание



Очень важно выбрать страну вашего проживания или нахождения, так как по ней определяется часовой пояс, который будет настроен в установленной системе.

Если вы выберете комбинацию языка и страны, для которой нет локаль и есть несколько локалей для этого языка, то программа установки предложит вам выбрать желаемую локаль по умолчанию для устанавливаемой системы<sup>2</sup>. Во всех других случаях локаль по умолчанию будет выбрана на основе языка и страны.

В любой выбранной локаль по умолчанию, как описано в предыдущем параграфе, будет использована кодировка символов *UTF-8*.

Если вы производите установку с низким приоритетом, то вас попросят выбрать дополнительные локаль, включая так называемые «старые» локаль<sup>3</sup> для их генерации в установленной системе; если вы что-то выберете, то вас попросят выбрать локаль по умолчанию для устанавливаемой системы.

#### 6.3.1.3 Выбор клавиатуры

Клавиатуры часто приспособляют к алфавиту определённого языка. Выберите раскладку, которая подходит к вашей клавиатуре, или выберите наиболее подходящую, если таковой не имеется. После завершения установки вы сможете выбрать раскладку из большого количества вариантов (запустите `kbd-config` с правами суперпользователя после завершения установки).

Выберите нужную клавиатуру и нажмите **Enter**. Для перемещения используйте клавиши стрелок — они находятся в одном и том же месте во всех национальных клавиатурных раскладках, поэтому не зависят от конфигурации клавиатуры.

#### 6.3.1.4 Поиск образа ISO программы установки Debian

Если установка производится с помощью метода *hd-media*, настанет момент, когда вам нужно будет найти и смонтировать ISO-образ программы установки Debian, чтобы получить оставшиеся файлы установки. Для этого существует компонента `iso-scan`.

Сначала `iso-scan` автоматически смонтирует все блочные устройства (например, разделы), на которых есть распознанная файловая система, и последовательно будет искать на них файлы с именами, оканчивающимися на `.iso` (или `.ISO`, это важно). Обратите внимание, ищутся файлы только в корневом каталоге и подкаталогах первого уровня (то есть найдутся файлы `/имя.iso`, `/data/имя.iso`, но не `/data/tmp/имя.iso`). После нахождения iso-образа, `iso-scan` проверит его содержимое чтобы выяснить, является ли образ правильным iso-образом Debian или нет. В первом случае поиск завершается, в последнем `iso-scan` будет искать другой образ.

Если найти ISO-образ с программой установки не удалось, `iso-scan` спросит, хотите ли вы чтобы был выполнен более тщательный поиск. В этом случае будет произведён поиск по всей файловой системе.

<sup>1</sup> Технически говоря: когда для языка существует несколько локалей с различными кодами стран.

<sup>2</sup> Установив средний или низкий приоритет, вы всегда можете выбрать предпочитаемую локаль из тех, что доступны для выбранного языка (если есть больше одной).

<sup>3</sup> Старые локаль — это локаль, не использующие UTF-8, с устаревшей стандартной символьной кодировкой, например ISO 8859-1 (используется для западно-европейских языков) или EUC-JP (используется для японского языка).

Если **iso-scan** не нашла ISO образ программы установки, перезагрузитесь в первоначальную операционную систему и проверьте правильность имени (должно оканчиваться на `.iso`), если файл находится на распознаваемой `debian-installer` файловой системе, и что образ не повреждён (проверьте контрольную сумму). Опытные пользователи Unix могут это сделать со второй консоли без перезагрузки.

### 6.3.1.5 Настройка сети

Если в начале этого шага система обнаруживает, что имеется более одного сетевого устройства, то вас попросят указать, какое устройство будет *первичным* сетевым интерфейсом, то есть какое устройство будет использовано для установки системы. Другие интерфейсы останутся ненастроенными. Вы сможете настроить дополнительные интерфейсы после завершения установки; смотрите страницу руководства `interfaces(5)`.

**6.3.1.5.1 Автоматическая настройка сети** По умолчанию, `debian-installer` пытается настроить сетевой интерфейс вашего компьютера автоматически. Если это сделать не удаётся, то это могло произойти по многим причинам, начиная от не подключённого сетевого кабеля и кончая неправильной настройкой инфраструктуры автоматической настройки. Для выяснения причины посмотрите сообщения об ошибках на четвёртой консоли. В любом случае вас спросят, не хотите ли вы попробовать ещё раз, или хотите выполнить настройку вручную. Иногда сетевые службы автоматической настройки очень медленно отвечают, поэтому если вы уверены, что всё в порядке, попробуйте ещё раз. Если автоматическая настройка завершается с ошибкой постоянно, то вместо неё выберите настройку сети вручную.

**6.3.1.5.2 Настройка сети вручную** При ручной настройке сети вам зададут несколько вопросов о вашей сети, а именно про IP-адрес, маску сети, шлюз, адреса серверов определения имён и имен и машины. Кроме того, если используется интерфейс беспроводной сети, то вас попросят ввести личный Wireless ESSID и ключ WEP или ключевую фразу WPA/WPA2. Введите ответы из Раздел 3.3.

#### Замечание



Некоторые настройки иногда легко, но не всегда, определить вручную: программа предполагает, что IP-адрес сети вычисляется с помощью операции побитового И над IP-адресом системы и маской сети. Широковещательный адрес по умолчанию вычисляется с помощью операции побитового ИЛИ над IP-адресом системы и инверсной маской сети. Также угадывается шлюз. Если вы не можете найти какой-нибудь ответ, используйте предлагаемые системой — если нужно, вы можете изменить их сразу после установки системы, отредактировав `/etc/network/interfaces`.

**6.3.1.5.3 IPv4 и IPv6** Начиная с Debian GNU/Linux 7.0 («Wheezy»), `debian-installer` также поддерживает IPv6 (помимо «классического» IPv4). Поддерживаются все комбинации IPv4 и IPv6 (только IPv4, только IPv6 и обе адресации одновременно).

Автоматическая настройка IPv4 выполняется с помощью DHCP (протокола динамической настройки узла). Автоматическая настройка IPv6 поддерживает настройку без сохранения состояния (`stateless`) с помощью NDP (протокол обнаружения соседей, включающий назначение рекурсивного сервера DNS (RDNSS)), настройку с сохранением состояния с помощью DHCPv6 и смешанную автоматическую настройку без сохранения/с сохранением состояния (настройка адреса с помощью NDP, дополнительные параметры с помощью DHCPv6).

### 6.3.1.6 Настройка времени и часового пояса

Сначала программа установки попытается подключиться к серверу времени в Интернете (по протоколу *NTP*), чтобы выставить правильное системное время. Если это не удаётся, то программа установки будет считать правильными время и дату, полученные с системных часов при загрузке программы установки. В процессе установки невозможно выставить системное время вручную.

В зависимости от места, выбранного ранее в процессе установки, вам может быть показан список часовых поясов, возможных в вашей географической точке. Если в вашей местности только один часовой пояс и вы выполняете установку по умолчанию, то вопрос задан не будет и система выберет этот часовой пояс.

В экспертном режиме или при установке со средним приоритетом вам будет предложено выбрать «Универсальное координированное время» (UTC) в качестве часового пояса.

Если по какой-то причине вы хотите установить часовой пояс для устанавливаемой системы, который *не* является правильным в выбранном местонахождении, то есть два способа сделать это.

1. Простой способ: выберите другой часовой пояс после завершения установки и загрузки в новую систему. Это можно сделать с помощью следующей команды:

```
# dpkg-reconfigure tzdata
```

2. Второй способ: часовой пояс может быть задан при загрузке программы установки с помощью параметра `time/zone=значение`. Значение должно быть правильным часовым поясом, например **Europe/London** или **UTC**.

Для автоматизации установки часовой пояс может быть задан в файле ответов.

## 6.3.2 Добавление пользователей и паролей

Перед настройкой часов, программа установки предложит вам настроить учётную запись «root» и/или учётную запись для первого пользователя. Дополнительные учётные записи можно будет создать после завершения установки.

### 6.3.2.1 Установка пароля суперпользователя (root)

Учётная запись *root* также называемая *суперпользовательской* — это учётная запись, для которой нет никаких преград в вашей системе. Учётная запись *root* должна использоваться только для администрирования системы и как можно меньшее время.

Любой создаваемый вами пароль должен состоять как минимум из 6 символов и содержать заглавные и строчные буквы, а также знаки препинания. Уделите побольше внимания при установке пароля *root*, поскольку эта самая мощная учётная запись. Избегайте слов из словаря или использования любой персональной информации, которую можно угадать.

Если кто-то говорит вам, что ему нужен ваш пароль *root*, будьте очень осторожны. Обычно нет необходимости давать свой пароль *root* кому-либо, если только вы не администрируете машину вместе с другими системными администраторами.

In case you do not specify a password for the «root» user here, this account will be disabled but the **sudo** package will be installed later to enable administrative tasks to be carried out on the new system. By default, the first user created on the system will be allowed to use the **sudo** command to become root.

### 6.3.2.2 Создание учётной записи обычного пользователя

Система спросит вас, желаете ли вы сейчас создать учётную запись для обычного пользователя. Эта учётная запись должна использоваться в качестве вашей основной рабочей учётной записи. Вы *не должны* использовать учётную запись *root* в повседневной работе или в качестве персональной учётной записи.

Почему не должны? Одна из причин, по которой следует избегать использования привилегий *root* это то, что в качестве *root* можно очень легко сделать необратимые повреждения. Другая причина: вас могут обманом заставить запустить программу *троянского коня* — это программа, которая воспользуется преимуществами ваших возможностей суперпользователя, чтобы скомпрометировать систему. В любой хорошей книге по системному администрированию Unix эта тема рассматривается подробно — прочтите одну из них, если это для вас в новинку.

Сначала вас попросят ввести полное имя пользователя. Затем спросят имя пользовательской учётной записи; в общем случае, ваше имя или что-то похожее вполне подойдёт. И, наконец, вас попросят ввести пароль учётной записи.

Если в любой момент после установки вы захотите создать ещё одну учётную запись, воспользуйтесь командой **adduser**.

## 6.3.3 Разметка разделов и выбор точек монтирования

После того как определение оборудования было выполнено в последний раз, *debian-installer* должен быть во всей своей красе, подстроенный под нужды пользователя и готовый к выполнению кое-какой настоящей работы. Как написано в заголовке раздела, основной задачей нескольких следующих компонент будет

разметка дисков, создание файловых систем, назначение точек монтирования и, возможно, настройка RAID, LVM или шифрованных устройств.

Если вы плохо знакомы с процедурой разметки, или просто хотите знать детали процесса, смотрите Приложение С.

Сначала вам предложат автоматическую разметку или всего диска или только имеющего на нём свободного места. Это так называемая «направляющая» разметка. Если вы не хотите использовать авторазметку, выберите в меню Вручную.

### 6.3.3.1 Поддерживаемые параметры разметки

Программа разметки `debian-installer` довольно универсальна. Она позволяет создать много различных схем разметки используя различные таблицы разделов, файловые системы и расширенные блочные устройства.

Список доступных параметров зависит, в основном, от архитектуры, но есть и другие факторы. Например, на системах с ограниченной внутренней памятью доступны не все параметры. Параметры по умолчанию также могут различаться. Тип таблицы разделов, используемый по умолчанию, может, например, отличаться на дисках большого объёма по сравнению с дисками малого объёма. Некоторые параметры можно изменять только при установке со средним или низким приоритетом `debconf`; чем выше приоритет, тем более целесообразные настройки по умолчанию будут использоваться.

Программа установки поддерживает различные виды расширенных настроек и использование устройств хранения, которые во многих случаях можно использовать совместно.

- *Управление логическими томами (LVM)*
- *Программный RAID*  
Поддерживается RAID 0, 1, 4, 5, 6 и 10 уровня.
- *Шифрование*
- *Многолучевое устройство* (экспериментальная поддержка)  
Смотрите описание в [вики](#). Поддержка многолучевых устройств доступна только если её включить при запуске программы установки.

Поддерживаются следующие файловые системы.

- *ext2r0, ext2, ext3, ext4*  
В большинстве случаев файловой системой по умолчанию выбирается `ext4`; при использовании направляющей разметки для разделов `/boot` будет выбрана `ext2`.
- *jfs* (недоступна на всех архитектурах)
- *xfs* (недоступна на всех архитектурах)
- *reiserfs* (по выбору; недоступна на всех архитектурах)  
Поддержка файловой системы Reiser по умолчанию более недоступна. При запуске программы установки со средним или низким приоритетом `debconf`, её можно включить выбрав компонент `partman-reiserfs`. Поддерживается файловая система только третьей версии.
- *jffs2*  
Используется на некоторых системах только для чтения флеш-памяти. Создание новых разделов с `jffs2` не поддерживается.
- *FAT16, FAT32*

### 6.3.3.2 Направляющая разметка

Если вы выбрали направляющую разметку, то есть два пути: создавать разделы непосредственно на жёстком диске (классический метод), создавать их с помощью управления логическими томами (LVM) или создавать, используя LVM с шифрованием<sup>4</sup>.

<sup>4</sup> Программа установки шифрует групповые тома LVM 256-битным ключом AES и использует поддержку «dm-crypt» в ядре.

Замечание



LVM или LVM с шифрованием может быть недоступно на некоторых архитектурах.

При использовании LVM или LVM с шифрованием, программа установки создаст большинство разделов внутри одного большого раздела; преимущество этого метода в том, что у разделов внутри этого большого раздела можно легко изменить размеры позже. При использовании LVM с шифрованием большой раздел будет недоступен без специальной ключевой фразы, предоставляя тем самым дополнительную безопасность вашим (личным) данным.

При использовании зашифрованных LVM, программа установки также автоматически стирает диск, записывая на него произвольные данные. Это способствует усилению безопасности (так как это делает невозможным определить какие части диска используются и этим также стираются предыдущие следы установок), но это занимает некоторое время в зависимости от размера диска.

Замечание



При использовании направляющей разметки с LVM или LVM с шифрованием, перед тем как приступить к настройке LVM, нужно произвести запись на выбранный диск. Эти изменения сотрут все имеющиеся данные на выбранном жёстком диске и их больше нельзя будет восстановить. Однако программа установки перед записью на диск попросит вас подтвердить выполнение этих изменений.

Если вы выбрали направляющую разметку (или классическую, или использующую LVM (с шифрованием)) всего диска, то сначала вас попросят выбрать диск, который нужно использовать. Убедитесь, что предложены все имеющиеся диски, и если их несколько, правильно укажите нужный. Порядок в списке предлагаемых дисков может быть отличным от того, какой вы использовали ранее. Идентифицировать диски вам поможет их показанный размер.

Любые данные на выбранном диске впоследствии будут уничтожены, но перед любой операцией записи на диск у вас попросят подтверждения. Если выбран классический метод разметки, то вы можете отменить любые изменения даже в самом конце; при использовании LVM (с шифрованием) это невозможно.

Далее вам предложат выбрать одну из схем, описанных в таблице ниже. Любая схема имеет свои "за" и "против", некоторые обсуждаются в Приложение С. Если вы не знаете что выбрать, выберите первую. Помните, что для работы направляющей разметки требуется определённое минимальное количество свободного места с которым можно работать. Если вы не выделите по крайней мере около 1ГБ (зависит от выбранной схемы), направляющая разметка завершится неудачно.

Схема разметки	Минимальное пространство	Создаваемые разделы
Все файлы на одном разделе	600МБ	/, swap
Отдельный раздел /home	500МБ	/, /home, swap
Отдельные разделы /home, /var и /tmp	1ГБ	/, /home, /var, /tmp, swap

Если вы выбрали направляющую разметку с использованием LVM (с шифрованием), то программа установки также создаст отдельный раздел /boot. Остальные разделы, включая раздел подкачки, будут созданы внутри раздела LVM.

После выбора схемы, следующий экран покажет только что созданную таблицу разделов, включая информацию будут ли и как форматироваться разделы и куда они будут смонтированы.

Список разделов может выглядеть так:

```
SCSI1 (0,0,0) (sda) - 6.4 GB WDC AC36400L
#1 primary 16.4 MB B f ext2 /boot
#2 primary 551.0 MB swap swap
#3 primary 5.8 GB ntfs
pri/log 8.2 MB FREE SPACE
```

```

SCSI2 (1,0,0) (sdb) - 80.0 GB ST380021A
#1 primary 15.9 MB ext3
#2 primary 996.0 MB fat16
#3 primary 3.9 GB xfs /home
#5 logical 6.0 GB f ext4 /
#6 logical 1.0 GB f ext3 /var
#7 logical 498.8 MB ext3

```

Здесь показаны два жёстких диска, разделённые на несколько разделов; на первом диске осталось немного свободного места. Каждая строка раздела содержит номер раздела, тип, размер, необязательные флаги, файловую систему и точку монтирования (если есть). Замечание: такая настройка не может быть создана с помощью направляющей разметки, это просто пример, созданный с помощью разметки вручную.

На этом направляющая разметка завершена. Если вам нравится сгенерированная таблица разделов, то вы можете выбрать Закончить разметку и записать изменения на диск из меню для создания новой таблицы разделов (как описано в конце этого раздела). Если результат вам не нравится, то можно выбрать Отменить изменение разделов и запустить направляющую разметку снова или изменить предложенные изменения вручную, как описано ниже.

### 6.3.3.3 Разметка вручную

При ручной разметке появляется похожий экран, за исключением того, что отображается существующая таблица разделов и нет точек монтирования. Далее описано как вручную редактировать таблицу разделов и как использовать эти разделы в Debian.

Если вы выбрали чистый диск, на котором нет ни разделов ни свободного места, то вам предложат создать новую таблицу разделов (это необходимо для создания новых разделов). После этого, на выбранном диске должна появиться новая строка «СВОБОДНОЕ МЕСТО».

Если вы выберете свободное пространство, вам предложат создать новый раздел. Вы должны ответить на несколько коротких вопросов о размере, типе (первичный или логический) и местоположении (в начале или конце свободного пространства). После этого, вам покажут предлагаемые параметры нового раздела. Основным параметром является Использовать как:, который определяет, будет ли на разделе файловая система или он будет использован как раздел подкачки, программный RAID, LVM, будет содержать зашифрованную файловую систему или не будет использован вовсе. Другие настройки задают точку и параметры монтирования, флаг загрузки; список показываемых параметров зависит от того, как будет использоваться раздел. Если вам не нравятся значения по умолчанию, измените их как хотите. Например, в параметре Использовать как:, вы можете выбрать другую файловую систему, поменять параметры для раздела подкачки, программного RAID, LVM или совсем его не использовать. Когда вы настроите новый раздел, выберите Закончить с разделом и вы вернётесь обратно в главное меню **partman**.

Если вы решили, что хотите что-то изменить в разделе, просто выберите раздел и попадёте обратно в меню настройки раздела. Так как это тоже самое меню что и для создания раздела, вы можете изменить тот же набор параметров. Одна вещь, которая может показаться, на первый взгляд, не очевидной: вы можете изменить размер раздела, выбрав элемент, отображающий размер раздела. Это работает, как минимум, на файловых системах fat16, fat32, ext2, ext3 и с разделом подкачки. Также, из этого меню можно удалить раздел.

Убедитесь, что создали, по крайней мере, два раздела: один для *корневой* файловой системы (который должен быть смонтирован как /) и один раздел *подкачки*. Если вы забыли указать точку монтирования корневой файловой системы, **partman** не разрешит вам продолжить, пока вы не исправите ошибку.

Возможности **partman** могут быть расширены модулями программы установки, но это зависит от архитектуры компьютера. Поэтому, если вы не найдёте всех обещанных свойств, проверьте, загрузили ли вы все требуемые модули (например, `partman-ext3`, `partman-xfs` или `partman-lvm`).

Для завершения работы по разметке, выберите Закончить разметку и записать изменения на диск из меню разметки. Вам покажут отчёт об изменениях, которые будут произведены на диске и попросят подтвердить, что все представленные файловые системы должны быть созданы.

### 6.3.3.4 Настройка устройств Multidisk (программный RAID)

Если у вас более одного жёсткого диска<sup>5</sup> в компьютере, то можно воспользоваться программой **mdcfdg** для увеличения производительности дисков и/или повышения надёжности хранения данных. В результате получается так называемое *устройство Multidisk* (или другое, более известное название, *программный RAID*).

<sup>5</sup> Если честно, вы можете собрать MD устройство даже из обычных разделов, расположенных на одном физическом жёстком диске, но это не даст никакой выгоды.

Основу MD составляют группы разделов, расположенные на различных дисках и объединённые вместе в *логическое* устройство. Затем это устройство можно использовать как обычный раздел (то есть в **partman** вы можете отформатировать его, назначить ему точку монтирования и т.д.).

Получаемые преимущества зависят от типа создаваемого MD устройства. В данный момент поддерживаются:

**RAID0** Применяется где нужна производительность. RAID0 разбивает все входящие данные на *полосы* и равномерно распределяет их по дискам в массиве. Это может увеличить скорость операций чтения/записи, но если один из дисков сломается, вы потеряете *всё* (часть информации всё ещё на исправных дисках(е), другая часть *хранилась* на сломавшемся диске).

Обычно, разделы RAID0 используют при редактировании видео.

**RAID1** Используется там, где надёжность превыше всего. Он состоит из нескольких (обычно двух) одинаковых по размеру разделов, и на каждом разделе хранятся одинаковые данные. В сущности, это означает три вещи. Первое: если один из дисков сломается, вы не теряете данные, так как они продублированы на оставшихся дисках. Второе: вам доступна только часть имеющегося объёма (более точно, размер самого маленького раздела в RAID). Третье: нагрузка при чтении файла сбалансированно распределяется по всем дискам, что может повысить производительность, например, файлового сервера, на котором операций чтения больше чем операций записи.

Также, вы можете иметь в массиве запасной диск, который включится в работу в случае неисправности одного из дисков.

**RAID5** Хороший компромисс между скоростью, надёжностью и избыточностью. RAID5 разделяет все входящие данные на полосы и распределяет их пропорционально на все диски кроме одного (как и RAID0). В отличие от RAID0, RAID5 вычисляет *контрольную сумму*, которую записывает на оставшийся диск. Диск с контрольной суммой не статичен (тогда это был бы RAID4), а периодически изменяется, так что контрольная сумма распределяется по всем дискам. Когда один из дисков ломается, потерянную часть информации можно вычислить из оставшихся данных и её контрольной суммы. RAID5 должен состоять, как минимум из трёх активных разделов. Также, вы можете иметь в массиве запасной диск, который включится в работу в случае неисправности одного из дисков.

Как видите, RAID5 имеет немного надёжности от RAID1, но это достигается с меньшей избыточностью. С другой стороны, он немного медленнее на операциях записи, чем RAID0 из-за вычислений контрольной суммы.

