

# Linux Mint



Освойте на 100 %:

- установку и настройку Linux Mint
- оригинальные приложения ОС
- специфику работы в Интернете

Сергей Яремчук

на **100%**

# Linux Mint



Москва - Санкт-Петербург - Нижний Новгород - Воронеж  