**RAID6** Подобен RAID5, но контрольная сумма хранится на двух дисках, а не на одном.

Массив RAID6 останется рабочим даже после выхода из строя сразу двух дисков.

**RAID10** В RAID10 используется чередование (как в RAID0) и зеркалирование (как в RAID1). В нём создаётся *n* копий входящих данных, которые распределяются между разделами так, что все копии одинаковых данных хранятся на разных устройствах. По умолчанию *n* равно 2, но в режиме эксперта это можно изменить. Число задействованных разделов должно быть как минимум *n*. В RAID10 есть различные схемы распределения копий. По умолчанию используется схема близкого (*near*) распределения. При этом все одинаковые копии располагаются на дисках по одинаковому смещению. Далёкие (*far*) копии имеют разные смещения на дисках. Смещение копий относится ко всей полосе, а не к отдельной копии.

RAID10 может использоваться для достижения надёжности и избыточности без затрат на подсчёты контрольных сумм.

Итог:

Тип	Минимум устройств	Запасное устройство	Сохранятся ли данные при отказе диска?	Доступное пространство
RAID0	2	нет	нет	Размер самого маленького раздела умноженный на число устройств в RAID

Тип	Минимум устройств	Запасное устройство	Сохранятся ли данные при отказе диска?	Доступное пространство
RAID1	2	необязательно	да	Размер самого маленького раздела в RAID
RAID5	3	необязательно	да	Размер самого маленького раздела умноженный на (число устройств в RAID минус один)
RAID6	4	необязательно	да	Размер самого маленького раздела умноженный на (число устройств в RAID минус два)
RAID10	2	необязательно	да	Общее количество всех разделов, поделённое на число полос копий (по умолчанию два)

Если вы хотите узнать больше о программном RAID, смотрите [Software RAID HOWTO](#).

Для создания MD устройства нужно пометить желаемые разделы как используемые в RAID. (Это делается в **partman** из меню Настройки раздела, где вы должны выбрать Использовать как: → Использовать раздел как устройство RAID.)

**Замечание**



Убедитесь, что система может загружаться с планируемой вами схемы разметки. Обычно, требуется создать отдельную файловую систему для /boot, если в качестве корневой файловой системы (/) используется RAID. Большинство загрузчиков поддерживают зеркало (без чередования!) RAID1, так что решением может стать, например, RAID5 для / и RAID1 для /boot.

Далее, вы должны выбрать Настройка программного RAID из главного меню **partman**. (Меню появится только после того как вы отметите хотя бы один раздел используемый как физический том для RAID.) На первом экране **mdcfg** просто выберите Создать MD устройство. Вам будет предложен список поддерживаемых типов MD устройств, из которого вы должны выбрать один (например, RAID1). Дальнейшие действия зависят от типа выбранного MD.

- RAID0 очень прост — вам предложат список доступных разделов RAID, а вашей задачей будет выбор разделов, из которых вы хотите сформировать MD.
- С RAID1 немного сложнее. Во-первых, вас попросят ввести количество активных устройств и количество запасных устройств, из которых будет формироваться MD. Далее, вам потребуется выбрать из списка доступных разделов RAID те, которые будут активными и затем, те которые будут запасными. Число выбранных разделов должно быть равно числу, введённому несколькими секундами ранее. Не волнуйтесь. Если вы ошибётесь и выберете неверное число разделов, `debian-installer` не позволит вам продолжить, пока вы не исправите ошибку.
- Создание RAID5 напоминает процедуру RAID1 за исключением того, что вам нужно использовать как минимум *три* активных раздела.



- Создание RAID6 напоминает процедуру RAID1 за исключением того, что вам нужно использовать как минимум *четыре* активных раздела.
- Создание RAID10 также похоже на процедуру создания RAID1, если это происходит не в режиме эксперта. В экспертном режиме `debian-installer` попросит указать схему распределения. В схеме задаётся два параметра. Первый — это тип схемы. Его значение может быть `n` (для близких копий), `f` (для далёких копий), или `o` (смещение копий). Вторым параметром задаётся число копий с данными. Оно должно быть равно, как минимум, числу активных дисков, чтобы было возможно распределить все копии по разным дискам.

Можно иметь несколько типов MD одновременно. Например, если вы имеете три жёстких диска по 200 ГБ выделенных под MD, каждый из которых разбит на два 100 ГБ раздела, можно объединить первые разделы трёх дисков в RAID0 (быстрый 300 ГБ раздел для редактирования видео) и использовать оставшиеся три раздела (2 активных и 1 запасной) для RAID1 (вполне надёжный 100 ГБ раздел для `/home`).

После настройки MD устройств в соответствии с вашими желаниями, вы можете выбрать Закончить `mdcfg` и вернуться в `partman`, чтобы создать файловые системы на новых MD устройствах и назначить им обычные атрибуты типа точек монтирования.

### 6.3.3.5 Настройка менеджера логических томов (LVM)

Если вы работаете с компьютером на уровне системного администратора или «продвинутого» пользователя, то, определённо, попадали в ситуацию, когда на дисковом разделе (обычно на самом важном) заканчивалось место, а соседний раздел был почти пуст и вы выбирались из этой ситуации переносом данных с одного раздела на другой, созданием символических ссылок и т.д.

Чтобы избежать описанной ситуации вы можете воспользоваться менеджером логических томов (LVM). Проще говоря, с помощью LVM вы можете объединить разделы (*физические тома* в терминах LVM) в виртуальный диск (так называемую *группу томов*), который затем можно поделить на виртуальные разделы (*логические тома*). Выгода в том, что логические тома (и, конечно, содержащие их группы томов) можно распределить по нескольким физическим дискам.

Теперь, когда вы поняли, что вам нужно ещё место на имеющемся 160ГБ `/home` разделе, вы можете просто добавить новый 300ГБ диск в компьютер, объединить его с существующей группой томов и затем изменить размер логического тома, который содержит файловую систему с `/home` и вуаля, ваши пользователи имеют дополнительное место на новом 460ГБ разделе. Этот пример, конечно, сильно упрощён. Дополнительную информацию вы найдёте в [LVM HOWTO](#).

Настройка LVM в `debian-installer` очень проста и полностью поддерживается `partman` Во-первых, отметьте раздел(ы), которые будут использоваться в качестве физических томов LVM. Это делается с в меню Настройки раздела, где вы должны выбрать Использовать как: → физический том для LVM.

После того как вы вернётесь в главное меню `partman`, вы увидите новый пункт Настройка менеджера логических томов (LVM). Выбрав его, сначала вас попросят подтвердить сделанные изменения (если есть) в таблице разделов, а затем будет показано меню настройки LVM. Сверху меню кратко показана настройка LVM. Сам состав меню зависит от выбранных настроек и содержит только допустимые действия. Возможные действия:

- Показать настройку подробнее: показывает структуру LVM устройства, имена и размеры логических томов и т.д.
- Создать группу томов
- Создание логических томов
- Удалить группу томов
- Удаление логических томов
- Расширить группу томов
- Уменьшить группу томов
- Закончить: вернуться в главное меню `partman`

С помощью этого меню сначала создайте логическую группу, а затем внутри неё логические тома.

После возвращения в главное меню `partman`, вы должны увидеть все созданные логические тома, как обычные разделы (и настраиваются они одинаково).

### 6.3.3.6 Настройка шифрованных томов

`debian-installer` позволяет настраивать шифрованные разделы. Каждый файл, записываемый на такой раздел, немедленно сохраняется на устройстве в шифрованном виде. Доступ к шифрованным данным разрешается только после ввода *ключевой фразы*, которая задаётся при создании шифрованного раздела. Эта возможность полезна для защиты конфиденциальных данных на ноутбуке или жёстком диске при краже. Вор может получить физический доступ к жёсткому диску, но не зная правильной ключевой фразы, данные на диске выглядят как случайный набор символов.

Два наиболее важных раздела для шифрования: раздел `/home`, где хранятся ваши личные данные, и раздел подкачки, куда важные данные могут временно попасть во время работы с ними. Естественно, нет никаких препятствий, чтобы зашифровать любые другие разделы, которые могут представлять интерес. Например, `/var`, где сервера баз данных, почтовые сервера или сервера печати хранят свои данные, или `/tmp`, который используется разными программами для хранения потенциально важных временных файлов. Некоторые люди могут захотеть зашифровать всю систему. Исключение составляет лишь раздел `/boot`, который должен остаться не шифрованным, так как пока нельзя загрузить ядро с шифрованного раздела.

#### Замечание



Заметим, что производительность шифрованных разделов будет меньше чем не шифрованных, так как при каждом чтении и записи данных происходит шифрование и дешифрование. Производительность зависит от быстродействия процессора, выбранного алгоритма шифрования и длины ключа.

Для использования шифрования создайте новый раздел выделив свободное пространство в главном меню разметки. Также можно выбрать уже существующий раздел (например, обычный раздел, логический том LVM или том RAID). В меню Настройки раздела для Использовать как: нужно выбрать физический том для шифрования. После этого в меню для раздела появятся несколько параметров шифрования.

Поддерживаемый `debian-installer` метод шифрования: `dm-crypt` (включён в новых ядрах Linux, доступен для физических томов LVM узла).

Давайте рассмотрим возможные параметры, если в качестве метода шифрования выбран **Device-mapper (dm-crypt)**. Как и всегда: если не знаете, используйте значения по умолчанию, так как они вдумчиво выбирались исходя из соображений безопасности.

**Шифрование: aes** Этот параметр позволяет выбрать алгоритм шифрования (*шифр*), который будет использоваться для шифрования данных раздела. `debian-installer` поддерживает следующие блочные шифры: `aes`, `blowfish`, `serpent` и `twofish`. Здесь не будет обсуждаться стойкость этих алгоритмов, однако, при выборе вам наверно будет небезынтересно узнать, что в 2000 году AES был выбран Национальным Институтом Стандартизации США в качестве стандартного алгоритма шифрования для защиты конфиденциальной информации в 21-ом веке.

**Размер ключа: 256** Здесь вы можете указать длину ключа шифрования. С увеличением размера ключа стойкость шифрования обычно усиливается. С другой стороны, увеличение длины ключа обычно негативно сказывается на производительности. Допустимые размеры ключа зависят от шифра.

**IV алгоритм: xts-plain64** Вектор инициализации или IV алгоритм используется в криптографии для того, чтобы при шифровании одного и того же *обычного текста* одинаковым ключом всегда получался уникальный *шифрованный текст*. Это было введено для того, чтобы при взломе шифра предотвратить нахождение повторяющихся участков в шифрованных данных.

Из предлагаемых возможностей, значение по умолчанию `xts-plain64` является на сегодняшний момент наименее уязвимым к известным атакам. Используйте альтернативы только когда это нужно для совместимости с ранее установленной системой, в которой невозможно задействовать новые алгоритмы.

**Ключ шифрования: ключевая фраза** Здесь можно ввести тип ключа шифрования для этого раздела.

**Ключевая фраза** Ключ шифрования будет создан<sup>6</sup> на основе ключевой фразы, которую вы сможете ввести позже.

<sup>6</sup> Использование ключевой фразы в качестве ключа означает, что раздел будет настроен с помощью **LUKS**.

**Произвольный ключ** Новый ключ шифрования будет генерироваться из произвольных данных всякий раз при монтировании раздела. Другими словами: при каждом выключении машины содержимое раздела теряется, так как ключ удаляется из памяти. (Естественно, вы можете попробовать найти ключ перебором всех вариантов, но если в алгоритме шифрования нет неизвестных уязвимостей, это недостижимо за нашу продолжительность жизни.)

Произвольные ключи полезны для разделов подкачки, так как вам не нужно запоминать ключевую фразу из вычищать конфиденциальную информацию из раздела подкачки перед выключением компьютера. Однако, это также означает, что вы *не* сможете использовать функцию «suspend-to-disk» из новых ядер Linux, так как невозможно (во время последующей загрузки) восстановить приостановленные данные записанные в раздел подкачки.

**Стереть данные: да** Задаёт, нужно ли перезаписать содержимое раздела произвольными данными перед настройкой шифрования. Это рекомендуется делать, так как иначе это может позволить злоумышленнику разгадать какие части раздела используются, а какие нет. Также, это усложнит процесс восстановления любых оставшихся от предыдущей установки данных<sup>7</sup>.

После выбора желаемых параметров для шифруемых разделов, вернитесь обратно в главное меню разметки. Там должен появиться новый пункт Настроить шифрование для томов. После того как вы его выберете, вас попросят подтвердить удаление данных на разделах, помеченных на стирании и возможно, другие действия, типа записи новой таблицы разделов. Для огромных разделов это может занять некоторое время.

Затем вас попросят ввести ключевую фразу для разделов, настроенные на её использование. Хорошие ключевые фразы должны быть больше 8 знаков, должны содержать вперемешку буквы, числа и другие символы и не должны содержать общеупотребительных словарных слов или информации, которая как-то относится к вам лично (например, дни рождения, хобби, имена домашних животных, имена родственников и тому подобное).

#### Внимание



Перед вводом ключевых фраз вы должны убедиться, что клавиатура настроена правильно и генерирует ожидаемые символы. Если вы не уверены, можно переключиться на вторую виртуальную консоль и попробовать ввести текст. Это застрахует вас от неожиданностей в последствии, например, если вы вводите ключевую фразу на клавиатуре с раскладкой qwerty, а во время установки вы использовали раскладку azerty. Такая ситуация может случиться по нескольким причинам. Вы могли переключиться на другую раскладку клавиатуры во время установки, или выбранная раскладка ещё не активирована ко времени ввода ключевой фразы для корневой файловой системы.

Если для создания ключей шифрования вы выбрали использование методов без ключевой фразы, то они будут сгенерированы. Так как ядро могло не собрать достаточное количество энтропии на данной ранней стадии установки, процесс может занять продолжительное время. Вы можете ускорить процесс генерации энтропии: например, нажимая произвольные клавиши, или переключиться в оболочку командной строки на второй виртуальной консоли и создать некоторый обмен данными по сети или с диском (загрузив какие-то файлы, скопировать большие файлы в /dev/null и т.д.). Это нужно сделать для каждого шифруемого раздела.

После возвращения в главное меню разметки, вы увидите все зашифрованные тома как дополнительные разделы, которые можно настраивать как обычные разделы. В следующем примере показан том, зашифрованный с помощью dm-crypt

```
Encrypted volume (sda2_crypt) - 115.1 GB Linux device-mapper
#1 115.1 GB F ext3
```

Теперь можно назначить точки монтирования и изменить тип файловых систем, если значения по умолчанию чем-то не устраивают.

Запишите где-нибудь информацию о соответствии идентификаторов в скобках (*sda2\_crypt* в данном случае) и точках монтирования, которые вы назначили зашифрованным томам. Эти данные вам потребуются

<sup>7</sup> Однако, есть мнение, что ребята из служб на три буквы могут восстанавливать данные даже после их неоднократной перезаписи на магнитооптических носителях.

позже при загрузки новой системы. Различия между обычным процессом загрузки и загрузки с шифрованием описаны далее в Раздел 7.2.

После того как вас полностью устроит схема разметки, продолжайте установку.

### 6.3.4 Установка базовой системы

Хотя эта стадия наименее проблематична, она занимает наибольшую часть времени установки, потому что загружается, проверяется и распаковывается вся базовая система. Если у вас медленный компьютер или сетевое соединение, это может занять определённое время.

Во время установки базовой системы сообщения о распаковке и настройке пакетов переправляются на **tty4**. Вы можете перейти на этот терминал, нажав левый Alt-F4; чтобы вернуться обратно в основной процесс установки нажмите левый Alt-F1.

Сообщения распаковки/настройки во время этого этапа сохраняются в файле `/var/log/syslog`. Вы можете посмотреть их, если установка выполняется через консоль на последовательном порту.

В одной из частей установки выполняется установка ядра Linux. С приоритетом по умолчанию программа установки выберет ядро наиболее подходящее для вашего оборудования. В режимах с низким приоритетом, это предложат сделать вам из списка доступных ядер.

При установке пакетов с помощью системы управления пакетами по умолчанию также производится установка пакетов, рекомендуемых устанавливаемыми пакетами. Рекомендуемые пакеты не требуются для работоспособности основных возможностей выбранного ПО, но они добавляют новые возможности этому ПО и должны, по мнению сопровождающих, обычно устанавливаться вместе с этим ПО.

#### Замечание



По техническим причинам, пакеты, устанавливаемые во время установки базовой системы, устанавливаются без их «Рекомендуемых» пакетов. Правило, описанное выше, действует только после данного этапа процесса установки.

### 6.3.5 Установка дополнительного программного обеспечения

На этом этапе уже есть рабочая, но ограниченная по возможностям система. Большинство пользователей хотело бы добавить дополнительное программное обеспечение в систему, чтобы решать свои задачи, и программа установки позволяет это сделать. Этот этап может продлиться дольше чем установка базовой системы, если у вас медленный компьютер или соединение с сетью.

#### 6.3.5.1 Настройка apt

One of the tools used to install packages on a Debian GNU/Linux system is the program **apt**, from the **apt** package<sup>8</sup>. Other front-ends for package management, like **aptitude** and **synaptic**, are also in use. These front-ends are recommended for new users, since they integrate some additional features (package searching and status checks) in a nice user interface.

Пакету **apt** нужно указать откуда получать пакеты. Результат настройки сохраняется в файле `/etc/apt/sources.list`. Вы можете проверить и изменить его по желанию после завершения установки.

Если вы выполняете установку с приоритетом по умолчанию, то программа установки выполнит большую часть настройки автоматически, основываясь на используемом методе установки и, возможно, на ответах, указанных вами ранее. В большинстве случаев программа установки автоматически добавит зеркало безопасности и, если вы используете стабильный дистрибутив, зеркало сервиса «stable-updates».

Если вы производите установку с низким приоритетом (например, в экспертном режиме), то вы сможете принять больше решений самостоятельно. Вы сможете выбрать использовать ли сервисы обновлений безопасности и/или stable-updates, а также добавлять ли секции пакетов «contrib» и «non-free».

<sup>8</sup> Note that the program which actually installs the packages is called **dpkg**. However, this program is more of a low-level tool. **apt** is a higher-level tool, which will invoke **dpkg** as appropriate. It knows how to retrieve packages from your CD, the network, or wherever. It is also able to automatically install other packages which are required to make the package you're trying to install work correctly.

**6.3.5.1.1 Установка с нескольких CD или DVD дисков** Если вы производите установку с одного CD или DVD из набора дисков, то программа установки спросит, хотите ли вы просканировать дополнительные CD или DVD диски. Если у вас есть дополнительные диски CD или DVD, то вы, вероятно, захотите это сделать для того, чтобы программа установки могла использовать пакеты с этих дисков.

Если у вас нет дополнительных дисков, это не проблема: их наличие не требуется. Если у вас также нет доступа к сетевому серверу-зеркалу пакетов (описано в следующей разделе) то, возможно, не все пакеты из выбранных на следующем этапе задач смогут быть установлены.

#### Замечание



Пакеты распределяются по дискам CD (и DVD) согласно популярности. Это значит, что для большинства людей хватит первого CD из набора, и только некоторые люди на самом деле используют пакеты с последнего CD-диска.

Это также означает, что покупка или загрузка и прожиг полного набора CD пустая трата денег, так как большинство дисков просто не будут использоваться. В большинстве случаев достаточно получить только первые от 3 до 8 CD, а устанавливать нужные дополнительные пакеты можно из Интернет с сервера-зеркала. Тоже касается и DVD наборов: первого DVD или, может быть, первых двух DVD хватит для большинства задач.

Если вы просканируете несколько CD или DVD-дисков, программа установки будет выдавать сообщение при необходимости смены CD/DVD в приводе. Заметим, что должны сканироваться CD или DVD из одного набора. Порядок сканирования значения не имеет, но сканирование по порядку уменьшает вероятность сделать ошибку.

**6.3.5.1.2 Использование сетевого сервера-зеркала пакетов** Один из часто задаваемых вопросов: использовать или нет сетевой сервер-зеркало в качестве источника пакетов. В большинстве случаев значение ответа по умолчанию вполне приемлемо, но не всегда.

Если вы *не* устанавливаете систему с полного CD или DVD диска или с полного образа CD/DVD, то вы определённо должны использовать зеркала из сети, иначе в конце установки получится очень ограниченная по возможностям система. Однако, если у вас дорогое или медленное подключение к Интернет, то лучше *не* выбирать задачу рабочий стол на следующем шаге установки.

Если вы устанавливаете систему с одного полного CD-диска или с полного образа CD, то использование зеркала не обязательно, но настоятельно рекомендуется, так как на одном CD содержится совсем немного пакетов. Если у вас дорогое или медленное подключение к Интернет, то пока лучше *не* выбирать сетевое зеркало, а закончить установку с доступного CD и установить дополнительные пакеты после установки (то есть после перезагрузки в новую систему).

Если вы производите установку с DVD диска или образа, то все нужные для установки пакеты будут на первом DVD. Все пакеты также доступны, если вы просканируете несколько CD-дисков, как это описывалось в предыдущем разделе. Использовать сетевой сервер-зеркало в данном случае необязательно.

Преимуществом добавления сетевого зеркала является то, что обновления пакетов, произошедшие с момента прожига CD/DVD дисков, становятся доступными сразу во время установки, что расширяет срок службы вашего набора CD/DVD без ущерба безопасности или стабильности устанавливаемой системы.

Итог: выбор сетевого сервера-зеркала обычно хорошая идея, если у вас хорошее соединение с Интернет. Если нужная версия пакета есть на CD/DVD, то программа установки будет всегда использовать этот носитель. Размер данных, которые требуется скачать, если вы выбрали зеркало, зависит от

1. задач, которые вы выберете на следующем этапе установки,
2. пакетов, которые необходимы для этих задач,
3. какие из этих пакетов есть на просканированных CD/DVD, и
4. есть ли обновлённые версии пакетов, имеющихся на CD или DVD, на сервере-зеркале (или на обычном зеркале пакетов, или на зеркалах обновлений безопасности или stable-updates).

Заметим, что даже если вы выберете не использовать сетевое зеркало, некоторые пакеты могут всё равно быть скачаны из Интернет, если для них есть, согласно сервису безопасности или stable-updates, новые версии, и если эти были сервисы настроены.

**6.3.5.1.3 Выбор сетевого сервера-зеркала пакетов** Если при установке вы выбрали использование сетевого сервера-зеркала пакетов (необязательно при установке с CD/DVD, обязательно для образов netboot), вам будет показан список ближайших к вам географически сетевых зеркал (и, скорее всего, более быстрых) на основе ранее указанной вами страны. Предлагаемый выбор по умолчанию, обычно, работает хорошо.

Зеркало также можно задать вручную выбрав «ввести информацию вручную». После этого вы можете указать имя узла-зеркала и необязательный порт. Как в Wheezy, он может быть подобен URL, т. е. при указании адреса IPv6 вокруг можно ввести квадратные скобки, например «[2001:db8::1]».

Если ваш компьютер имеет только адрес IPv6 (что, вероятно, не так, для подавляющего большинства пользователей), указанное по умолчанию зеркало для вашей страны может не работать. Все зеркала из списка доступны по адресам IPv4, и только некоторые из них имеют адреса IPv6. Доступность отдельных зеркал может с течением времени меняться, эта информация недоступна программе установки. Если для вашей страны зеркало по умолчанию недоступно через IPv6, то вы можете попробовать другое зеркало или «ввести информацию вручную». При этом в качестве имени зеркала вы можете указать «ftp.ipv6.debian.org», которое является псевдонимом зеркала, доступного по IPv6, хотя оно не будет таким быстрым.

Другой возможностью выбора зеркала вручную является ввод «http.debian.net». Именем «httpredir.debian.org» адресуется не физический сервер-зеркало, а служба перенаправления, которая автоматически перенаправит вашу систему на ближайшее к вам с точки зрения сетевой топологии реально существующее зеркало. Она учитывает протокол подключения, то есть, если вы используете IPv6, то она перенаправит вас на ближайшее зеркало, поддерживающее IPv6.

### 6.3.5.2 Выбор и установка программного обеспечения

Во время процесса установки вам будет предложено выбрать дополнительное программное обеспечение для установки. Вместо выбора каждой программы из 74360 доступных пакетов, этот этап позволяет сосредоточиться на выборе и установке заранее подготовленных наборов ПО, чтобы быстро настроить компьютер на выполнение определённых задач.

Поэтому, сначала вы можете выбрать *задачи*, а позже установить дополнительные пакеты. В общем, под задачей подразумеваются какие-то работы или задания, которые вы хотите выполнять на компьютере, например, использовать его как «среду рабочего стола», чтобы он служил в качестве «веб-сервера» или «сервера печати»<sup>9</sup>. В таблице Раздел D.2 показано необходимое пространство для доступных задач.

Некоторые задачи могут быть уже выбраны в зависимости от характеристик компьютера на который производится установка. Если вы не согласны с таким выбором, то можете отменить лишние задачи. Вы даже можете вообще ничего не устанавливать с помощью задач.

#### Подсказка



В стандартном пользовательском интерфейсе программы установки для выбора/отмены задач используется клавиша пробел.

<sup>9</sup> Вы должны знать, что для предоставления этого списка программа установки просто вызывает программу **tasksel**. Её можно запустить в любое время после установки, чтобы установить (или удалить) некоторые пакеты, или же вы можете воспользоваться другим более точным инструментом **aptitude**. Если вам нужен какой-то определённый пакет, то после завершения установки просто запустите **aptitude install пакет**, где *пакет* — это имя требуемого пакета.

**Замечание**

По задаче «Окружение рабочего стола» будет установлено графическое окружение рабочего стола.

По умолчанию `debian-installer` устанавливает окружение рабочего стола. Во время установки возможно выбрать другое окружение рабочего стола. Также возможно установить несколько окружений, но некоторые комбинации не могут быть установлены одновременно.



Заметим, что это сработает, если доступны все пакеты, необходимые для желаемого рабочего стола. Если вы производите установку только с одного полного CD образа, то пакеты нужно скачивать с сервера-зеркала, так как большинство необходимых пакетов включены только на следующих CD-дисках. Установка любого окружения рабочего стола таким способом замечательно работает при использовании образа DVD или другого метода установки.

Согласно различным серверным задачам будет установлено ПО, указанное ниже. Веб-сервер: `apache2`; сервер печати: `cups`; сервер SSH: `openssh`.

По задаче «Стандартная система» установятся все пакеты с приоритетом «стандарт». Сюда входит множество общих утилит, которые есть в любой Linux или Unix системе. Вы должны оставить эту задачу выбранной, если вы новичок и не хотите получить действительно минимально работоспособную систему.

Если в следствии выбора языка локаль по умолчанию стала отличаться от локали «C», то `tasksel` проверит, определены ли какие-либо задачи локализации для этой локали, и автоматически попытается установить подходящие пакеты локализации. Например, к таким пакетам относятся пакеты со словарями или специальными шрифтами для вашего языка. Если выбрана установка окружения рабочего стола, то также будут установлены соответствующие пакеты локализации (если есть).

После выбора задач нажмите Continue. Теперь `aptitude` начнёт установку пакетов, которые являются частью указанных задач. Если какой-то программе нужна дополнительная информация от пользователя, то она задаст её во время данного процесса.

Хотим сообщить, что есть очень большие задачи, особенно Рабочий стол. Если установка происходит с обычного CD-ROM и с зеркала пакетов, которых нет на CD-ROM, то программе установки может потребоваться получить много пакетов из сети. Если у вас медленное соединение с интернет, то это займёт много времени. После начала установки пакетов её никак нельзя прервать.

Даже если пакеты есть на CD-ROM, программа установки всё равно может скачивать пакеты с зеркала, если на зеркале есть более новая версия пакета, чем на CD-ROM. Если вы устанавливаете стабильную версию дистрибутива, такое может происходить после обновления выпуска (обновление первого стабильного выпуска); если вы устанавливаете тестируемую версию, то это может происходить, если у вас устаревший образ.

### 6.3.6 Установка и настройка системного загрузчика

Если вы выполняете установку на бездисктовую рабочую станцию, очевидно, что загрузка с локального диска не нужна, и этот шаг можно пропустить.

#### 6.3.6.1 Обнаружение операционных систем

Перед установкой системного загрузчика программа установки попытается обнаружить другие операционные системы, установленные на машине. Если она обнаружит поддерживаемую операционную систему, то это будет отражено на этапе установки системного загрузчика, и компьютер сможет загружать эту операционную систему в дополнении к Debian.

Надо заметить, что загрузка множества операционных систем на одной машине всё ещё напоминает чёрную магию. Автоматическая поддержка обнаружения и настройки системных загрузчиков на загрузку различных операционных систем различается на разных архитектурах и даже субархитектурах. Более подробную информацию можно найти в документации по менеджеру загрузки.

### 6.3.6.2 Настройка системы на загрузку с помощью flash-kernel

Так как не существует единого интерфейса микропрограммы для платформы ARM, шаги для создания загрузаемой системы на каждом устройстве ARM различаются. Для этого Debian использует инструмент **flash-kernel**. Flash-kernel содержит базу данных различных устройств, которая описывает операции, требуемые для создания загрузаемой системы. Он определяет, поддерживается ли имеющееся устройство, и если да, то выполняет необходимые действия.

На устройствах, загрузаемых с внутренней памяти NOR- или NAND-flash, flash-kernel записывает ядро и начальный ramdisk в эту внутреннюю память. Данный метод распространён среди старых устройств armel. Отметим, что большинство этих устройств не может содержать несколько ядер и ramdisk-ов в своей внутренней памяти, т. е. запуск flash-kernel, обычно, перезаписывает содержимое памяти!

Для систем ARM, использующих U-Boot в качестве системной микропрограммы и загружающих ядро и начальный ramdisk с внешнего носителя информации (например карты MMC/SD, USB или жёсткие диски IDE/SATA), flash-kernel генерирует соответствующий сценарий загрузки, который позволяет выполнить автозагрузку без действий пользователя.

### 6.3.6.3 Продолжение без системного загрузчика

Эта опция может быть использована для завершения установки без установки системного загрузчика, если для этой архитектуры/субархитектуры его не существует или потому что, он вам не нужен (например, чтобы использовать существующий системный загрузчик).

Если вы планируете настройку системного загрузчика вручную, то должны узнать имя файла устанавливаемого ядра, которое лежит в каталоге `/target/boot`. Также, проверьте существование каталога `initrd`; если он есть, то вероятно нужно указать это в системном загрузчике, чтобы он использовал его. Ещё вам потребуется знать диск и раздел, который выбран в качестве / файловой системы и, если вы выбрали установку с отдельным разделом `/boot`, то также и раздел файловой системы `c /boot`.

## 6.3.7 Завершение установки

Это последний этап установки Debian в процессе которого программа установки выполнит ещё пару действий. В основном, это уборка за `debian-installer`.

### 6.3.7.1 Настройка системного времени

Программа установки может спросить вас показывают ли часы компьютера время по Гринвичу (UTC). Обычно этот вопрос, если возможно, не задаётся и программа установки решает, что часы показывают время по Гринвичу, если на машине не установлено других операционных систем.

В экспертном режиме вы всегда можете выбрать установлены ли часы по Гринвичу или нет.

На этом этапе `debian-installer` попытается сохранить текущее время в аппаратных часах системы. Оно будет выставлено в UTC или местном времени, в зависимости от выбора, который вы недавно делали.

### 6.3.7.2 Перезагрузка системы

Вам предложат вынуть загрузочный носитель (CD, дискету, другой), который использовался для загрузки программы установки. После этого компьютер будет перезагружен в новую систему Debian.

## 6.3.8 Устранение проблем

Перечисленные в этом разделе компоненты обычно не используются в процессе установки, но они готовы прийти на помощь пользователю в случае, если что-то пойдёт не так.

### 6.3.8.1 Сохранение протокола установки

Если установка прошла успешно, файлы протокола установки, созданные во время процесса установки, будут автоматически записаны в `/var/log/installer/` в новой системе Debian.

Выбор Сохранить файлы журнала с отладочной информацией из главного меню позволяет записать файлы протокола на дискету, через сеть, на жёсткий диск или другой носитель. Это может быть полезно, если вы обнаружили неустранимую проблему при установке и хотите изучить протокол на другой системе или послать его вместе с отчётом об ошибке.



### 6.3.8.2 Использование командной строки и просмотр журнальных файлов

Есть несколько способов перейти в оболочку командной строки во время выполнения установки. На большинстве систем, и если вы не производите установку с консоли на последовательном порту, самый простой метод — это переключиться на вторую *виртуальную консоль* нажав левый Alt-F2<sup>10</sup> (на клавиатуре Mac: Option-F2). Чтобы вернуться обратно в программу установки используйте левый Alt-F1.

Если вы не можете переключаться между консолями, также есть пункт в меню Запуск оболочки, который запускает оболочку. Чтобы из большинства диалогов попасть обратно в главное меню нажмите кнопку Go Back один или несколько раз. Чтобы вернуться обратно в программу установки, введите **exit** для завершения оболочки.

В данный момент вы загрузились с RAM диска и имеете ограниченный набор Unix утилит. Вы можете выяснить какие утилиты доступны с помощью команды **ls /bin /sbin /usr/bin /usr/sbin** или набрав **help**. Клон Bourne оболочки командной строки называется **ash** и имеет некоторые полезные свойства, такие как автодополнение и история команд.

Для редактирования и просмотра файлов имеется текстовый редактор **nano**. Журнальные файлы системы установки можно найти в каталоге `/var/log`.

#### Замечание



Хотя вы и можете делать в оболочке в помощью команд всё что угодно, не забывайте, что оболочка, прежде всего, предназначалась для отладки и если что-то заработает неправильно.

Выполнение команд в оболочке может повлиять на процесс установки и привести к ошибкам или неполной установке. В частности, вы всегда должны позволить программе установки активировать раздел подкачки и не выполнять это самостоятельно из оболочки командной строки.

### 6.3.9 Установка по сети

Одной из наиболее интересных компонент является *network-console*. Она позволяет выполнить огромную часть процесса установки через сеть по протоколу SSH. Чтобы использовать сеть, вам нужно выполнить первые шаги установки из консоли, по крайней мере настройку сети. (Хотя вы можете автоматизировать и эту часть с помощью Раздел 4.4.)

По умолчанию данная компонента не загружается в главное меню установки, поэтому вам потребуется явно указать это. Если для установки используется CD, вам нужно загрузиться со средним приоритетом или как-то иначе получить главное меню установки и выбрать Загрузка компонент программы установки с компакт-диска и в списке дополнительных компонент выбрать `network-console: Continue installation remotely using SSH`. При успешной загрузке в меню появится новый пункт Продолжение установки через SSH.

Далее после выбора данного пункта попросят ввести новый пароль и его подтверждение для подключения к устанавливаемой системе. Это всё. Теперь вы должны увидеть инструкции по удалённому подключению в качестве пользователя *installer* с только что введённым паролем. Также здесь будет показан отпечаток (fingerprint) системы. Вам нужно передать этот отпечаток безопасным образом человеку, который будет продолжать установку удалённо.

Если вы решите продолжить установку локально, то всегда можете нажать клавишу **Enter** и вернуться обратно в меню чтобы выбрать другую компоненту.

Теперь давайте перейдём на другую сторону соединения. Здесь вам потребуется настроить терминал для работы с кодировкой UTF-8, потому что её использует устанавливаемая система. Если вы не можете этого сделать, удалённая установка всё ещё возможна, но в этом случае вы будете наблюдать на экране странные закорючки вместо границ окон и не-ascii символов. Чтобы установить соединение с удалённой системой просто наберите:

```
$ ssh -l installer install_host
```

где *install\_host* это имя или IP-адрес компьютера с запущенной программой установки. Перед приглашением в систему будет показан идентификатор машины (fingerprint), чтобы вы смогли убедиться, что это именно тот компьютер.

<sup>10</sup> То есть: нажмите клавишу **Alt** с левой стороны от клавиши **пробел** и функциональную клавишу **F2** одновременно.

**Замечание**

Сервер **ssh** из программы установки использует настройки по умолчанию, то есть не отправляет пакеты *keep-alive*. В принципе, соединение с системой, будучи установленным, должно сохраняться открытым бесконечно. Однако, в некоторых ситуациях — в зависимости настроек вашей локальной сети — соединение может быть потеряно после некоторого периода неактивности. Одним из случаев когда это случается — использование одной из форм трансляции сетевых адресов (NAT) где-то между клиентом и устанавливаемой системой. В зависимости от этапа установки, при котором было разорвано соединение, вы не всегда сможете продолжить установку после переподключения.

Чтобы избежать потери подключения, добавьте параметр `-o ServerAliveInterval=значение` при запуске соединения **ssh**, или добавьте этот параметр в свой файл настройки **ssh**. Заметим однако, что в некоторых случаях добавление этого параметра вообще *может приводить* к потере соединения (например, если пакеты *keep-alive* посылались во время краткосрочных проблем в сети, хотя без параметра **ssh** сам бы восстановился), поэтому используйте это только когда очень нужно.

**Замечание**

Если вы производите установку на несколько компьютеров последовательно и так получилось, что они имеют одинаковый IP-адрес или имя, то **ssh** не будет подключаться к таким машинам. Причина в том, что они имеют разные отпечатки, что обычно указывает на атаку подмены. Если вы уверены, что это не так, то вам нужно удалить соответствующую строку из `~/.ssh/known_hostsa` и попробовать подключиться ещё раз.

<sup>a</sup> Для удаления существующей записи об узле можно воспользоваться командой: `ssh-keygen -R <имя_узла|IP-адрес>`.

После регистрации на удалённой системе вам предложат два варианта: *Start menu* и *Start shell*. Выбрав первый, вы попадёте в главное меню программы установки, откуда можно продолжить установку как обычно. Во втором варианте запускается оболочка командной строки, из которой можно проверить и возможно, что-то исправить на удалённой системе. Разрешается запускать только одну сессию SSH для установочного меню и несколько для оболочки командной строки.

**Внимание**

После запуска установки через SSH, вы не должны возвращаться к установке с консоли, так как это может привести к порче базы данных, которая содержит настройки новой системы. Это, в свою очередь, может привести к неудачному завершению установки или проблемам в установленной системой.

## 6.4 Загрузка отсутствующих микропрограмм

Как описывалось в Раздел 2.2, в некоторые устройства требуется загрузить микропрограмму. В большинстве случаев, устройство вообще не будет работать при отсутствии микропрограммы; иногда в устройстве всё-же работают базовые функции, а микропрограмма нужна только для предоставления дополнительных возможностей.

Если драйвер устройства запрашивает микропрограмму, а она недоступна, то `debian-installer` выведет окно диалога, предлагающее загрузить отсутствующую микропрограмму. Если ответить утвердительно-

но, то `debian-installer` просканирует доступные устройства в поисках файлов или пакетов с нужной микропрограммой. Если файлы найдены, то микропрограмма будет скопирована в нужное место (`/lib/firmware`) и будет предпринята попытка перезагрузки модуля с драйвером.

#### Замечание



Список сканируемых устройств и файловых систем зависит от архитектуры, метода установки и этапа установки. На самых первоначальных этапах установки успешная загрузка микропрограммы наиболее вероятна с дискет или USB-носителей с файловой системой FAT.

Заметим, что можно пропустить загрузку микропрограммы, если вы уверены, что устройство будет работать и без неё, или если устройство не требуется во время установки.

`debian-installer` запрашивает микропрограммы только к модулям ядра, которые необходимы для загрузки на время установки. Не все драйверы включены в `debian-installer`, в частности, отсутствует `gadeon`, из чего следует, что возможности некоторых устройств могут не отличаться в конце установки от того, какими они были в начале. Другими словами, возможности некоторых аппаратных компонентов могут использоваться не полностью. Если вы подозреваете, что это ваш случай, или просто из любопытства, проверьте вывод команды `dmesg` в только что загруженной системе и поищите «микропрограмму».

### 6.4.1 Подготовка носителя

В официальных образах CD не содержатся несвободные микропрограммы. Самым распространённым способом загрузки этих микропрограмм — с какого-нибудь сменного носителя, например, с карты памяти USB. Также существуют неофициальные сборки CD с микропрограммами, которые можно найти в <http://cdimage.debian.org/cdimage/unofficial/non-free/cd-including-firmware/>. Чтобы подготовить карту памяти USB (или другой носитель, например, раздел жёсткого диска или дискету) файлы или пакеты с микропрограммами должны быть размещены в корневом каталоге или каталоге с именем `/firmware` в файловой системе носителя. Рекомендуется использовать файловую систему FAT, так как она, скорее всего, поддерживается на самых ранних этапах установки.

Тар-архивы и файлы `zip` с пакетами наиболее распространённых микропрограмм доступны здесь:

- <http://cdimage.debian.org/cdimage/unofficial/non-free/firmware/>

Просто скачайте `tar`-архив или `zip`-файл под нужный выпуск и распакуйте его на файловую систему носителя.

Если требуемой микропрограммы нет в `tar`-архиве, то также вы можете скачать нужные пакеты с микропрограммами из архива (из раздела `non-free`). Вот список наиболее скачиваемых пакетов микропрограмм, но он может быть не полон и также может содержать пакеты без микропрограмм:

- <http://packages.debian.org/search?keywords=firmware>

Также, можно копировать на носитель отдельные файлы микропрограммы. Такие микропрограммы можно получить, например, из уже установленной системы или от поставщика оборудования.

### 6.4.2 Микропрограмма в установленной системе

Все микропрограммы, загруженные во время установки, будут автоматически скопированы в установленную систему. В большинстве случаев это гарантирует, что устройство, которому требуется микропрограмма, будет также работать правильно после перезагрузки компьютера в установленную систему. Однако, если установленная система работает на версии ядра, отличной от использованного в программе установки, есть небольшая вероятность, что микропрограмма не загрузится из-за изменений ядра.

Если микропрограмма загружена из пакета, то `debian-installer` также установит этот пакет в устанавливаемую систему и автоматически добавит раздел `non-free` архива пакетов в АРТ-файл `sources.list`. Это позволит автоматически обновлять микропрограмму при появлении новой версии.

Если загрузка микропрограммы была пропущена во время установки, то соответствующее устройство, вероятно, не заработает в установленной системе пока микропрограмма (пакет) не будет установлена вручную.

**Замечание**

Если микропрограмма была загружена из отдельных файлов, то скопированная в установленную систему микропрограмма *не* будет автоматически обновляться, пока не не будет установлен соответствующий пакет с микропрограммой (если есть) после завершения установки.

## Глава 7

# Загрузка только что установленной системы Debian

### 7.1 Момент истины

Первую самостоятельную загрузку системы инженеры-электрики называют «smoke test».

Если система не загружается, не волнуйтесь. Если установка завершилась успешно, то это всего лишь результат какой-то несущественной проблемы, которая не позволяет системе загрузить Debian. В большинстве случаев такие проблемы решаются без повторной установки. Одним из методов решения проблем с загрузкой является встроенный в программу установки режим восстановления (Раздел 8.7).

Если вы новичок в Debian и Linux, вам может потребоваться помощь более опытных пользователей. Для менее распространённых архитектур типа 32-bit hard-float ARMv7, лучше задать вопрос в [список рассылки debian-arm](#). Также вы можете отправить отчёт об установке как это описано в Раздел 5.4.5. Описывайте проблему как можно понятнее, добавьте все сообщения которые отображались, они могут помочь определить причину проблемы.

### 7.2 Монтирование зашифрованных томов

Если вы создали зашифрованные тома во время установки и назначили им точки монтирования, то во время загрузки машины вас попросят ввести ключевую фразу на каждый том.

Для разделов, зашифрованных dm-crypt, во время загрузки вы увидите следующее сообщение:

```
Starting early crypto disks... part_crypt(starting)
Enter LUKS passphrase:
```

В первой строке приглашения, вместо *part* будет указано название используемого раздела, например *sda2* или *md0*. Вы, вероятно, задумаетесь: *для какого тома* нужно вводить ключевую фразу? Для */home*? Или для */var*? Естественно, если у вас только один зашифрованный том, это легко и вы можете просто ввести ключевую фразу, которую задали при настройке тома. Если же у вас несколько зашифрованных томов, то пригодятся заметки, которые вы сделали на последнем этапе Раздел 6.3.3.6. Если вы ранее не записали соответствие между *part\_crypt* и точкой монтирования, то можете выяснить это в файлах */etc/crypttab* и */etc/fstab* на новой системе.

При монтировании зашифрованного корневого раздела приглашение может выглядеть несколько иначе. Это зависит от генератора *initramfs*, который использовался при создании *initrd* для загрузки системы. Пример для *initrd*, созданного с помощью *initramfs-tools*:

```
Begin: Mounting root file system... ..
Begin: Running /scripts/local-top ...
Enter LUKS passphrase:
```

Во время ввода ключевой фразы символы не отображаются (даже звёздочками). Если вы введёте неверную ключевую фразу, то у вас ещё есть две попытки. После третьей попытки процесс загрузки пропустит этот том и начнётся монтирование следующей файловой системы. Подробнее смотрите в Раздел 7.2.1.

После ввода всех ключевых фраз загрузка продолжится как обычно.

### 7.2.1 Устранение неполадок

Если некоторые зашифрованные тома не были смонтированы из-за ввода неправильной ключевой фразы, их можно смонтировать вручную после загрузки. Но есть несколько случаев.

- Первый случай касается корневого раздела. Если он не смонтирован, то процесс загрузки остановится, а вам придётся перезагрузить компьютер, чтобы попробовать ещё раз.
- Самый простой случай — зашифрованные тома с данными `/home` или `/srv`. Вы можете просто смонтировать их вручную после загрузки машины.

Для `dm-crypt` немного сложнее. Во-первых, вам нужно зарегистрировать тома с помощью `device mapper` — выполнив:

```
# /etc/init.d/cryptdisks start
```

Эта операция просканирует все тома, указанные в `/etc/crypttab`, и создаст соответствующие устройства в каталоге `/dev` после ввода правильных ключевых фраз (тома, зарегистрированные ранее, будут пропущены, поэтому вы можете спокойно запускать эту команду несколько раз). После успешной регистрации вы можете смонтировать тома как обычно:

```
# mount /точка_монтирования
```

- Если на любых несмонтированных томах содержатся некритичные системные файлы (`/usr` или `/var`), система должна загрузиться и вы сможете смонтировать тома вручную как в предыдущем случае. Однако, может потребоваться (пере)запуск каждой службы, которая обычно запускается на уровне выполнения по умолчанию, так как, скорее всего, они не смогли запуститься. Самый простой способ сделать это — просто перезагрузить компьютер.

## 7.3 Вход в систему

После загрузки системы будет показано приглашение к входу в систему. Для входа используйте персональный логин и пароль, которые вы указали во время процесса установки. Система готова к работе.

Если вы новичок, то начните с просмотра документации, которая уже установлена в системе. Есть несколько систем документации, продолжается работа над их объединением. Вот несколько мест, откуда можно начать.

Документация на установленные программы находится в `/usr/share/doc/` в подкаталоге с именем этой программы (точнее, с именем пакета Debian, в котором содержится эта программа). Однако более полная документация часто упакована отдельно в специальные пакеты с документацией, которые по умолчанию не устанавливаются. Например, документацию на программу управления пакетами **apt** можно найти в пакетах `apt-doc` или `apt-howto`.

Также, `/usr/share/doc/` содержит несколько специальных каталогов. Linux HOWTO установлены в формате `.gz` в каталог `/usr/share/doc/HOWTO/en-txt/`. После установки **dhhelp** появляется файл `/usr/share/doc/HTML/index.html` со ссылками на документацию.

Простой способ просмотреть эту документацию — запустить текстовый браузер с помощью команд:

```
$ cd /usr/share/doc/
$ w3m .
```

Точка после команды **w3m** служит для показа содержимого текущего каталога.

Если у вас установлено графическое окружение рабочего стола, то также вы можете использовать веб-браузер. Запустите веб-браузер из меню приложений и введите `/usr/share/doc/` в строке адреса.

Также вы можете набрать **info команда** или **man команда**, чтобы получить документацию по большинству команд доступных в командной строке. Набрав **help**, вы увидите помощь по командам интерпретатора командной строки. А при вводе имени команды с параметром **--help** обычно выдаётся короткая справка по использованию программы. Если начало результата выполнения команды уехало за пределы экрана, добавьте **| more** после команды, и это включит паузу при заполнении всего экрана. Чтобы увидеть список всех доступных команд, начинающихся с определённого символа, введите символ и нажмите клавишу табуляции два раза.

## Глава 8

# Следующие шаги и что делать дальше

### 8.1 Выключение системы

Чтобы выключить работающую Debian GNU/Linux систему, вы не должны использовать кнопку сброса на передней панели корпуса компьютера или просто выключать электропитание. Debian GNU/Linux нужно выключать правильно, иначе можно потерять файлы и/или повредить жёсткий диск. Если вы работаете в окружении рабочего стола, то обычно есть возможность «завершить сеанс» из меню приложений, которая позволяет выключить (или перезагрузить) систему.

Alternatively you can press the key combination `Ctrl-Alt-Del`. If the key combinations do not work, a last option is to log in as root and type the necessary commands. Use **reboot** to reboot the system. Use **halt** to halt the system without powering it off <sup>1</sup>. To power off the machine, use **poweroff** or **shutdown -h now**. The systemd init system provides additional commands that perform the same functions; for example **systemctl reboot** or **systemctl poweroff**.

### 8.2 Если вы новичок в Unix

Если вы новичок в Unix, то, вероятно, вам стоит купить несколько книг и что-нибудь почитать. Много полезной информации можно найти в [справочнике по Debian](#). В [списке Unix FAQs](#) содержатся ссылки на некоторые интересные документы UseNet.

Linux — это одна из реализаций Unix. [Linux Documentation Project \(LDP\)](#) собирает различные HOWTO и электронные книги, посвящённые Linux. Большинство из этих документов можно получить просто установив пакет `doc-linux-html` (HTML версия) или пакет `doc-linux-text` (ASCII версия), а затем перейти в каталог `/usr/share/doc/HOWTO`. Переводы LDP HOWTO также доступны в виде пакетов Debian.

### 8.3 Самостоятельное ориентирование в Debian

Debian слегка отличается от других дистрибутивов. Даже если вы знакомы с другими дистрибутивами Linux, есть вещи, которые вы должны знать о Debian, чтобы поддерживать систему в рабочем состоянии. Эта глава содержит материалы, которые помогут вам ориентироваться; она не является руководством по использованию Debian, она всего лишь кратко описывает систему для занятых людей.

#### 8.3.1 Система управления пакетами Debian

Наиболее важно разобраться в системе управления пакетами Debian. В сущности, большая часть системы находится под контролем системы управления пакетами. Сюда входят:

- `/usr` (кроме `/usr/local`)
- `/var` (вы можете сделать `/var/local`, там меняться ничего не будет)
- `/bin`
- `/sbin`

---

<sup>1</sup> Under the SysV init system **halt** had the same effect as **poweroff**, but with systemd as init system (the default since jessie) their effects are different.

- /lib

Например, если вы замените /usr/bin/perl, он будет работать, но позже, если вы обновите пакет perl, файл, который вы положили будет заменён. Эксперты могут обойти это, установив на пакеты флаг «hold» в **aptitude**.

One of the best installation methods is apt. You can use the command line version of **apt** or full-screen text version **aptitude**. Note apt will also let you merge main, contrib, and non-free so you can have export-restricted packages as well as standard versions.

### 8.3.2 Дополнительное программное обеспечение для Debian

Есть официальные и неофициальные репозитории ПО, которые не включены в установку Debian по умолчанию. В них содержится ПО, которое многие считают полезным и хотели бы его иметь. Информацию о дополнительных репозиториях можно найти на вики-странице Debian [The Software Available for Debian's Stable Release](#).

### 8.3.3 Управление версиями приложений

Альтернативность версий приложений управляется с помощью программы **update-alternatives**. Если вы работаете с несколькими версиями одного приложения, прочитайте страницу руководства к **update-alternatives**.

### 8.3.4 Управление заданиями cron

Любые работы относящиеся к администрированию системы должны проводиться в каталоге /etc, так как там расположены конфигурационные файлы. Если у вас есть ежедневные, еженедельные или ночные задания cron для суперпользователя, поместите их в каталог /etc/cron.{daily, weekly, monthly}. Они вызываются из /etc/crontab и запускаются в алфавитном порядке.

С другой стороны, если задания: (а) нужно запускать от определённого пользователя или (б) нужно запустить в определённое время или с определённой частотой, вы можете использовать или /etc/crontab, или, что лучше, /etc/cron.d/имя\_задания. Это особые файлы, они содержат дополнительное поле, позволяющее вам указать учётную запись пользователя, от которого будет выполняться задание cron.

В любом случае, просто отредактируйте файлы, и cron обнаружит эти изменения автоматически. Не нужно запускать специальную команду. Подробная информация доступна в **cron(8)**, **crontab(5)** и /usr/share/doc/cron/README.Debian.

## 8.4 Что почитать

Если вам требуется информация по определённой программе, сначала, попробуйте выполнить **man название программы** или **info название программы**.

Также, есть много полезной документации в каталоге /usr/share/doc. В частности, в каталогах /usr/share/doc/HOWTO и /usr/share/doc/FAQ содержится море интересных сведений. Информацию об ошибках смотрите в /usr/share/doc/debian/bug\*. Отличия в работе определённой программы, касающиеся Debian, смотрите в файле /usr/share/doc/(имя пакета)/README.Debian.

[Веб-сайт Debian](#) содержит огромное количество документации по Debian. В частности, посмотрите [Debian GNU/Linux FAQ](#) и [Справочник по Debian](#). Каталог документации Debian находится по адресу [Проект Документации Debian](#). Можно обратиться с вопросом к сообществу Debian подписавшись на один или более списков рассылки Debian [Подписка на списки рассылки](#). Также, не последнее место занимают [Архивы списков рассылки Debian](#), содержащие ценную информацию о Debian.

Общая информация о GNU/Linux есть в [проекте документации Linux](#). Здесь вы найдёте разные HOWTO и ссылки на другие превосходные источники информации на компоненты системы GNU/Linux.

## 8.5 Настройка почты

Сегодня электронная почта (email) — это очень важная часть жизни многих людей. Так как при настройке почты используется много параметров, а правильность настройки влияет на многие программы в Debian, в этом разделе мы попытаемся описать основные моменты настройки.

Почтовая система состоит из трёх частей. Первая: *пользовательский почтовый агент [Mail User Agent] (MUA)* — это программа, с которой работает пользователь для написания и чтения писем. Вторая: *агент*



пересылки почты [*Mail Transfer Agent*] (MTA) — отвечает за пересылку сообщений между компьютерами. Третья: *агент доставки почты* [*Mail Delivery Agent*] (MDA) — отвечает за доставку входящей почты в почтовый ящик пользователя.

Данные части могут выполняться разными программами, но могут быть и объединены в одной или двух программах. Также можно настроить, чтобы разные виды почты обрабатывались разными программами.

Исторически, в системах Linux и Unix очень популярным MUA является **mutt**. Как и большинство классических программ Linux он имеет текстовый интерфейс. Часто он используется в связке с **exim** или **sendmail**, работающих в качестве MTA и **procmail** в качестве MDA.

With the increasing popularity of graphical desktop systems, the use of graphical e-mail programs like GNOME's **evolution**, KDE's **kmail** or Mozilla's **thunderbird** has becoming more popular. These programs combine the function of a MUA, MTA and MDA, but can — and often are — also be used in combination with the traditional Linux tools.

### 8.5.1 Почтовая конфигурация по умолчанию

Даже если вы планируете использовать почтовую программу с графическим интерфейсом, очень важно, чтобы в системе Debian GNU/Linux также были установлены и правильно настроены классические MTA/MDA. Они нужны для различных утилит, которые работают в системе<sup>2</sup>, чтобы они могли послать важные уведомления по почте системному администратору о (потенциальных) проблемах или изменениях.

По этой причине по умолчанию устанавливаются пакеты **exim4** и **mutt** (если вы не отменили задачу «стандарт» во время установки). **exim4** — это комбинация MTA/MDA, имеет относительно небольшой размер, но очень гибок в настройке. По умолчанию, он настраивается на обработку только локальной почты самой системы, а почта, предназначенная системному администратору (учётная запись **root**), будет доставлена на учётную запись обычного пользователя, созданного во время установки<sup>3</sup>.

При доставке почты она добавляется в файл `/var/mail/учётная_запись`. Почту можно читать с помощью **mutt**.

### 8.5.2 Отправка почты из системы

Как было упомянуто ранее, установленная система Debian настроена для обработки только локальной почты, она не может отправлять или принимать почту от других систем.

Если вы хотите, чтобы **exim4** обрабатывал внешнюю почту, перейдите в следующий подраздел, в котором описаны основные параметры настройки. Обязательно протестируйте приём и отправку почтовых сообщений.

Если вы будете использовать почтовую программу с графическим интерфейсом с почтовым сервером вашего провайдера интернет (ISP) или вашей компании, то необязательно настраивать **exim4** для обработки внешней почты. Просто укажите в любимой почтовой программе с графическим интерфейсом нужные серверы для отправки и приёма сообщений (не описано в данном руководстве).

Однако, в этом случае вам потребуется правильно настроить отдельные утилиты, чтобы они могли отправлять почту. Одной из таких утилит является **reportbug**, программа для отправки сообщений об ошибках в пакетах Debian. По умолчанию, она настроена на использование **exim4**.

Чтобы правильно настроить **reportbug** на использование внешнего почтового сервера, запустите команду **reportbug --configure** и ответьте «нет» на вопрос о доступности MTA. Затем вас попросят указать сервер SMTP, который нужно использовать для отправки сообщений об ошибках.

### 8.5.3 Настройка агента пересылки почты Exim4

Если вы хотите, чтобы система также работала с внешней почтой, то вам нужно перенастроить пакет **exim4**<sup>4</sup>:

```
# dpkg-reconfigure exim4-config
```

После ввода этой команды (имея права суперпользователя), вас спросят, хотите ли вы разделить конфигурацию на маленькие файлы. Если не знаете что делать, выберите установку по умолчанию.

Затем вам будет предложено несколько общих почтовых сценариев. Выберите тот, который подходит вам больше всего.

<sup>2</sup> Примеры: **cron**, **quota**, **logcheck**, **aide**, ...

<sup>3</sup> Пересылка почты **root** на учётную запись обычного пользователя настраивается в файле `/etc/aliases`. Если учётная запись обычного пользователя не была создана, почта, естественно, будет направляться самой учётной записи **root**.

<sup>4</sup> Конечно, также вы можете удалить **exim4** и заменить его другим MTA/MDA.

**интернет-сайт** Ваша система подключена к сети, почта отправляется и получается напрямую по протоколу SMTP. На следующих экранах вам зададут несколько основных вопросов, таких как почтовое имя вашей машины или список доменов, для которых вы принимаете или передаёте почту.

**отправка почты через компьютер-шлюз (smarthost)** По этому сценарию ваша исходящая почта переправляется на другую машину, называемую «smarthost», которая и отправит письмо по назначению. Smarthost обычно ещё и хранит входящую почту, адресованную вашему компьютеру, так что вам не надо постоянно находиться в сети. Это также означает, что вам нужно загружать почту со smarthost с помощью программ типа fetchmail.

В большинстве случаев, в качестве smarthost используется почтовый сервер вашего ISP, что очень подходит пользователям, которые используют модемное подключение по телефонной линии. Также это может быть почтовый сервер компании или другая система в вашей сети.

**отправка почты через smarthost; локальная почта отсутствует** То же самое, что и предыдущее значение, кроме того, что система не будет настроена на обработку почты локального домена. Почта от самой системы (например, для системного администратора) всё равно будет обработана.

**доставка только локальной почты** Настроено в системе по умолчанию.

**в данный момент конфигурация отсутствует** Выберите этот вариант, если вы абсолютно уверены в том, что делаете. Это оставит вас с ненастроенной почтовой системой; до тех пор пока вы не настроите её, вы не сможете отправлять или получать почту и вы можете пропустить некоторые важные сообщения от ваших системных утилит.

Если ни один из этих сценариев не подходит, или нужна более тонкая настройка, то после завершения установки вам нужно отредактировать файлы конфигурации в каталоге `/etc/exim4`. Дополнительную информацию про `exim4` можно найти в каталоге `/usr/share/doc/exim4`; в файле `README.Debian.gz` содержатся дополнительные подробности о настройке `exim4` и указано где найти дополнительную документацию.

Заметим, что если у вас нет официального доменного имени и вы отправляете почту в интернет напрямую, это может привести к тому, что ваша почта не будет приниматься серверами, так как на них установлены средства борьбы со спамом. Предпочтительней использовать почтовый сервер ISP. Если вы всё равно хотите отправлять почту напрямую, то вам нужно использовать другой почтовый адрес, а не сгенерированный по умолчанию. Если в качестве MTA используется `exim4`, это это делается добавлением записи в файл `/etc/email-addresses`.

## 8.6 Сборка нового ядра

Зачем пересобирать ядро? Обычно этого делать не нужно, так как ядро, устанавливаемое по умолчанию в Debian, подходит для многих конфигураций. Кроме этого, в Debian есть ещё несколько сборок ядра. Поэтому, сначала проверьте, может другой пакет с ядром лучше подходит для вашего компьютера. Однако, полезно пересобрать ядро в следующих случаях:

- если установлено специфичное оборудование или возникает конфликт аппаратного обеспечения со стандартным ядром
- чтобы задействовать свойства, которых нет в поставляемых сборках ядра (например, поддержка `high memory`)
- для оптимизации ядра, удаляя ненужные драйверы для уменьшения времени загрузки
- создания монолитного ядра, без модулей
- для установки обновлённого или разрабатываемого ядра
- чтобы больше узнать о ядрах linux

### 8.6.1 Управление образами ядра

Не бойтесь пересобирать ядро. Это забавно и полезно.

Чтобы собрать ядро по технологии Debian, вам потребуются некоторые пакеты: `fakeroot`, `kernel-package`, `linux-source-2.6` и некоторые другие, которые, вероятно, уже установлены (в `/usr/share/doc/kernel-package/README.gz` есть полный список).

С помощью этого метода создаётся `.deb` файл из исходных текстов ядра, а также, если у вас имеются нестандартные модули, делается `.deb` файл с синхронизированными зависимостями. Это наилучший способ управления образами ядра; в `/boot` лежит ядро, `System.map` и журнальный конфигурационный файл от сборки.

Заметим, что вам *не обязательно* собирать ядро «по технологии Debian»; но мы пришли к выводу, что управлять ядрами с помощью системы управления пакетами на самом деле безопаснее и легче. Фактически, вы можете получить исходные тексты ядра прямо у Линуса, а не в виде `linux-source-2.6`, и собрать ядро с помощью `kernel-package`.

Заметим, что полную документацию по использованию `kernel-package` можно найти в `/usr/share/doc/kernel-package`. Этот раздел содержит лишь краткое руководство.

Далее мы предполагаем, что вы можете делать на своей машине что угодно и распаковываете исходные тексты ядра в домашний каталог<sup>5</sup>. Также мы предполагаем, что у вас ядро версии 3.16. Проверьте, что вы находитесь в каталоге куда хотите распаковать исходные тексты ядра, распакуйте их с помощью команды `tar xf /usr/src/linux-source-3.16.tar.xz` и перейдите в только что появившийся каталог `linux-source-3.16`.

Теперь вы можете выполнить настройку ядра. Если установлены, настроены и запущены X11, то запустите `make xconfig`, иначе `make menuconfig` (вам потребуется установить пакет `libncurses5-dev`). Внимательно прочитайте интерактивную документацию и внимательно выберите нужные параметры. Если что-то непонятно, обычно лучше включить драйвер устройства (программное обеспечение, которое управляет установленным оборудованием, например, картами Ethernet, SCSI-контроллерами и так далее). Будьте осторожны: для других параметров, не относящихся к определённому аппаратному обеспечению, нужно оставить значения по умолчанию, если вы не понимаете, что они делают. Не забудьте выбрать «Kernel module loader» в «Loadable module support» (не включён по умолчанию). Если его не включить, возникнут проблемы с установкой Debian.

Очистите дерево исходных текстов и сбросьте параметры `kernel-package`. Для этого выполните `make-kpkg clean`.

Теперь соберите ядро: `fakeroot make-kpkg --initrd --revision=1.0.custom kernel_image`. Номер версии «1.0» можно изменить; этот номер версии нужен только для отслеживания ваших сборок. Более того, вы можете добавить любое слово вместо «custom» (например, имя машины). Сборка ядра может занять некоторое время в зависимости от мощности машины.

После завершения сборки вы можете установить собранное ядро как обычный пакет. От суперпользователя выполните `dpkg -i ../linux-image-3.16-субархитектура_1.0.custom_armhf.deb`. Часть *субархитектура* — это необязательная субархитектура, зависящая от установленных параметров ядра. `dpkg -i` установит ядро вместе с некоторыми полезными файлами поддержки. Например, будет правильно установлен `System.map` (полезен при отладке проблем с ядром), а также `/boot/config-3.16`, содержащий параметры сборки ядра. Также, пакет способен автоматически использовать системный загрузчик платформы для обновления загрузки. Если вы создали пакет с модулями, то вам нужно установить и его.

Настало время перезагрузить систему: внимательно прочитайте все предупреждения, которые могут появиться, затем наберите `shutdown -r now`.

Подробнее о ядрах Debian и сборке ядра смотрите в [Debian Linux Kernel Handbook](#). Полная информация о `kernel-package` лежит в каталоге `/usr/share/doc/kernel-package`.

## 8.7 Восстановление неработающей системы

Иногда что-то идёт неправильно, и система, которую вы аккуратно установили больше не загружается. Возможно, испортились настройки системного загрузчика во время изменения, или не хочет загружаться новое ядро, или космические лучи попали в жёсткий диск и исправили единичку на нолик в файле `/sbin/init`. В любом случае, вам нужна система с которой можно всё исправить, и режим восстановления может помочь в этом.

<sup>5</sup> Есть несколько мест куда можно распаковывать исходные тексты и собрать своё ядро, но это самый лёгкий способ без необходимости раздачи специальных прав.

To access rescue mode, select **rescue** from the boot menu, type **rescue** at the `boot :` prompt, or boot with the **rescue/enable=true** boot parameter. You'll be shown the first few screens of the installer, with a note in the corner of the display to indicate that this is rescue mode, not a full installation. Don't worry, your system is not about to be overwritten! Rescue mode simply takes advantage of the hardware detection facilities available in the installer to ensure that your disks, network devices, and so on are available to you while repairing your system.

Вместо программы разметки, вам покажут список разделов, имеющихся в системе, попросят выбрать один из них. Обычно, нужно выбрать раздел, содержащий корневую систему, которую нужно восстановить. Кроме созданных непосредственно на дисках, вы можете выбирать разделы на устройствах RAID и LVM.

Если возможно, программа установки запустит оболочку командной строки в выбранной файловой системе, где вы сможете провести любые дополнительные процедуры восстановления.

Иногда программе установки не удаётся запустить оболочку командной строки в выбранной корневой файловой системе. Возможной причиной может быть повреждение файловой системы, и в этом случае выдаётся предупреждение и предлагается запустить оболочку командной строки в среде программы установки. Здесь не так много полезных программ, но и их часто достаточно для восстановления системы. Выбранная корневая файловая система будет смонтирована в каталог `/target`.

В обоих случаях, после выхода из оболочки командной строки система будет перезагружена.

И напоследок заметим, что восстановление нерабочей системы может оказаться очень трудным делом, и это руководство не пытается описать все возможные случаи, которые могут произойти и способы их решения. Если у вас есть проблемы — обратитесь к профессионалам.

# Приложение А

## Краткая инструкция по установке

Данный документ описывает установку Debian GNU/Linux buster на 32-bit hard-float ARMv7 («armhf») с помощью нового `debian-installer`. Это краткое описание процесса установки, в нём содержится вся информация, необходимая для большинства установок. В тех местах, где полезно прочитать более обширное описание, будет приведена ссылка на соответствующую главу этого документа.

### А.1 Предварительные сведения

`Debian-installer` пока находится в состоянии тестирования. Если вы обнаружите ошибки во время установки, обратитесь к Раздел 5.4.5, чтобы прочитать как сообщить о них. Если у вас есть вопросы, на которые нет ответа в этом документе, смело задавайте их в списке рассылки `debian-boot` (`debian-boot@lists.debian.org`) или в IRC-канале (`#debian-boot` в сети OFTC).

### А.2 Загрузка программы установки

Несколько ссылок на CD-образы есть на [домашней странице `debian-installer`](#). CD-образы с `debian-installer`, подготовленные командой `debian-cd` можно найти на [странице Debian CD](#). Более подробно об этом написано в Раздел 4.1.

Для некоторых методов установки требуются дополнительные образы помимо CD-образа. На [домашней странице `debian-installer`](#) находятся ссылки на такие образы. В Раздел 4.2.1 описано, как их найти на серверах-зеркала Debian.

В следующих подразделах указано, какие образы требуются для каждого конкретного вида установки.

#### А.2.1 Загрузка с CDRROM

Образ CD `netinst` очень популярен для установки `buster` с помощью `debian-installer`. Он предназначен для загрузки с CD и установки дополнительных пакетов по сети, отсюда и название «`netinst`». В нём содержатся компоненты, необходимые для запуска программы установки и базовые пакеты, предоставляющие минимальную систему `buster`. При желании можно воспользоваться полным CD-образом, который не требует наличия сети для установки. Для этого достаточно первого диска из набора.

Загрузите любой понравившийся вам тип образа и запишите его на CD.

#### А.2.2 Загрузка по сети

`debian-installer` можно загрузить полностью из сети. Методы загрузки по сети (`netboot`) зависят от архитектуры и настройки сетевой загрузки. Файлы для загрузки `debian-installer` по сети лежат в каталоге `netboot/`.

#### А.2.3 Загрузка с жёсткого диска

Возможна загрузка программы установки не со сменного носителя, а просто с имеющегося жёсткого диска, который может содержать различные операционные системы. Скачайте файлы `hd-media/initrd.gz`, `hd-media/vmlinuz` и образ Debian CD в корневой каталоге жёсткого диска. Проверьте, что образ CD имеет имя файла, оканчивающееся на `.iso`. Теперь просто выполняется загрузка `linux` с `initrd`.

## А.3 Установка

После запуска программа установки выводит приветствие. Нажмите **Enter** для загрузки, или прочитайте инструкции про другие методы загрузки и параметры (смотрите Раздел 5.3).

Через некоторое время вас попросят выбрать язык. Используйте клавиши управления курсором для выбора языка и нажмите **Enter**, чтобы продолжить. Затем вас попросят выбрать страну из списка стран, где говорят на этом языке. Если её нет в коротком списке, то доступен список стран всего мира.

Вас могут попросить выбрать раскладку клавиатуры. Выберите предлагаемую по умолчанию, если не знаете какая лучше.

Теперь можно отдохнуть, пока `debian-installer` определяет необходимое аппаратное обеспечение вашей машины, и загружает оставшуюся часть самого себя с CD, дискеты, USB и т.д.

Затем программа установки попытается определить сетевое аппаратное обеспечение и настроить сеть с помощью DHCP. Если у вас нет сети или вы не имеете DHCP, то вам будет предложено настроить сеть вручную.

Затем настраиваются часовой пояс и время. Программа установки попытается подключиться к серверу времени в Интернет, чтобы проверить точность времени компьютера. Часовой пояс выбирается на основе ранее указанной вами страны, и программа установки попросит вас указать его явно только, если страна расположена в нескольких часовых поясах.

Setting up your clock and time zone is followed by the creation of user accounts. By default you are asked to provide a password for the «root» (administrator) account and information necessary to create one regular user account. If you do not specify a password for the «root» user, this account will be disabled but the **sudo** package will be installed later to enable administrative tasks to be carried out on the new system. By default, the first user created on the system will be allowed to use the **sudo** command to become root.

Теперь пришло время разметки дисков. Сначала вам предложат автоматическую разметку всего диска или только свободного пространства (см. Раздел 6.3.3.2). Этот режим рекомендуется для новичков или спешащих людей. Если вы не хотите авторазметку, выберите в меню пункт Вручную.

На следующем экране вы увидите текущую таблицу разделов, формат разделов и куда они будут монтироваться. Выберите раздел для изменения или удаления. Если вы выбрали автоматическую разметку, то вам доступен только пункт меню Закончить разметку и записать изменения на диск, нажав на который вы подтвердите изменения. Не забудьте назначить по крайней мере один раздел для подкачки и смонтировать один раздел в качестве корневого раздела `/`. В Раздел 6.3.3 подробно описано как использовать программу разметки; в Приложение С рассказано более подробно о процедуре разметки.

Далее `debian-installer` отформатирует разделы и начнёт установку базовой системы, что может занять некоторое время. Затем устанавливается ядро.

Основная часть системы, установленная ранее, уже работает, но умеет очень немного. Чтобы сделать систему более полезной, следующий этап позволяет установить дополнительные пакеты для определённых задач. Чтобы указать откуда получать пакеты, перед установкой пакетов нужно настроить `apt`. Задача «Стандартная система» выбрана по умолчанию и обычно нужна для установки. Выберите задачу «Окружение рабочего стола», если после установки хотите работать с графическим рабочим столом. Подробнее об этом этапе смотрите в Раздел 6.3.5.2.

На последнем шаге устанавливается системный загрузчик. Если программа установки обнаружит другую операционную систему на компьютере, то добавит её в меню загрузки и скажет вам об этом.

Далее `debian-installer` скажет, что установка завершена. Вытащите компакт-диск или другой носитель с программой установки и нажмите **Enter** для перезагрузки машины. После этого должна произойти загрузка в установленную систему и появиться приглашение к входу в систему. Это описано в Глава 7.

Если вам требуется дополнительная информация по процессу установки, посмотрите Глава 6.

## А.4 Отправьте нам отчёт об установке

If you successfully managed an installation with `debian-installer`, please take time to provide us with a report. The simplest way to do so is to install the `reportbug` package (**`apt install reportbug`**), configure `reportbug` as explained in Раздел 8.5.2, and run **`reportbug installation-reports`**.

Если завершить установку не получилось, то вероятно вы обнаружили ошибку в `debian-installer`. Для улучшения программы установки необходимо, чтобы мы узнали об этом, поэтому, пожалуйста, сообщите об ошибке. Вы можете использовать отчёт об установке, чтобы сообщить о проблеме; если установить вообще ничего не удалось, смотрите Раздел 5.4.4.

## **А.5 И напоследок...**

Мы надеемся, что установка Debian была несложной и что вы найдёте Debian полезным. Вероятно, вам пригодится информация из Глава 8.

# Приложение В

## Автоматическая установка с помощью списка ответов

В этом приложении описывается файл ответов на вопросы, который используется `debian-installer` для автоматизации установки операционной системы.

Фрагменты конфигурации, используемые в этом приложении, также доступны в примере файла ответов <http://d-i.alioth.debian.org/manual/example-preseed.txt>.

### В.1 Введение

Автоматическая установка позволяет автоматизировать ручной ввод ответов на вопросы программы установки. Это даёт возможность автоматизации наиболее типичных установок, а также предоставляет дополнительные возможности, недоступные во время обычной установки.

Автоматическая установка не является обязательной. Если вы укажете пустой файл ответов, то программа установки станет работать также, как при установке вручную. Каждый автоматический ответ на вопрос меняет (если поняли его правильно!) ход установки.

#### В.1.1 Методы использования файла ответов

Есть три метода использования для автоматизации: *initrd*, *файловый* и *сетевой*. Метод *initrd* работает при любом способе установки системы и поддерживает ответы на большинство вопросов, но он наиболее трудоёмок. Файловый и сетевой методы могут использоваться при различных способах установки.

В следующей таблице показано соответствие методов автоматической установки со способами установки операционной системы.

способ установки системы	initrd	файловый	сетевой
CD/DVD	да	да	да <sup>1</sup>
netboot	да	нет	да
hd-media	да	да	да <sup>1</sup>

An important difference between the preseeding methods is the point at which the preconfiguration file is loaded and processed. For *initrd* preseeding this is right at the start of the installation, before the first question is even asked. Preseeding from the kernel command line happens just after. It is thus possible to override configuration set in the *initrd* by editing the kernel command line (either in the bootloader configuration or manually at boot time for bootloaders that allow it). For file preseeding this is after the CD or CD image has been loaded. For network preseeding it is only after the network has been configured.

<sup>1</sup> только если есть подключение к сети и указан соответствующий `preseed/url`



**Важно**

Очевидно, что на любые вопросы, задаваемые перед загрузкой файла ответов, нельзя указать готовые ответы (сюда включаются вопросы, которые задаются только на среднем и низком уровне важности, например запуск определения аппаратуры). И не существует удобного способа избежать этих вопросов кроме указания ответов на них через параметры загрузки, как описано в Раздел **В.2.2**.



Чтобы просто избежать вопросов, которые обычно задаются перед тем как начнёт действовать автоматизация ответов, вы можете запустить программу установки в режиме «auto». Это позволит отложить вопросы, которые обычно задаются перед автоматической установкой (например, про язык, страну и выбор клавиатурной раскладки), до появления работоспособной сети. Также этот режим используется когда установка выполняется с критическим уровнем важности, что позволяет избежать многих неважных вопросов. Подробнее смотрите в Раздел **В.2.3**.

## В.1.2 Ограничения

Хотя для большинства вопросов `debian-installer` могут быть заданы предварительные ответы, есть несколько заметных исключений. Вы можете (пере)разметить только весь диск или использовать только доступное место на диске; невозможно задействовать существующие разделы.

## В.2 Использование автоматической установки

Во-первых, нужно создать файл ответов и сделать его доступным для использования. Создание файла ответов описывается позже в этом приложении. Разместить его в нужном месте довольно просто для сетевой автоматической установки и при чтении файла с дискеты или usb-носителя. Если нужно включить этот файл на CD или DVD, то придётся обновлять ISO образ. Здесь не будет описано как поместить файл ответов в `initrd`; обратитесь к документации разработчика `debian-installer`.

Пример файла ответов, который можно взять за основу доступен здесь <http://d-i.alioth.debian.org/manual/example-preseed.txt>. Данный файл основан на фрагментах конфигурации, включённых в это приложение.

### В.2.1 Загрузка файла ответов

Если используется метод `initrd`, то нужно убедиться, что файл с именем `preseed.cfg` лежит в корневом каталоге `initrd`. Программа установки автоматически проверяет наличие этого файла и загружает его.

Для других методов автоматической установки нужно указать при загрузке программе установки какой файл использовать при загрузке. Это можно сделать через параметр загрузки ядра, вручную во время загрузки или изменив файл настройки системного загрузчика (например, `syslinux.cfg`) и добавить параметр в конец строки `append` для ядра.

Если вы указываете файл ответов в настройке системного загрузчика, то можно изменить конфигурацию таким образом, чтобы не нажимать клавишу ввод для загрузки программы установки. Для `syslinux` это достигается установкой `timeout` равным `1` в файле `syslinux.cfg`.

Чтобы убедиться, что программа установки получила правильный файл ответов, можно дополнительно указать контрольную сумму файла. Это значение вычисляется с помощью `md5sum`, и если его указать, то оно должно соответствовать указываемому файлу, иначе программа установки не будет использовать данный файл.

Задаваемые параметры при загрузке:

```
- при загрузке из сети:
preseed/url=http://host/path/to/preseed.cfg
preseed/url/checksum=5da499872beccfed2c4872f9171c3d
- или
preseed/url=tftp://host/path/to/preseed.cfg
preseed/url/checksum=5da499872beccfed2c4872f9171c3d
```

```

- при загрузке с изменённого CD:
preseed/file=/cdrom/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d

- при установке с USB носителя (поместите файл ответов в
корневой каталог USB носителя):
preseed/file=/hd-media/preseed.cfg
preseed/file/checksum=5da499872becccfeda2c4872f9171c3d

```

Заметим, что `preseed/url` можно укоротить до `url`, `preseed/file` — до `file`, а `preseed/file/checksum` — от `preseed-md5`, если они передаются в параметрах загрузки.

## В.2.2 Использование параметров загрузки для автоматизации

Если файл ответов не может быть использован для автоматизации некоторых шагов, установку всё ещё можно сделать полностью автоматической, передав нужные значения ответов в командной строке запуска программы установки.

Параметры загрузки также можно использовать не только для автоматизации установки, а просто для того, чтобы указать ответ на определённый вопрос. Несколько примеров такого использования есть в разных местах этого руководства.

Чтобы установить значение, используемое внутри `debian-installer`, просто укажите *путь/до/переменная=значение* для любой автоматизируемой переменной, описанной в примерах в этом приложении. Если значение будет использоваться для настройки пакетов в устанавливаемой системе, то нужно перед переменной указывать *владельца*<sup>2</sup>, например, *владелец: путь/до/переменной=значение*. Если вы не указываете владельца, то значение переменной не копируется в базу данных `debconf` устанавливаемой системы и поэтому не будет использоваться во время настройки соответствующего пакета.

Обычно, автоматизация ответов приводит к тому, что вопрос задан не будет. Чтобы установить для вопроса ответ по умолчанию и оставить вопрос задаваемым, используйте в качестве оператора «?=>» вместо «=>». Смотрите также Раздел [В.5.2](#).

Заметим, что для некоторых переменных, которые часто задаются в приглашении загрузчика, имеются короткие названия. Если имеется короткое название, то оно используется в примерах в этом приложении вместо полного имени переменной. Например, переменная `preseed/url` сокращается до `url`. Ещё пример: псевдоним `tasks` раскрывается в `tasksel:tasksel/first`.

Обозначение «---» в параметрах загрузки имеет специальное значение. Параметры ядра, указанные после «---», могут быть скопированы в конфигурацию установленного системного загрузчика (если это умеет программа установки для загрузчика). Программа установки автоматически фильтрует все параметры (например, параметры предварительной настройки), которые распознаёт.

### Замечание



В настоящее время ядра `linux` версии (2.6.9 и новее) максимально могут принять 32 параметра в командной строке и 32 параметра через переменные окружения, считая все параметры, добавляемые программой установки по умолчанию. Если это число превышено, то ядро прекращает работу с ошибкой. (Для ранних версий ядер эти значения меньше.)

В большинстве установок некоторые параметры по умолчанию в файле конфигурации загрузчика, типа `vga=normal`, можно безопасно удалить, что может позволить вам добавить больше параметров автоматизации.

<sup>2</sup> Владелец переменной `debconf` (или шаблона) — это, обычно, имя пакета, в котором содержится соответствующий шаблон `debconf`. Переменными самой программы установки владеет «d-i». Шаблоны и переменные могут иметь более одного владельца, что влияет на их удаление из базы данных `debconf` при вычистке пакета.

## Замечание



Не всегда возможно задать значения с пробелами в параметрах загрузки, даже если заключить их в кавычки.

### В.2.3 Автоматический режим

Есть несколько возможностей программы установки Debian, объединив которые с помощью простых команд в приглашении к загрузке, можно получить автоматическую установку любого уровня настраиваемости.

Это включается с помощью выбора `Automated install` при загрузке, также называемом `auto` для некоторых архитектур или методов загрузки. В этом разделе `auto` означает не параметр, а выбор метода загрузки и добавляется в загрузочные параметры в приглашении к загрузке.

Вот один из примеров использования настроек в приглашении к загрузке:

```
auto url=autoserver
```

В этом случае для установки согласно полученным настройкам с DHCP сервера будет найдена машина `autoserver` из записей DNS, возможно к имени будет добавлено имя локального домена, если он передаётся в DHCP. Если в сети используется доменное имя `example.com`, и этот домен передаётся по DHCP, то в результате файл с ответами будет получен с `http://autoserver.example.com/d-i/buster./preseed.cfg`.

Последняя часть этого url (`d-i/buster./preseed.cfg`) берётся из `auto-install/defaultroot`. По умолчанию в него включён каталог `buster`, чтобы для будущих версий можно было задать своё название, что позволит людям контролировать процесс перехода на новую версию. Часть `./` используется для указания корня, относительно которого будут заданы остальные пути (для использования в `preseed/include` и `preseed/run`). Это позволяет указывать файлы с полным URL (пути, начинающиеся с `/`) или даже пути относительно места, где был найден последний файл ответов. Это может быть использовано для написания более переносимых сценариев, для того чтобы полное дерево сценариев можно было переместить в новое место ничего при этом не сломав, например скопировать файлы на USB-носитель, если ранее они брались с веб-сервера. Продолжая пример, если в файле ответов значение `preseed/run` заменить на `/scripts/late_command.sh`, то файл будет получен с `http://autoserver.example.com/d-i/buster./scripts/late_command.sh`.

Если в сети нет локальных серверов DHCP или DNS, или, если вы не хотите использовать пути по умолчанию к `preseed.cfg`, то вы всё ещё можете явно задать url, и, если вы не используете элемент `./`, то это будет началом пути (третий `/` в URL). Вот пример, для которого нужна минимальная поддержка служб в локальной сети:

```
auto url=http://192.168.1.2/path/to/mypreseed.file
```

Это работает так:

- если в URL не указан протокол, то используется `http`,
- если в имени машины нет точек, то будет добавлен домен, полученный по DHCP, и
- если нет `/` после имени машины, то добавляется путь по умолчанию.

Также, помимо указания url, вы можете задать настройки, которые не влияют напрямую на поведение самого `debian-installer`, но будут переданы во все сценарии, заданные заданы с помощью `preseed/run` в загруженном файле ответов. В настоящий момент, как пример использования, есть только `auto-install/classes` с псевдонимом `classes`. Это можно использовать так:

```
auto url=example.com classes=class_A;class_B
```

`classes`, например, может означать тип устанавливаемой системы или используемую локализацию.

Естественно, можно расширить эту концепцию, и если вы это делаете, имеет смысл задействовать пространство имён `auto-install`. Это может выглядеть как `auto-install/style`, что затем будет использовано в ваших сценариях. Если вы думаете, что вам это нужно, напишите в список рассылки [debian-boot@lists.debian.org](mailto:debian-boot@lists.debian.org), для того чтобы мы смогли избежать конфликта имён, и возможно, добавили псевдонимы для ваших параметров.

Выбор загрузки `auto` пока определён не для всех архитектур. Такой же эффект можно получить просто добавив два параметра `auto=true priority=critical` в командной строке параметров ядра. Параметр ядра `auto` — это псевдоним для `auto-install/enable` и установка его в `true` задерживает выдачу вопросов о локали и клавиатуре, чтобы была возможность настроить их автоматически, а `priority` является псевдонимом для `debconf/priority`, и установка в значение `critical` заставляет программу установки не задавать вопросы с меньшим уровнем важности.

Для автоматизации с использованием DHCP могут пригодиться следующие дополнительные параметры: `interface=auto netcfg/dhcp_timeout=60`, будет выбрана первая рабочая сетевая карта, ответ на DHCP запрос будет ожидаться больше чем обычно.

#### Подсказка



Большой пример использования данной интегрированной среды, включающий примеры сценариев и классов, можно найти на [сайте разработчика](#). На примерах с сайта также показано много других замечательных эффектов, которых можно достичь, если творчески подойти к процессу автоматизации.

### В.2.4 Полезные в автоматизации псевдонимы

При использовании автоматизации установки (режим `auto`) могут пригодиться следующие псевдонимы. Отметим, что сокращаются только имена вопросов, и вам всегда нужно указывать ещё и значение. Например, `auto=true` или `interface=eth0`.

<code>priority</code>	<code>debconf/priority</code>
<code>fb</code>	<code>debian-installer/framebuffer</code>
<code>language</code>	<code>debian-installer/language</code>
<code>country</code>	<code>debian-installer/country</code>
<code>locale</code>	<code>debian-installer/locale</code>
<code>theme</code>	<code>debian-installer/theme</code>
<code>auto</code>	<code>auto-install/enable</code>
<code>classes</code>	<code>auto-install/classes</code>
файловый	<code>preseed/file</code>
<code>url</code>	<code>preseed/url</code>
<code>domain</code>	<code>netcfg/get_domain</code>
<code>hostname</code>	<code>netcfg/get_hostname</code>
<code>interface</code>	<code>netcfg/choose_interface</code>
<code>protocol</code>	<code>mirror/protocol</code>
<code>suite</code>	<code>mirror/suite</code>
<code>modules</code>	<code>anna/choose_modules</code>
<code>recommends</code>	<code>base-installer/install-recommends</code>
<code>tasks</code>	<code>tasksel:tasksel/first</code>
<code>desktop</code> (рабочий стол)	<code>tasksel:tasksel/desktop</code>
<code>dmraid</code>	<code>disk-detect/dmraid/enable</code>
<code>keymap</code>	<code>keyboard-configuration/xkb-keymap</code>
<code>preseed-md5</code>	<code>preseed/file/checksum</code>

### В.2.5 Использование сервера DHCP для указания файлов автоматической конфигурации

Также возможно использовать DHCP для указания файла автоматической конфигурации, загружаемого по сети. DHCP позволяет указать имя файла. Обычно, он указывается для способа установки по сети, но если он задан в виде URL, то установочный носитель, который поддерживает автоматическую установку по сети, загрузит файл с указанного в URL адреса и использует его как файл автоматической установки. Вот пример, как можно настроить `dhcpd.conf` для третьей версии сервера ISC DHCP (Debian пакет `isc-dhcp-server`).

```
if substring (option vendor-class-identifier, 0, 3) = "d-i" {
    filename "http://host/preseed.cfg";
}
```

```
}
```

Заметим, что пример выше ограничивает использование имени файла только DHCP клиентами с идентификатором "d-i", поэтому он не влияет на простых DHCP клиентов, а только на программу установки. Вы также можете поместить текст в строфу указав определённую машину, чтобы избежать автоматизации всех установок в сети.

При автоматизации с помощью DHCP лучше указывать только значения специфичные для вашей сети, например, желаемые сервера-зеркала Debian. В этом случае при установке по сети машины получают правильные сервера-зеркала, а оставшаяся установка будет выполнена в интерактивном режиме. Использование DHCP для полной автоматической установки Debian тоже возможна, но с большой осторожностью.

## В.3 Создание файла ответов

Файл ответов аналогичен формату файла используемому программой **debconf-set-selections**. Общая форма строки файла ответов:

```
<владелец> <имя вопроса> <тип вопроса> <значение>
```

The file should start with `#_preseed_V1`

При написании файла ответов есть несколько правил, которые нужно помнить.

- Между типом и значением вставляйте только один пробел или отступ(tab): любой дополнительный пробельный символ будет включён в значение.
- Строка может быть разбита на несколько строк, с помощью добавления обратного слеша («\») в конец строки. Хорошо разделять строку после имени вопроса; плохо — между типом и значением. При обратном объединении нескольких строк все начальные/конечные пробельные символы будут сжаты в один.
- Владельцем переменных debconf (шаблонов), используемых только в самой программе установки, нужно назначать «d-i»; для переменных автоматизации установки, используемых в установленной системе, должно использоваться имя пакета, содержащее соответствующий шаблон debconf. В базу данных debconf в установленной системе попадут только переменные, у которых владельцем установлено нечто, отличное от «d-i».
- Для большинства вопросов нужно указывать ответы на английском языке, а не их перевод. Однако, есть несколько вопросов (например, в `partman`), где нужно использовать ответы на родном языке.
- В ответах некоторых вопросов вместо английского текста указывается код, который показывается во время установки.
- Start with `#_preseed_V1`

Самый простой путь создания файла ответов это использовать файл примера Раздел [В.4](#) в качестве основы и изменить его.

Альтернативный метод — выполнить установку вручную и затем, после перезагрузки, использовать команду **debconf-get-selections** из пакета `debconf-utils`, чтобы сохранить базу данных debconf и базу данных программы установки `sdebconf` в один файл:

```
$ echo "#_preseed_V1" > file
$ debconf-get-selections --installer >> file
$ debconf-get-selections >> file
```

Однако, таким образом созданный файл будет содержать некоторые элементы, которые не должны быть автоматизированы, и всё-таки для большинства пользователей лучше использовать файл примера.

**Замечание**

Этот метод основан на том, что в конце установки, база данных программы установки `cdebconf` сохраняется в установленной системе в `/var/log/installer/cdebconf`. Однако, так как база данных может содержать частную информацию, по умолчанию файл разрешено читать только суперпользователю.

Каталог `/var/log/installer` и все файлы в нём будут удалены из системы при вычистке пакета `installation-report`.

Для выяснения возможных ответов на вопросы можно использовать редактор **nano** для просмотра файла `/var/lib/cdebconf` прямо во время процесса установки. В файле `templates.dat` хранятся необработанные шаблоны, а в `questions.dat` — текущие значения и значения, назначенные переменным.

Для проверки правильности формата файла ответов до установки, можно использовать команду **debconf-set-selections -c preseeds.cfg**.

## В.4 Содержимое файла ответов (для buster)

Фрагменты конфигурации, используемые в этом приложении, также доступны в примере файла ответов <http://d-i.aliioth.debian.org/manual/example-preseed.txt>.

Заметим, что показанный пример основан на установке для архитектуры Intel x86. Если вы выполняете установку на другую архитектуру, то некоторые примеры (например, выбор клавиатуры и установка загрузчика) могут различаться и должны быть заменены на соответствующие настройки `debconf` для этой архитектуры.

О том, как работают различные компоненты Debian Installer, можно найти в Раздел [6.3](#).

### В.4.1 Локализация

During a normal install the questions about localization are asked first, so these values can only be preseeded via the `initrd` or `kernel boot parameter` methods. Auto mode (Раздел [B.2.3](#)) includes the setting of `auto-install/enable=true` (normally via the `auto` preseed alias). This delays the asking of the localisation questions, so that they can be preseeded by any method.

Для указания языка и страны можно использовать локаль; возможна любая комбинация для языка, если она поддерживается `debian-installer`, и страны. Если в локали указана недопустимая комбинация, то программа установки автоматически выберет локаль, подходящую для выбранного языка. Чтобы указать локаль в параметре загрузки, используйте **locale=ru\_RU**.

Хотя этот метод очень прост в использовании, с его помощью нельзя задать любые комбинации языка, страны и локали<sup>3</sup>. Поэтому есть возможность задать автоматический ответ для каждого параметра отдельно. Язык и страну также можно указывать в параметрах запуска.

```
# Автоответ только для языка, страны и локали.
d-i debian-installer/locale string en_US

# Значения можно указывать отдельно для каждого параметра.
#d-i debian-installer/language string en
#d-i debian-installer/country string NL
#d-i debian-installer/locale string en_GB.UTF-8
# В добавок, можно указать дополнительные локали для генерации.
#d-i localechooser/supported-locales multiselect en_US.UTF-8, nl_NL.UTF-8
```

При настройке клавиатуры выбирается раскладка и (для нелатинских раскладок) клавиша переключения между нелатинской раскладкой и раскладкой US. Во время установки доступны только основные варианты раскладок. Дополнительные варианты доступны только в установленной системе; их можно выбрать с помощью команды **dpkg-reconfigure keyboard-configuration**.

```
# Выбор клавиатуры.
```

<sup>3</sup> При задании значения для `locale` равным `en_NL`, например, приведёт к установке системы с локалью по умолчанию `en_US.UTF-8`. Если, например, требуется `en_GB.UTF-8`, то значения для автоматического ответа нужно задавать отдельно для каждого параметра.

```
d-i keyboard-configuration/xkb-keymap select ru
# d-i keyboard-configuration/toggle select No toggling
```

Чтобы не выполнять настройку клавиатуры укажите в keymap значение **skip-config**. В результате останется активной клавиатурная раскладка ядра.

## В.4.2 Настройка сети

Естественно, автоматическая настройка сети не работает, если вы загружаете файл ответов из сети. Но она отлично работает, если загрузка происходит с CD или USB накопителя. Если вы загружаете файл ответов из сети, то можете указать конфигурацию сети через загрузочные параметры ядра.

Если нужно использовать определённый интерфейс для загрузки по сети перед загрузкой файла ответов из сети, используйте параметр загрузки **interface=eth1**.

Хотя автоматическая настройка сети обычно невозможна при загрузке файла ответов по сети (с помощью «preseed/url»), вы можете обойти этот недостаток, например, если задав статический адрес на сетевом интерфейсе. Данное обходное решение заставит запуститься настройку сети ещё раз после загрузки файла ответов, если создать сценарий «preseed/run» со следующими строчками:

```
kill-all-dhcp; netcfg
```

Следующие переменные debconf относятся к настройке сети.

```
# Выключить настройку сети полностью. Это полезно при установке с cdrom
# на устройства, не имеющих сети, где вопросы о сети,
# предупреждения и длительные паузы неуместны.
#d-i netcfg/enable boolean false

# netcfg выбирает активный сетевой интерфейс, если такой имеется. Если
# есть более одного интерфейса, то при данной настройке список
# интерфейсов выводиться не будет.
d-i netcfg/choose_interface select auto

# Чтобы использовать нужный интерфейс:
#d-i netcfg/choose_interface select eth1

# Задаёт время ожидания появления связи (link) (по умолчанию 3 секунды).
# Значения указываются в секундах.
#d-i netcfg/link_wait_timeout string 10

# Если у вас медленный сервер dhcp и программа установки не дожидается
# ответа, то данная настройка может оказаться полезной.
#d-i netcfg/dhcp_timeout string 60
#d-i netcfg/dhcpv6_timeout string 60

# Если вы предпочитаете настраивать сеть вручную, раскомментируйте эту
# строку и статические настройки сети ниже.
#d-i netcfg/disable_autoconfig boolean true

# Если вы хотите, чтобы файл ответов работал при наличии и без сервера
# dhcp, раскомментируйте эти строки и статические настройки сети ниже.
#d-i netcfg/dhcp_failed note
#d-i netcfg/dhcp_options select Configure network manually

# Статические настройки сети.
#
# Пример для IPv4
#d-i netcfg/get_nameservers string 192.168.1.1
#d-i netcfg/get_ipaddress string 192.168.1.42
#d-i netcfg/get_netmask string 255.255.255.0
#d-i netcfg/get_gateway string 192.168.1.1
#d-i netcfg/confirm_static boolean true

# Пример для IPv6
#d-i netcfg/get_ipaddress string fc00::2
```

```
#d-i netcfg/get_netmask string ffff:ffff:ffff:ffff::
#d-i netcfg/get_gateway string fc00::1
#d-i netcfg/get_nameservers string fc00::1
#d-i netcfg/confirm_static boolean true

# Имя узла и домен, назначенные с помощью dhcp, имеют приоритет над
# значениями, установленными здесь. Однако, установка значений всё ещё
# предотвращает показ вопросов, даже если значения получены по dhcp.
d-i netcfg/get_hostname string unassigned-hostname
d-i netcfg/get_domain string unassigned-domain

# Если вы хотите принудительно задать имя узла, не учитывая того, что
# предложит сервер DHCP или какая будет запись для этого IP в
# обратной зоне DNS, раскомментируйте
# и измените следующую строку.
#d-i netcfg/hostname string somehost

# Выключает показ надоедливового диалога о ключе WEP.
d-i netcfg/wireless_wep string
# Странное имя dhcp-сервера, которое используют некоторые провайдеры в
# качестве некоего пароля.
#d-i netcfg/dhcp_hostname string radish

# Если для работы сети или другого оборудования требуется несвободная
# микропрограмма, то вы можете указать программе установки всегда пытаться
# её загружать не задавая вопросов. Или замените значение на false, чтобы
# этот вопрос вообще не задавался.
#d-i hw-detect/load_firmware boolean true
```

Заметим, что **netcfg** автоматически выберет маску сети, если она не указана в `netcfg/get_netmask`. В этом случае, переменная будет помечена как `seen` при автоматической установке. Аналогично **netcfg** выберет подходящий адрес шлюза, если не указано значение в `netcfg/get_gateway`. Но есть специальная настройка: `netcfg/get_gateway` можно установить в «none», чтобы указать, что шлюз вообще не должен использоваться.

### В.4.3 Сетевая консоль

```
# Используйте следующие настройки, если хотите использовать компоненту
# network-console для удалённой установки через SSH. Это имеет смысл, если вы
# намереваетесь выполнять оставшуюся часть установки вручную.
#d-i anna/choose_modules string network-console
#d-i network-console/authorized_keys_url string http://10.0.0.1/openssh-key
#d-i network-console/password password r00tme
#d-i network-console/password-again password r00tme
```

### В.4.4 Настройка сервера-зеркала

В зависимости от используемого метода установки для загрузки дополнительных компонент программы установки, установки базовой системы и настройки файла `/etc/apt/sources.list` в установленной системе можно использовать сервер-зеркало.

Параметр `mirror/suite` определяет версию дистрибутива для устанавливаемой системы.

Параметр `mirror/udeb/suite` определяет версию дистрибутива для дополнительных компонент программы установки. Он нужен только если компоненты загружаются по сети и должны соответствовать версии, которая была использована для сборки `initrd`, который используется для установки с помощью одноимённого метода. Обычно программа установки автоматически определяет правильное значение, и этот параметр задавать ненужно.

```
# Если вы выберете ftp, то mirror/country string устанавливать не нужно.
#d-i mirror/protocol string ftp
d-i mirror/country string manual
d-i mirror/http/hostname string http.us.debian.org
d-i mirror/http/directory string /debian
```



```
d-i mirror/http/proxy string

# Устанавливаемая версия.
#d-i mirror/suite string testing
# Версия загружаемых компонент (необязательно).
#d-i mirror/udeb/suite string testing
```

### В.4.5 Настройка учётной записи

The password for the root account and name and password for a first regular user's account can be preseeded. For the passwords you can use either clear text values or `crypt(3)` hashes.

#### Внимание



Be aware that preseeding passwords is not completely secure as everyone with access to the preconfiguration file will have the knowledge of these passwords. Storing hashed passwords is considered secure unless a weak hashing algorithm like DES or MD5 is used which allow for bruteforce attacks. Recommended password hashing algorithms are SHA-256 and SHA512.

```
# Skip creation of a root account (normal user account will be able to
# use sudo).
#d-i passwd/root-login boolean false
# Alternatively, to skip creation of a normal user account.
#d-i passwd/make-user boolean false

# Root password, either in clear text
#d-i passwd/root-password password r00tme
#d-i passwd/root-password-again password r00tme
# or encrypted using a crypt(3) hash.
#d-i passwd/root-password-crypted password [crypt(3) hash]

# To create a normal user account.
#d-i passwd/user-fullname string Debian User
#d-i passwd/username string debian
# Normal user's password, either in clear text
#d-i passwd/user-password password insecure
#d-i passwd/user-password-again password insecure
# or encrypted using a crypt(3) hash.
#d-i passwd/user-password-crypted password [crypt(3) hash]
# Create the first user with the specified UID instead of the default.
#d-i passwd/user-uid string 1010

# The user account will be added to some standard initial groups. To
# override that, use this.
#d-i passwd/user-default-groups string audio cdrom video
```

Также, в начале значений переменных `passwd/root-password-crypted` и `passwd/user-password-crypted` можно указать знак «!». В этом случае, соответствующая учётная запись будет заблокирована. Это может быть удобно для учётной записи суперпользователя, если используется другой метод настройки, предоставляющий возможность выполнить административные действия или вход в систему от имени суперпользователя (например, используя аутентификацию по SSH ключу или `sudo`).

The following command (available from the `whois` package) can be used to generate a SHA-512 based `crypt(3)` hash for a password:

```
mkpasswd -m sha-512
```

### В.4.6 Настройка времени и часового пояса

```
# Задаёт, установлены или нет аппаратные часы по Гринвичу.
d-i clock-setup/utc boolean true

# Здесь вы можете указать любое допустимое значение для $TZ; смотрите
# допустимые значения в файле /usr/share/zoneinfo/.
d-i time/zone string US/Eastern

# Определяет, нужно ли использовать NTP для установки часов во время установки
d-i clock-setup/ntp boolean true
# Используемый сервер NTP. Значение по умолчанию указывается здесь.
#d-i clock-setup/ntp-server string ntp.example.com
```

## В.4.7 Разметка дисков

Использование настроек разметки жёсткого диска в автоматической установке очень ограничено; поддерживаются параметры только для `partman-auto`. Вы можете выбрать разметку только свободного места на диске или всего диска. Разметку диска можно выбрать из одного из имеющихся в программе установки наборов, пользовательского набора из файла или набора из файла ответов.

Расширенные возможности разметки с использованием RAID, LVM и шифрования поддерживаются в автоматической установке, но не так гибки как при разметке без автоматической установки.

Представленные далее примеры показывают только основы по использованию наборов. Подробности смотрите в файлах `partman-auto-recipe.txt` и `partman-auto-raid-recipe.txt` из пакета `debian-installer`. Также эти файлы доступны в [репозитории `debian-installer`](#). Заметим, что список поддерживаемых возможностей может поменяться между выпусками.

### Внимание



Идентификация дисков зависит от порядка загрузки драйверов. Если в системе есть несколько дисков, проверьте, что выбирается правильный диск перед тем как использовать автоматическую установку.

### В.4.7.1 Пример разметки

```
# If the system has free space you can choose to only partition that space.
# This is only honoured if partman-auto/method (below) is not set.
#d-i partman-auto/init_automatically_partition select biggest_free

# Alternatively, you may specify a disk to partition. If the system has only
# one disk the installer will default to using that, but otherwise the device
# name must be given in traditional, non-devfs format (so e.g. /dev/sda
# and not e.g. /dev/discs/disc0/disc).
# For example, to use the first SCSI/SATA hard disk:
#d-i partman-auto/disk string /dev/sda
# In addition, you'll need to specify the method to use.
# The presently available methods are:
# - regular: use the usual partition types for your architecture
# - lvm:      use LVM to partition the disk
# - crypto:  use LVM within an encrypted partition
d-i partman-auto/method string lvm

# If one of the disks that are going to be automatically partitioned
# contains an old LVM configuration, the user will normally receive a
# warning. This can be preseeded away...
d-i partman-lvm/device_remove_lvm boolean true
# The same applies to pre-existing software RAID array:
d-i partman-md/device_remove_md boolean true
# And the same goes for the confirmation to write the lvm partitions.
d-i partman-lvm/confirm boolean true
```

```

d-i partman-lvm/confirm_nooverwrite boolean true

# You can choose one of the three predefined partitioning recipes:
# - atomic: all files in one partition
# - home:   separate /home partition
# - multi:  separate /home, /var, and /tmp partitions
d-i partman-auto/choose_recipe select atomic

# Or provide a recipe of your own...
# If you have a way to get a recipe file into the d-i environment, you can
# just point at it.
#d-i partman-auto/expert_recipe_file string /hd-media/recipe

# If not, you can put an entire recipe into the preconfiguration file in one
# (logical) line. This example creates a small /boot partition, suitable
# swap, and uses the rest of the space for the root partition:
#d-i partman-auto/expert_recipe string \
#   boot-root :: \
#       40 50 100 ext3 \
#           $primary{ } $bootable{ } \
#           method{ format } format{ } \
#           use_filesystem{ } filesystem{ ext3 } \
#           mountpoint{ /boot } \
#       . \
#       500 10000 1000000000 ext3 \
#           method{ format } format{ } \
#           use_filesystem{ } filesystem{ ext3 } \
#           mountpoint{ / } \
#       . \
#       64 512 300% linux-swaps \
#           method{ swap } format{ } \
#       .

# The full recipe format is documented in the file partman-auto-recipe.txt
# included in the 'debian-installer' package or available from D-I source
# repository. This also documents how to specify settings such as file
# system labels, volume group names and which physical devices to include
# in a volume group.

# This makes partman automatically partition without confirmation, provided
# that you told it what to do using one of the methods above.
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

# When disk encryption is enabled, skip wiping the partitions beforehand.
#d-i partman-auto-encrypt/erase_disks boolean false

```

### В.4.7.2 Разметка с использованием RAID

Также вы можете автоматизировать настройку массивов программного RAID. Поддерживаются RAID уровни 0, 1, 5, 6 и 10, создание неполных массивов (degraded arrays) и можно указать запасные устройства.

#### Внимание



В данном типе автоматизации разметки легко ошибиться. Также эта возможность тестировалась относительно мало разработчиками `debian-installer`. Ответственность при создании правильного набора (то есть, чтобы он был логически верным и не имел конфликтов) лежит на пользователе. Если возникают проблемы -- см. `/var/log/syslog`.

```

# Метод должен быть установлен в значение "raid".
#d-i partman-auto/method string raid
# Задать размечаемые диски. Они будут иметь одинаковую разметку,
# поэтому это работает только, если диски одинакового размера.
#d-i partman-auto/disk string /dev/sda /dev/sdb

# Далее нужно указать какие физические разделы будут использоваться.
#d-i partman-auto/expert_recipe string \
#     multiraid ::                                \
#         1000 5000 4000 raid                      \
#         $primary{ } method{ raid }              \
#     .                                            \
#         64 512 300% raid                          \
#         method{ raid }                          \
#     .                                            \
#         500 10000 1000000000 raid                \
#         method{ raid }                          \
#     .

# Наконец, нужно указать как использовать определённые ранее разделы
# в настройке RAID. Указывайте правильные номера разделов
# для логических разделов. Поддерживается RAID 0, 1, 5, 6 и 10;
# устройства разделяются с помощью "#".
# Параметры:
# <raidtype> <devcount> <sparecount> <fstype> <mountpoint> \
#     <devices> <sparedevices>

#d-i partman-auto-raid/recipe string \
#     1 2 0 ext3 /                                  \
#         /dev/sda1#/dev/sdb1                      \
#     .                                            \
#     1 2 0 swap -                                  \
#         /dev/sda5#/dev/sdb5                      \
#     .                                            \
#     0 2 0 ext3 /home                             \
#         /dev/sda6#/dev/sdb6                      \
#     .

# Полностью формат набора описан в файле partman-auto-raid-recipe.txt,
# включённом в пакет debian-installer и репозиторий D-I.

# Это заставит partman сделать разметку автоматически, без подтверждения.
d-i partman-md/confirm boolean true
d-i partman-partitioning/confirm_write_new_label boolean true
d-i partman/choose_partition select finish
d-i partman/confirm boolean true
d-i partman/confirm_nooverwrite boolean true

```

### В.4.7.3 Управление монтированием разделов

Обычно, файловые системы монтируются с использованием в качестве ключа универсального уникального идентификатора (UUID); это позволяет монтировать их правильно, даже если изменились их имена устройств. UUID-ы длинные и их сложно запоминать, поэтому, по вашему желанию, программа установки может смонтировать файловые системы основываясь на традиционных именах устройств, или на основе назначенных вами меток. Если вы укажете программе установки монтировать по меткам, то все файловые системы без метки будут смонтированы на основе UUID.

Для устройств со стабильными именами, такими как логические тома LVM, при монтировании будет продолжено использование их традиционных имён, а не UUID-ы.

**Внимание**

Традиционные имена устройств могут изменяться из-за порядка, в котором ядро обнаруживает устройства при загрузке, что может приводить к монтированию не тех файловых систем. Подобным образом, могут встретиться одинаковые метки, если вы подключаете новый диск или USB-устройство, и если это случается, то ваша система начинает работать непредсказуемым образом.

```
# По умолчанию, используется монтирование по UUID, но вы можете
# выбрать "traditional", чтобы использовать обычные имена устройств
# или "label", чтобы попробовать монтирование по меткам, перед
# тем как откатиться к UUID-ам.
#d-i partman/mount_style select uuid
```

**В.4.8 Установка базовой системы**

Для этого этапа, фактически, не так много автоматизируемых настроек. В основном они касаются установки ядра.

```
# Настроить АРТ для отмены автоматической установки рекомендуемых пакетов.
# Использование данного параметра может привести к установке неполноценной
# системы, и он должен использоваться только
# очень опытными пользователями.
#d-i base-installer/install-recommends boolean false
```

```
# Устанавливаемый пакет (мета) с образом ядра; можно указать «none»,
# если ядро устанавливать не нужно.
#d-i base-installer/kernel/image string linux-image-686
```

**В.4.9 Настройка Apt**

Настройка файла /etc/apt/sources.list и основных параметров конфигурации выполняется автоматически на основании метода установки и ответов на ранее заданные вопросы. Необязательно вы можете добавить другие (локальные) репозитории.

```
# You can choose to install non-free and contrib software.
#d-i apt-setup/non-free boolean true
#d-i apt-setup/contrib boolean true
# Uncomment this if you don't want to use a network mirror.
#d-i apt-setup/use_mirror boolean false
# Select which update services to use; define the mirrors to be used.
# Values shown below are the normal defaults.
#d-i apt-setup/services-select multiselect security, updates
#d-i apt-setup/security_host string security.debian.org

# Additional repositories, local[0-9] available
#d-i apt-setup/local0/repository string \
#     http://local.server/debian stable main
#d-i apt-setup/local0/comment string local server
# Enable deb-src lines
#d-i apt-setup/local0/source boolean true
# URL to the public key of the local repository; you must provide a key or
# apt will complain about the unauthenticated repository and so the
# sources.list line will be left commented out
#d-i apt-setup/local0/key string http://local.server/key

# By default the installer requires that repositories be authenticated
# using a known gpg key. This setting can be used to disable that
# authentication. Warning: Insecure, not recommended.
#d-i debian-installer/allow_unauthenticated boolean true
```

```
# Uncomment this to add multiarch configuration for i386
#d-i apt-setup/multiarch string i386
```

### В.4.10 Выбор пакетов

Вы можете выбрать любую доступную комбинацию устанавливаемых задач. Доступные задачи:

- **standard** (стандартные инструменты)
- **desktop** (графический рабочий стол)
- **gnome-desktop** (рабочий стол Gnome)
- **xfce-desktop** (рабочий стол XFCE)
- **kde-desktop** (KDE Plasma desktop)
- **cinnamon-desktop** (рабочий стол Cinnamon)
- **mate-desktop** (рабочий стол MATE)
- **lxde-desktop** (рабочий стол LXDE)
- **web-server** (веб-сервер)
- **print-server** (сервер печати)
- **ssh-server** (сервер SSH)

Также вы можете не выбирать задачи и выполнить установку списка пакетов каким-то другим способом. Мы рекомендуем всегда включать задачу **standard**.

Если вы хотите установить несколько отдельных пакетов в дополнении к пакетам, устанавливаемым задачами, то вы можете использовать параметр `pkgsel/include`. В значении этого параметра в качестве разделителя можно использовать запятые или пробелы, поэтому также его можно указать в командной строке ядра.

```
#tasksel tasksel/first multiselect standard, web-server, kde-desktop

# Отдельные дополнительные пакеты для установки
#d-i pkgsel/include string openssh-server build-essential
# Обновлять ли пакеты после debootstrap.
# Возможные значения: none, safe-upgrade, full-upgrade
#d-i pkgsel/upgrade select none

# Некоторые версии программы установки могут отсылать отчёт
# об установленных и используемых пакетах. По умолчанию данная
# возможность выключена, но отправка отчёта помогает проекту
# определить популярность программ и какие из них включать на CD.
#popularity-contest popularity-contest/participate boolean false
```

### В.4.11 Завершение установки

```
# При установке с консоли на последовательном порту обычные виртуальные
# консоли (VT1-VT6) выключаются в /etc/inittab. Раскомментируйте
# следующую строку, чтобы этого не происходило.
#d-i finish-install/keep-consoles boolean true

# Не показывать последнее сообщение о том, что установка завершена.
d-i finish-install/reboot_in_progress note

# Не выдвигать лоток CD при перезагрузке, что иногда может пригодиться.
#d-i cdrom-detect/eject boolean false
```

```
# Остановить систему после завершения установки, а
# не перегружаться в установленную систему.
#d-i debian-installer/exit/halt boolean true
# Эта настройка позволяет выключить питание машины, а не просто остановить её.
#d-i debian-installer/exit/poweroff boolean true
```

### В.4.12 Автоматизация других пакетов

```
# В зависимости от выбранного для установки ПО, или если что-то идёт
# неправильно во время процесса установки, существует вероятность появления других
# вопросов. Естественно, их тоже можно автоматизировать. Чтобы получить список
# всех возможных вопросов, которые могут быть заданы во время установки, выполните
# установку, а затем запустите следующие команды:
# debconf-get-selections --installer > файл
# debconf-get-selections >> файл
```

## В.5 Дополнительные параметры

### В.5.1 Запуск пользовательских команд во время установки

Инструментарий автоматической установки предлагает очень мощную и гибкую возможность — способность запуска команд или сценариев в определённые моменты установки.

Когда целевая файловая система смонтирована, она доступна в `/target`. При использовании установочного CD, когда он смонтирован, то доступен в `/cdrom`.

```
# Автоматизация d-i по сути небезопасна. Программа установки не выполняет
# проверок на переполнения буфера или другого использования значений
# такого направления из файла ответов. Используйте файлы ответов только из
# надёжных источников! Для использования дома, и потому что это обычно
# полезно, здесь показано как запускать свои команды оболочки внутри
# программы установки автоматически.

# Эта первая команда, которая выполняется как можно раньше сразу после
# чтения файла ответов.
#d-i preseed/early_command string anna-install какой-то-udeb
# Эта команда выполняется непосредственно перед запуском утилиты разметки.
# Это может пригодиться для динамической автоматической настройки утилиты
# разметки, зависящей от состояния дисков (которое может быть
# непонятно при работе preseed/early_command).
#d-i partman/early_command \
#     string debconf-set partman-auto/disk "$(list-devices disk | head -n1)"
# Эта команда выполняется перед окончанием установки, пока ещё
# есть рабочий каталог /target. Вы можете выполнить chroot в /target
# и непосредственно использовать его, или же воспользоваться командами
# apt-install и in-target для облегчения задачи установки пакетов и
# запуска команд в целевой системе.
#d-i preseed/late_command string apt-install zsh; in-target chsh -s /bin/zsh
```

### В.5.2 Использование автоматической установки для изменения значений по умолчанию

С помощью автоматической установки можно изменить ответы по умолчанию, но вопросы всё равно будут задаваться. Для этого после задания ответа на вопрос нужно установить флаг *seen* в «false» после указания ответа на вопрос.

```
d-i foo/bar string value
d-i foo/bar seen false
```

Того же эффекта можно достигнуть для *всех* вопросов, указав параметр `preseed/interactive=true` в приглашении к загрузке. Также это может помочь при отладке или тестировании вашего файла ответов.

Заметим, что владелец «d-i» должен использоваться только для переменных, используемых в самой программе установки. Для переменных, принадлежащих устанавливаемым в систему пакетам, вы должны использовать имя этого пакета. См. сноску к Раздел [В.2.2](#).

Если вы выполняете автоматизацию через параметры загрузки, то можете заставить программу установки задавать соответствующий вопрос с помощью оператора «?=», а именно `foo/bar?=value` (или `owner:foo/bar?=value`). Естественно, это действует только на параметры, которые относятся к отображаемым во время установки вопросам, но никак не на «внутренние» параметры.

Чтобы получить более подробную отладочную информацию, используйте загрузочный параметр `DEB_CONF_DEBUG=5`. Он заставит `debconf` печатать намного больше подробностей о текущих настройках каждой переменной и о ходе выполнения сценариев установки каждого пакета.

### В.5.3 Последовательная загрузка нескольких файлов ответов

В файлах ответов поддерживаются ссылки на дополнительные файлы ответов. Любые настройки файлов, загруженных позже, будут перекрывать настройки из файлов, загруженных ранее. Это позволяет поместить, например, общие настройки сети в один файл, а более специфичные для определённых конфигураций в другие файлы.

```
# Можно указать более одного файла, разделив их пробелами; все будут
# загружены. Включаемые файлы, в свою очередь, тоже могут содержать
# директивы preseed/include. Заметим, что если имена файлов
# относительные, то они берутся
# из того же каталога что и файл ответов, в который они включены.
#d-i preseed/include string x.cfg

# Программу установки можно заставить проверять контрольные суммы файлов
# ответов перед их использованием. В данный момент поддерживается только
# md5sums; в списке md5sums файлы должны указываться в том же порядке,
# что и при включении файлов.
#d-i preseed/include/checksum string 5da499872becccfeda2c4872f9171c3d

# Для большей гибкости можно запустить команду оболочки и если она выдаст
# имена файлов ответов, включить эти файлы.
#d-i preseed/include_command \
#     string if [ "'hostname'" = bob ]; then echo bob.cfg; fi

# Наиболее гибкий из всех способ -- загрузить и выполнить программу.
# Программа может использовать команды, например, debconf-set для
# управления базой данных debconf.
# Можно указать более одного файла, разделив их пробелами
# Заметим, что если имена файлов относительные, то они ищутся в том же
# каталоге откуда взят файл ответов, который запускает их.
#d-i preseed/run string foo.sh
```

Также возможно цепочечное выполнение автоматизации из `initrd` или файла, а затем автоматизацию по сети, если задать `preseed/url` в файлах исполняемых первыми. Это приведёт выполнению автоматизации по сети, когда сеть станет доступна. Нужно быть осторожным при настройке, так как будут выполняться два разных по сути метода автоматизации, что, например, приведёт к выполнению команды из `preseed/early` дважды: один раз из первого метода, второй раз после включения сети.



# Приложение С

## Разметка дисков в Debian

### С.1 Дисковые разделы Debian и их размеры

Как минимум, для GNU/Linux требуется один раздел. Вы можете иметь единственный раздел, содержащий всю операционную систему, приложения и ваши личные файлы. Большинство людей считают, что также необходим раздел подкачки, хотя это не совсем верно. «Раздел подкачки (swap)» — это рабочее пространство для операционной системы, которое позволяет использовать дисковое пространство в качестве «виртуальной памяти». Linux работает с виртуальной памятью более эффективно, если она расположена в отдельном разделе. Linux может использовать для подкачки обычный файл, но это не рекомендуется.

Однако большинство людей выделяют под GNU/Linux больше одного раздела. Есть две причины, по которым возможно вы захотите разделить файловую систему на несколько маленьких разделов. Первая — это безопасность. Если что-то повредится в файловой системе, пострадает только один раздел. То есть, вам нужно будет восстановить (из резервной копии, которую вы заботливо храните) только часть системы. Как минимум, вам нужно создать так называемый «корневой раздел». Он содержит наиболее важные компоненты системы. Если любой другой раздел будет повреждён, вы всё равно сможете загрузиться в GNU/Linux чтобы починить систему. Это может спасти вас от неприятной полной переустановки системы.

Вторая причина обычно более важна для бизнеса, но на самом деле это зависит от того, в каком качестве используется машина. Например, почтовый сервер, получая спам, может легко переполнить раздел письмами. Если вы сделаете для `/var/mail` отдельный раздел на почтовом сервере, большинство систем будут продолжать работать даже если вас заспамят.

Единственным настоящим препятствием по использованию нескольких разделов является то, что часто трудно предсказать размер, который для них нужно выбрать. Если вы сделаете раздел слишком маленьким, то вам придётся или переустанавливать всю систему или постоянно перемещать файлы из-за нехватки места на разделе. С другой стороны, если сделать раздел слишком большим, то это трата места впустую, а оно могло пригодиться где-нибудь ещё. В наши дни дисковое пространство дёшево, но зачем выбрасывать деньги на ветер?

### С.2 Структура каталогов

Debian GNU/Linux придерживается стандарта [Filesystem Hierarchy Standard](#) для каталогов и имён файлов. Этот стандарт позволяет пользователям и программному обеспечению быть уверенным в расположении файлов и каталогов. Уровень корневого каталога представляется просто косой чертой `/`. На корневом уровне, все системы Debian содержат следующие каталоги:

Каталог	Содержит
<code>bin</code>	необходимые исполняемые файлы
<code>boot</code>	статичные файлы системного загрузчика
<code>dev</code>	файлы устройств
<code>etc</code>	настройки системы данной машины
<code>home</code>	домашние каталоги пользователей
<code>lib</code>	необходимые библиотеки общего пользования и модули ядра
<code>media</code>	содержит точки монтирования для съёмных носителей

Каталог	Содержит
mnt	точка монтирования для временно монтируемой файловой системы
proc	виртуальный каталог для системной информации
root	домашний каталог суперпользователя
run	изменяемые данные времени выполнения
sbin	необходимые системные исполняемые файлы
sys	виртуальный каталог для системной информации
tmp	временные файлы
usr	вторичная иерархия
var	изменяемые данные
srv	Данные сервисов, предоставляемых системой
opt	дополнительное программное обеспечение

Далее следует список важных предположений относительно каталогов и разделов. Заметьте, что дисковое пространство может использоваться совершенно по-разному, в зависимости от настроек системы и специфики решаемых задач. Приводимые здесь рекомендации являются лишь отправной точкой в решении задачи разметки.

- На корневом разделе / всегда физически должны находиться каталоги /etc, /bin, /sbin, /lib и /dev, иначе будет невозможна загрузка системы. Обычно, для корневого раздела требуется 250–350 МБ.
- В /usr хранятся: все пользовательские программы (/usr/bin), библиотеки (/usr/lib), документация (/usr/share/doc) и так далее. Этой части файловой системы требуется много места. Вы должны выделить как минимум 500 МБ дискового пространства. Объём этого раздела зависит от числа и типа пакетов, планируемых к установке. Для рабочих станций общего назначения или серверов цифра может вырасти до 4–6 ГБ.
- В настоящее время рекомендуется создавать /usr в корневом разделе /, в противном случае возможна проблема с запуском системы. Это означает, что вы должны иметь не менее boot time. This means that you should provide at least 600–750 МБ на диске в корневом разделе, содержащим /usr, или 5–6 ГБ для установки на рабочую станцию или сервер.
- Каталог /var содержит: все изменяющиеся данные типа новостных статей, сообщений электронной почты, веб-сайтов, баз данных, кэша системы управления пакетами и т.д. Размер каталога целиком зависит от степени использования компьютера, но для большинства людей он определяется работой с системой управления пакетами. Если вы собираетесь установить всё, что предлагает Debian, при чём сразу, то 2 или 3 гигабайта для /var должно хватить. Если вы собираетесь производить установку по частям (то есть, установить службы и утилиты, потом программы для работы с текстом, потом X, ...), то можно уложиться в 300–500 МБ. Если пространство жёсткого диска дорого и вы не планируете делать глобальных обновлений системы, то вам будет достаточно 30 или 40 МБ.
- Каталог /tmp содержит: временные данные создаваемые запущенными программами. Обычно достаточно 40–100 МБ. Некоторые приложения — управление резервным копированием, утилиты работы с CD/DVD и мультимедийными данными — могут использовать /tmp для временного хранения файлов. Если вы планируете использовать данные приложения, то должны выделить пространство под /tmp согласно вашим потребностям.
- Каталог /home содержит: отдельный подкаталог для каждого пользователя, где хранятся личные данные. Его размер зависит от количества пользователей в системе и какие файлы они будут хранить в своих каталогах. В зависимости от планируемого использования вы должны зарезервировать около 100 МБ для каждого пользователя, но можете изменить это значение согласно своим требованиям. Выделите больше места, если планируете хранить много мультимедийных файлов (картинок, MP3, фильмов) в своём домашнем каталоге.

### С.3 Рекомендуемые схемы разметки

Для новичков, устанавливающих Debian только для себя на домашние машины, создание единственного раздела для / (плюс раздел подкачки) вероятно, самый лёгкий и простой путь. Однако, если ваш раздел более

6 ГБ, в качестве типа раздела выберите ext3. Разделы ext2 требуют периодической проверки целостности файловой системы, а это на больших разделах может вызвать паузу при загрузке.

Для многопользовательских систем или систем с большим дисковым пространством будет лучше создать для /var, /tmp и /home собственные разделы, отдельные от /.

Вам может потребоваться отдельный раздел /usr/local, если вы планируете установку многих программ, которые не являются частью дистрибутива Debian. Если ваша машина будет почтовым сервером, вам может потребоваться сделать отдельный раздел под /var/mail. Часто, полезно выделить /tmp в отдельный раздел размером 20–50 МБ. Если вы настраиваете сервер, обслуживающий большое количество пользователей, обычно, хорошо иметь отдельный большой /home раздел. В общем, ситуация с разметкой зависит от цели, которой служит компьютер.

Для очень сложных систем вы должны прочитать [Multi Disk HOWTO](#). В нём содержится подробная информация, которая будет наиболее интересна провайдерам Интернет и людям, настраивающим серверы.

По поводу размера раздела подкачки существует много точек зрения. Согласно проверенному опытом методу, размер подкачки должен совпадать с размером оперативной памяти. Также, в большинстве случаев он должен быть не менее 16 МБ. Естественно, есть исключения из правил. Если вы пытаетесь одновременно решить 10000 уравнений на машине с 256 МБ памяти, то вам может потребоваться гигабайтный (или более) раздел подкачки.

На некоторых 32-битных архитектурах (m68k и PowerPC) максимальный размер раздела подкачки равен 2 ГБ. Этого должно хватить почти для любого варианта установки. Однако, если ваши требования к разделу подкачки превышают данную цифру, то, вероятно, лучше попытаться разместить пространство подкачки на нескольких дисках (так называемых «шпинделей») и, если возможно, на разные каналы SCSI или IDE. Ядро будет балансировать нагрузку между различными разделами подкачки, давая большую производительность.

Пример: старая домашняя машина имеет 32 МБ оперативной памяти и 1.7 ГБ жёсткий диск IDE /dev/sda. Здесь можно сделать 500 МБ раздел для другой операционной системы /dev/sda1, 32 МБ раздел подкачки /dev/sda3 и около 1.2 ГБ раздел Linux /dev/sda2.

Про то, как будет использовано дисковое пространство можно определить по задачам, которые вы захотите установить после установки системы; смотрите Раздел [D.2](#).

## С.4 Имена устройств в Linux

Имена дисков и разделов в Linux могут отличаться от имён в других операционных системах. Для того чтобы создавать и монтировать разделы вам нужно знать какие имена используются в Linux. Вот основы схемы именования:

- Первый дисковод гибких дисков называется /dev/fd0.
- Второй дисковод гибких дисков называется /dev/fd1.
- The first hard disk detected is named /dev/sda.
- The second hard disk detected is named /dev/sdb, and so on.
- Первый SCSI CD-ROM называется /dev/scd0, также можно использовать /dev/sr0.

Имена разделов диска получают добавлением к имени диска десятичного числа: sda1 и sda2 представляют первый и второй разделы на первом диске SCSI в системе.

Вот реальный пример. Предположим, что вы имеете систему с 2-мя SCSI дисками, один имеет SCSI адрес 2, а другой SCSI адрес 4. Первый диск (с адресом 2) называется sda, а второй sdb. Если диск sda содержит 3 раздела, то их имена будут sda1, sda2 и sda3. То же относится к диску sdb и его разделам.

Заметим, что если вы имеете два адаптера SCSI (то есть, контроллера), то порядок дисков может показаться непонятным. Лучшее решение в этом случае посмотреть загрузочные сообщения; предполагается, что вы знаете модели дисков и/или их объём.

## С.5 Программы разметки в Debian

Для работы с различными типами жёстких дисков и компьютерных архитектур разработчиками Debian были адаптированы несколько разновидностей программ разметки. На вашей машине могут работать программы, представленные в списке далее.

**partman** Рекомендуемая утилита разметки в Debian. Этот швейцарский армейский нож может также изменять размеры разделов, создавать файловые системы и назначать им точки монтирования.

**fdisk** Самая первая программа разметки дисков в Linux, хороша для гуру.

Будьте осторожны, если у вас на машине есть разделы FreeBSD. В устанавливаемые ядра включена поддержка этих разделов, но способ их отображения в **fdisk** может отличаться. Смотрите [Linux+FreeBSD HOWTO](#)

**cdisk** Простая в использовании, полноэкранная программа разметки дисков для повседневного использования.

Заметим, что **cdisk** не понимает разделов FreeBSD, и ещё раз, поэтому имена устройств могут различаться.

Одна из этих программ будет запущена по умолчанию, когда вы выберете Разметка дисков (или типа того). Возможно использовать другую утилиту разметки из командной строки на VT2, но это не рекомендуется.

# Приложение D

## Разное

### D.1 Устройства Linux

В Linux в каталоге `/dev` содержатся различные специальные файлы. Эти файлы называются файлами устройств. Доступ к устройствам в мире Unix отличается от того, как это сделано в других операционных системах. Здесь это делается через специальный файл, который в действительности обслуживается драйвером (часть ядра Linux), осуществляющим доступ к устройству. Другим, менее используемым типом файла устройства является *канал*. Ниже перечислены наиболее важные файлы устройств.

<code>fd0</code>	Первый дисковод
<code>fd1</code>	Второй дисковод

<code>sda</code>	First hard disk
<code>sdb</code>	Second hard disk
<code>sda1</code>	First partition of the first hard disk
<code>sdb7</code>	Seventh partition of the second hard disk

<code>sr0</code>	First CD-ROM
<code>sr1</code>	Second CD-ROM

<code>ttyS0</code>	Последовательный порт 0, COM1 под MS-DOS
<code>ttyS1</code>	Последовательный порт 1, COM2 под MS-DOS
<code>psaux</code>	Устройство мышь PS/2
<code>gpmdata</code>	Псевдо-устройство, повторитель данных от демона GPM (мыши)

<code>cdrom</code>	Символическая ссылка на привод CD-ROM
<code>mouse</code>	Символическая ссылка на файл устройства мыши

<code>null</code>	Всё, что записывается в это устройство, исчезнет
<code>zero</code>	Можно бесконечно читать нули из этого устройства

#### D.1.1 Установка мыши

Мышь можно использовать одновременно в консоли Linux (с помощью `gpm`) и в среде X window. Обычно, для этого достаточно установить `gpm` и X-сервер. Они должны быть настроены на использование `/dev/input/mice` в качестве устройства мыши. В `gpm` нужно установить протокол работы с мышью `exps2`, а в X — `ExplorerPS/2`. Это производится в соответствующих файлах настройки `/etc/gpm.conf` и `/etc/`

X11/xorg.conf.

Чтобы мышь заработала, должны быть загружены определённые драйверы ядра. В большинстве случаев нужные модули определяются автоматически, но это происходит не всегда, если используются старые мыши, подключаемые к последовательному порту и по шине<sup>1</sup>, что есть большая редкость и встречается только на очень старых компьютерах. Краткая справка по модулям ядра Linux, которые нужны для различных типов мышей:

Модуль	Описание
psmouse	мышь PS/2 (должна определяться автоматически)
usbhid	мышь USB (должна определяться автоматически)
sermouse	Большинство мышей, подключаемых к последовательному порту
logibm	Мышь, подключаемая по шине к карте Logitech
inport	Мышь, подключаемая по шине к карте ATI или Microsoft InPort

Для загрузки драйвера мыши, вы можете воспользоваться командой **modconf** (из пакета с тем же названием) и посмотреть в категории **kernel/drivers/input/mouse**.

## D.2 Дискосое пространство, необходимое для задач

Для стандартной установки на amd64 с ядром по умолчанию, со всеми стандартными пакетами требуется 800 МБ дискового пространства. Минимальная базовая установка без выбранной задачи «Стандартная система» занимает 613 МБ.

### Важно



В обоих случаях — это реальное дисковое пространство, которое используется *после* завершения установки и удаления всех временных файлов. Сюда не входит место, занятое в файловой системе, например, журнальными файлами. Это означает, что во время *процесса* установки и просто работы системы потребуется значительно больше места на диске.

В таблице ниже приведено количество используемого места при установке задач, описанных в tasksel. Данные величины получены с помощью aptitude. Заметим, что некоторые пакеты входят в несколько задач одновременно, поэтому полный размер установки для двух задач вместе может быть меньше, чем полный размер, полученный сложением размеров.

По умолчанию программа установки устанавливает окружение рабочего стола GNOME, но можно выбрать другие окружения рабочего стола, или используя один из специальных CD образов, или указав желаемое окружение рабочего стола при запуске программы установки (см. Раздел 6.3.5.2).

Заметьте, что при подсчёте размера разделов к размеру стандартной установки нужно добавить размеры из таблицы. Большинство размеров из колонки «Размер после установки» нужно приплюсовывать к /usr и /lib, а размеры из колонки «Размер загрузки» занимает место (временно) в /var.

Задача	Размер после установки (МБ)	Размер загрузки (МБ)	Требуемое место для установки (МБ)
Окружение рабочего стола			
• GNOME (по умолчанию)	3163	935	4098
• KDE Plasma	3044	911	3955
• Xfce	2122	593	2715
• LXDE	2133	602	2735
• MATE	2288	644	2932

<sup>1</sup> Мышь, подключаемая к последовательному порту, имеет трапециевидный разъём с 9 отверстиями; мышь с подключением по шине имеет 8-штырьковый круглый разъём, не перепутайте с 6-штырьковым круглым разъёмом мыши PS/2 или 4-штырьковым круглым разъёмом мыши ADB.

Задача	Размер после установки (МБ)	Размер загрузки (МБ)	Требуемое место для установки (МБ)
• Cinnamon	2878	843	3721
Для ноутбука	29	9	38
веб сервер	40	9	49
сервер печати	407	95	502
SSH-сервер	1	0	1

Если вы выбрали установку не на английском языке, **tasksel** может автоматически установить задачу по локализации, если она существует для вашего языка. Необходимое пространство на диске зависит от языка; добавьте около 350 МБ на загрузку и установку.

## D.3 Установка Debian GNU/Linux из системы Unix/Linux

В этом разделе объясняется, как установить Debian GNU/Linux из существующей системы Unix или Linux без использования управляемой из меню программы установки, основанной на библиотеке ncurses, как это описывается в остальном руководстве. Это «междистрибутивное» HOWTO попросили пользователи, переходящие на Debian GNU/Linux из Redhat, Mandriva и SUSE. В этом разделе подразумевается наличие некоторого опыта работы с командами \*nix и навигации по файловой системе. Здесь символ \$ означает команду, вводимую в текущей пользовательской системе, а # относится к команде, вводимой в Debian chroot.

Как только вы получите новую систему Debian, настроенную по вашим предпочтениям, вы можете перенести свои данные (если они есть) в неё и продолжать работу. Поэтому такая установка Debian GNU/Linux считается «с нулевым временем простоя» (zero downtime). Также это мудрый способ работы с аппаратурой, которая не может нормально работать с разными загрузочными или установочными носителями.

### Замечание



Так как это, по большей части, ручная процедура, вы должны помнить, что вам придётся сделать множество начальных настроек системы самостоятельно, которые также требуют больше знаний о Debian и Linux в общем, чем при выполнении обычной установки. Вы не должны ожидать, что эта процедура настроит вам систему так, как это делается при обычной установке. Также вы должны помнить, что эта процедура выполняет только основные этапы настройки системы. Может потребоваться дополнительная установка и/или настройка.

### D.3.1 Начинаем

With your current \*nix partitioning tools, repartition the hard drive as needed, creating at least one filesystem plus swap. You need around 613MB of space available for a console only install, or about 2133MB if you plan to install X (more if you intend to install desktop environments like GNOME or KDE Plasma).

Затем создайте файловые системы на разделах. Например, чтобы создать файловую систему ext3 на разделе /dev/sda6 (это корневой раздел в нашем примере):

```
# mke2fs -j /dev/sda6
```

Чтобы создать файловую систему ext2, не указывайте параметр `-j`.

Проинициализируйте и активируйте раздел подкачки (замените номер раздела на свой):

```
# mkswap /dev/sda5
# sync
# swapon /dev/sda5
```

Смонтируйте один раздел как /mnt/debinst (точка установки, она будет корневой (/) файловой системой в вашей новой системе). Имя точки монтирования абсолютно произвольное.

```
# mkdir /mnt/debinst
# mount /dev/sda6 /mnt/debinst
```

**Замечание**

Если вы хотите монтировать некоторые файловые системы (например, /usr) с отдельных разделов, то нужно создать и смонтировать эти каталоги вручную, перед тем как перейти к следующему этапу.

**D.3.2 Установка debootstrap**

Утилита, которой пользуется программа установки Debian и которая признана официальным способом установки базовой системы Debian является **debootstrap**. Она использует **wget** и **ar**, но зависит только от /bin/sh и часто используемых инструментов Unix/Linux<sup>2</sup>. Установите **wget** и **ar**, если их ещё нет в системе, а затем загрузите и установите **debootstrap**.

Или вы можете использовать следующую процедуру установки вручную. Создайте рабочий каталог для распаковки в него .deb:

```
# mkdir work
# cd work
```

Двоичный файл **debootstrap** расположен в архиве Debian (убедитесь, что вы выбрали правильный файл, соответствующий вашей архитектуре). Скачайте **debootstrap** .deb из [пула](#), скопируйте пакет в рабочий каталог и извлеките из него файлы. Для установки файлов вам потребуются привилегии суперпользователя (root).

```
# ar -x debootstrap_0.X.X_all.deb
# cd /
# zcat /full-path-to-work/work/data.tar.gz | tar xv
```

**D.3.3 Запуск debootstrap**

Команда **debootstrap** может скачивать необходимые файлы прямо из архива. В примере ниже вы можете заменить [http.us.debian.org/debian](http://us.debian.org/debian) на любой сервер-зеркало архива Debian, лучше всего использовать из наиболее близкой к вам сети. Список серверов <http://www.debian.org/mirror/list>.

Если у вас есть компакт-диск с buster Debian GNU/Linux, смонтированный в /cdrom, вы можете подставить URL типа file вместо http URL: **file:/cdrom/debian/**

Substitute one of the following for ARCH in the **debootstrap** command: **amd64**, **arm64**, **armel**, **armhf**, **i386**, **mips**, **mips64el**, **mipsel**, **powerpc**, **ppc64el**, or **s390x**.

```
# /usr/sbin/debootstrap --arch ARCH buster \
  /mnt/debinst http://ftp.us.debian.org/debian
```

If the target architecture is different than the host, you should add the **--foreign** option.

**D.3.4 Настройка базовой системы**

Now you've got a real Debian system, though rather lean, on disk. **chroot** into it:

```
# LANG=C.UTF-8 chroot /mnt/debinst /bin/bash
```

If the target architecture is different from the host, you will need to first copy qemu-user-static to the new host:

```
# cp /usr/bin/qemu-ARCH-static /mnt/debinst/usr/bin
# LANG=C.UTF-8 chroot /mnt/debinst qemu-ARCH-static /bin/bash
```

After chrooting you may need to set the terminal definition to be compatible with the Debian base system, for example:

```
# export TERM=xterm-color
```

Depending on the value of TERM, you may have to install the ncurses-term package to get support for it.

If the target architecture is different from the host, you need to finish the multi-stage boot strap:

<sup>2</sup> Сюда входят базовые утилиты GNU и команды типа **sed**, **grep**, **tar** и **gzip**.



```
/debootstrap/debootstrap --second-stage
```

### D.3.4.1 Создание файлов устройств

В данный момент `/dev/` содержит только основные файлы устройств. Для следующих этапов установки могут потребоваться дополнительные файлы устройств. Есть несколько способов их создания; выбор более подходящего зависит от: а) системы, с которой производится установка; б) будет ли использоваться модульность ядра; в) используются в новой системе динамические (например, посредством `udev`) или статические файлы устройств.

Некоторые доступные способы:

- установка пакета `makedev` и создание используемого по умолчанию набора статических файлов устройств (после выполнения `chroot`)

```
# apt install makedev
# mount none /proc -t proc
# cd /dev
# MAKEDEV generic
```

- создание вручную только выбранных файлов устройств с помощью `MAKEDEV`
- привязка смонтированного `/dev` имеющейся системы поверх `/dev` в устанавливаемой системе; заметим, что сценарии `rootinst` некоторых пакетов могут пытаться создавать файлы устройств, поэтому данный параметр нужно использовать осторожно

### D.3.4.2 Монтирование разделов

Вам нужно создать файл `/etc/fstab`.

```
# editor /etc/fstab
```

Вот пример, который можно изменить в соответствии с вашими условиями:

```
# /etc/fstab: информация о статических монтируемых файловых системах.
#
# файл. система  точка монтир. тип      параметры                дамп проход
/dev/XXX         /               ext3    defaults                 0      1
/dev/XXX         /boot           ext3    ro,nosuid,nodev         0      2

/dev/XXX         none            swap    sw                        0      0
proc             /proc           proc    defaults                 0      0

/dev/fd0         /media/floppy  auto    noauto,rw,exec,usersync  0      0
/dev/cdrom       /media/cdrom   iso9660 noauto,ro,user,exec     0      0

/dev/XXX         /tmp            ext3    rw,nosuid,nodev         0      2
/dev/XXX         /var            ext3    rw,nosuid,nodev         0      2
/dev/XXX         /usr            ext3    rw,nodev                 0      2
/dev/XXX         /home          ext3    rw,nosuid,nodev         0      2
```

Выполните `mount -a` для монтирования всех файловых систем, указанных в файле `/etc/fstab`, а для монтирования определённой файловой системы выполните:

```
# mount /path # пример: mount /usr
```

В системах Debian точки монтирования для съёмных носителей расположены в `/media`, но для обратной совместимости в `/` созданы символические ссылки. Создайте их, если нужно. Пример:

```
# cd /media
# mkdir cdrom0
# ln -s cdrom0 cdrom
# cd /
# ln -s media/cdrom
```

Вы можете монтировать файловую систему `proc` несколько раз и в произвольные места, хотя принято в `/proc`. Если вы не использовали `mount -a`, то перед тем как продолжить, убедитесь, что `proc` смонтирована:

```
# mount -t proc proc /proc
```

Команда `ls /proc` теперь должна показывать, что каталог не пустой. Если это не так, можно смонтировать `proc` за пределами `chroot`:

```
# mount -t proc proc /mnt/debinst/proc
```

### D.3.4.3 Настройка часового пояса

Настройка в третьей строке файла `/etc/adjtime` имеет значение «UTC» или «LOCAL» и определяет как система должна интерпретировать аппаратные часы — установлены ли они по UTC или показывают местное время. Следующая команда позволяет указать это.

```
# editor /etc/adjtime
```

Пример:

```
0.0 0 0.0
0
UTC
```

Следующая команда позволяет указать часовой пояс.

```
# dpkg-reconfigure tzdata
```

### D.3.4.4 Настройка сети

Для настройки сети, отредактируйте `/etc/network/interfaces`, `/etc/resolv.conf`, `/etc/hostname` и `/etc/hosts`.

```
# editor /etc/network/interfaces
```

Здесь приведены некоторые простые примеры из `/usr/share/doc/ifupdown/examples`:

```
#####
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
# See the interfaces(5) manpage for information on what options are
# available.
#####

# We always want the loopback interface.
#
auto lo
iface lo inet loopback

# To use dhcp:
#
# auto eth0
# iface eth0 inet dhcp

# An example static IP setup: (broadcast and gateway are optional)
#
# auto eth0
# iface eth0 inet static
#     address 192.168.0.42
#     network 192.168.0.0
#     netmask 255.255.255.0
#     broadcast 192.168.0.255
#     gateway 192.168.0.1
```

Enter your nameserver(s) and search directives in `/etc/resolv.conf`:

```
# editor /etc/resolv.conf
```

A simple example `/etc/resolv.conf`:

```
search hqdom.local
nameserver 10.1.1.36
nameserver 192.168.9.100
```

Укажите DNS-сервер(ы) и директивы `search` в `/etc/resolv.conf`:

```
# editor /etc/resolv.conf
```

Простой пример `/etc/resolv.conf`:

```
search hqdom.local
nameserver 10.1.1.36
nameserver 192.168.9.100
```

Укажите имя узла (от 2 до 63 символов):

```
# echo DebianHostName > /etc/hostname
```

И базовый `/etc/hosts` с поддержкой IPv6:

```
127.0.0.1 localhost
127.0.1.1 DebianHostName

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0  ip6-localnet
ff00::0  ip6-mcastprefix
ff02::1  ip6-allnodes
ff02::2  ip6-allrouters
ff02::3  ip6-allhosts
```

Если у вас несколько сетевых карт, то вам нужно разместить имена драйверов в файле `/etc/modules` в желаемом порядке. Позже при загрузке каждая карта будет связана с именем интерфейса (`eth0`, `eth1` и т.д.) как вы и ожидали.

### D.3.4.5 Настройка Apt

Debootstrap will have created a very basic `/etc/apt/sources.list` that will allow installing additional packages. However, you may want to add some additional sources, for example for source packages and security updates:

```
deb-src http://ftp.us.debian.org/debian buster main

deb http://security.debian.org/ buster/updates main
deb-src http://security.debian.org/ buster/updates main
```

Make sure to run **apt update** after you have made changes to the sources list.

### D.3.4.6 Настройка раскладки клавиатуры и локалей

To configure your locale settings to use a language other than English, install the `locales` support package and configure it. Currently the use of UTF-8 locales is recommended.

```
# apt install locales
# dpkg-reconfigure locales
```

To configure your keyboard (if needed):

```
# apt install console-setup
# dpkg-reconfigure keyboard-configuration
```

Заметим, что клавиатура не может быть настроена, пока работа производится в `chroot`, но настройка будет произведена после следующей перезагрузки.

### D.3.5 Установка ядра

Если вы намерены загружать эту систему, то вам скорее всего понадобится ядро Linux и системный загрузчик. Определите доступные пакеты с ядрами с помощью:

```
# apt search linux-image
```

Затем установите пакет с ядром, указав подходящий вам вариант.

```
# apt install linux-image-arch-etc
```

### D.3.6 Установка системного загрузчика

To make your Debian GNU/Linux system bootable, set up your boot loader to load the installed kernel with your new root partition. Note that **debootstrap** does not install a boot loader, but you can use **apt** inside your Debian chroot to do so.

Заметим, что это предполагает, что файловое устройство `/dev/sda` было создано ранее. Есть другие методы установки **grub2**, но они здесь не описаны.

### D.3.7 Удалённый доступ: установка SSH и настройка доступа

In case you can login to the system via console, you can skip this section. If the system should be accessible via the network later on, you need to install SSH and set up access.

```
# apt install ssh
```

Root login with password is disabled by default, so setting up access can be done by setting a password and re-enable root login with password:

```
# passwd  
# editor /etc/ssh/sshd_config
```

This is the option to be enabled:

```
PermitRootLogin yes
```

Access can also be set up by adding an ssh key to the root account:

```
# mkdir /root/.ssh  
# cat << EOF > /root/.ssh/authorized_keys  
ssh-rsa ....  
EOF
```

Lastly, access can be set up by adding a non-root user and setting a password:

```
# adduser joe  
# passwd joe
```

### D.3.8 Заключительные шаги

As mentioned earlier, the installed system will be very basic. If you would like to make the system a bit more mature, there is an easy method to install all packages with «standard» priority:

```
# tasksel install standard
```

Of course, you can also just use **apt** to install packages individually.

После установки все скаченные пакеты будут сохранены в каталог `/var/cache/apt/archives/`. Вы можете освободить немного места запустив:

```
# apt clean
```

## D.4 Установка Debian GNU/Linux с использованием PPP через Ethernet (PPPoE)

В некоторых странах PPP через Ethernet (PPPoE) является широко распространённым протоколом для широкополосных подключений (ADSL или кабельных) к провайдеру сервиса Интернет. По умолчанию программа установки не поддерживает настройку сети с использованием PPPoE, но этого можно легко добиться. В этом разделе будет описано как это сделать.

PPPoE соединение, настроенное в время установки, будет также работать и после перезагрузки в установленную систему (смотрите Глава 7).

Для настройки и использования PPPoE во время установки вам нужно выполнять установку с одного из доступных образов CD-ROM/DVD. Другие методы установки не подойдут (например, netboot).

Процесс установки через PPPoE практически не отличается от любого другого способа установки. Далее будут описаны различия.

- Загрузите программу установки с параметром загрузки `modules=ppp-udeb`. Это гарантирует, что компонента PPPoE (`ppp-udeb`) будет загружена и и запущена автоматически.
- Далее выполните обычные начальные шаги установки (выбор языка, страны и клавиатуры; загрузка дополнительных компонент программы установки<sup>3</sup>).
- На следующем этапе происходит определение сетевого аппаратного обеспечения с целью распознавания имеющихся в системе карт Ethernet.
- После этого запускается настройка PPPoE. Программа установки попытается найти концентратор PPPoE (специальный сервер обработки подключений PPPoE) через все найденные интерфейсы Ethernet. Может случиться так, что концентратор не будет найден с первой попытки. Это может происходить из-за медленной или перегруженной сети или серверов. В большинстве случаев вторая попытка обнаружения концентратора будет успешной; чтобы попытаться ещё раз, выберите Настройка и установление PPPoE-соединения из главного меню программы установки.
- После того как концентратор найден, будет предложено ввести регистрационную информацию (имя пользователя и пароль для PPPoE).
- С этого момента программа установки будет использовать полученную информацию для установления соединения PPPoE. Если всё правильно, соединение PPPoE должно настроиться и программа установки сможет использовать его для подключения к Интернет и получения пакетов (если потребуется). Если регистрационная информация неверна или возникли какие-то ошибки, то программа установки остановится, но настройку можно попытаться повторить ещё раз выбрав в меню Настройка и установление PPPoE-соединения.

---

<sup>3</sup> Компонента `ppp-udeb` загружается на этом этапе как одна из дополнительных компонент. При выполнении установки со средним или низким приоритетом (экспертный режим), можно вручную выбрать `ppp-udeb`, а не вводить параметр «`modules`» в строке приглашения к загрузке.

# Приложение Е

## Общие сведения

### Е.1 Об этом документе

Настоящее руководство было создано для программы `debian-installer` из `Sarge`, оно основано на руководстве по установке с использованием `boot-floppies` дистрибутива `Woody`, которое в свою очередь было основано на более ранних руководствах по установке `Debian`, а также на руководстве дистрибутива `Progeny`, выпущенном под лицензией `GPL` в 2003 году.

При написании этого документа использовался формат `DocBook XML`. Выходные форматы созданы различными программами при помощи данных из пакетов `docbook-xml` и `docbook-xsl`.

В целях повышения удобства сопровождения этого документа, мы применяем некоторые возможности `XML`, такие как сущности (`entities`) и профилирующие атрибуты (`profiling attributes`). Они играют роль, схожую с переменными и условиями в языках программирования. Исходный `XML`-текст этого документа содержит информацию обо всех архитектурах, а для изоляции архитектурно-зависимых кусочков текста использованы профилирующие атрибуты.

### Е.2 Как помочь в разработке этого документа

При наличии проблем или предложений по этому документу, лучше всего отправить их в виде сообщения об ошибке в пакете `installation-guide`. Как это сделать можно прочитать в документации к пакету `reportbug` или на сайте [системы отслеживания ошибок Debian](#). Также загляните в [список открытых ошибок руководства по установке](#), не сообщил ли уже кто-то о вашей проблеме. Если такая ошибка есть, то вы можете послать дополнительную или полезную информацию на адрес `xxxx@bugs.debian.org`, где `xxxx` — это номер сообщённой ошибки.

Ещё лучше взять копию исходных текстов этого документа в формате `DocBook XML` и сделать заплатку к ним. Исходные тексты можно найти в [SVN-репозитории программы установки](#). Если вы не сильны в `DocBook`, не беспокойтесь: в каталоге с руководствами есть простая шпаргалка, которая поможет вам приступить к работе. `DocBook` похож на `HTML`, но ориентирован на содержание текста, а не на его внешний вид. Заплаткам в списке рассылки `debian-boot` (смотрите ниже) всегда рады. О том, как получить исходные тексты через `SVN`, можно прочитать в файле `README`, который находится в корне каталога с исходными текстами.

Не связывайтесь с авторами этого документа напрямую — для обсуждения в том числе и этого руководства существует список рассылки `debian-installer`, который доступен по адресу [debian-boot@lists.debian.org](#). На странице [подписки на списки рассылки Debian](#) описано, как на него можно подписаться. Также доступны [архивы](#) этого и других списков.

### Е.3 Люди, внёсшие значительный вклад в написание этого руководства

Первоначально, этот документ был написан Брюсом Перенсом (Bruce Perens), Свенем Рудольфом (Sven Rudolph), Игорем Гробманом (Igor Grobman), Джеймсом Триси (James Treacy) и Адамом Ди Карло (Adam Di Carlo). Себастьян Лей (Sebastian) написал "Краткое руководство по установке".

Мирослав Куре (Miroslav Kufe) описал много новых возможностей в `debian-installer` из `Sarge`. Франц Поп (Frans Pop) был главным редактором и ответственным за выпуск в дистрибутивах `Etch`, `Lenny` и `Squeeze`.

Очень многие пользователи и разработчики Debian внесли свой вклад в этот документ. Особо нужно отметить Михаэля Шмитца (Michael Schmitz) (поддержка m68k), Франка Нойманна (Frank Neumann) (первый автор [руководства по установке на Amiga](#)), Арто Астала (Arto Astala), Эрика Делонея (Eric Delaunay), Бена Коллинза (Ben Collins) (все трое — информация по SPARC), Тапио Лейтонена (Tapio Lehtonen) и Стефана Борццмайера (Stéphane Bortzmeyer) за тексты и многочисленные правки. Мы благодарим Паскаля ЛеБея (Pascal Le Bail) за полезную информацию о загрузке с USB-карт памяти.

Множество полезной информации можно найти в HOWTO по сетевой загрузке под авторством Джима Минта (Jim Mintha) (URL на данный момент недоступен), [Debian FAQ](#), [Linux/m68k FAQ](#), [FAQ по Linux для процессоров SPARC](#), [Linux/Alpha FAQ](#) и в других документах. Мы хотим поблагодарить сопровождающих этих свободно доступных и полезных источников информации.

При создании раздела о chroot установке (Раздел [D.3](#)) были использованы материалы, написанные Карстенем М. Селфом (Karsten M. Self).

## Е.4 Торговые марки

Все торговые марки являются собственностью их владельцев.

# Приложение F

## GNU General Public License

Version 2, June 1991

```
Copyright (C) 1989, 1991 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA.
```

```
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.
```

### F.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the gnu General Public License is intended to guarantee your freedom to share and change free software — to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the gnu Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

### F.2 GNU GENERAL PUBLIC LICENSE

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into



another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

**1.** You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

**2.** You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

**3.** You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

**4.** You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

**5.** You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

**6.** Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

**7.** If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

**8.** If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

**9.** The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

**10.** If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### **NO WARRANTY**

**11.** BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

**12.** IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL AND COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE

THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**END OF TERMS AND CONDITIONS**

### F.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and a brief idea of what it does.
Copyright (C) year name of author
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with absolutely no warranty; for details
type 'show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type 'show c'
for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they could even be mouse-clicks or menu items — whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program 'Gnomovision' (which makes passes at compilers) written
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License.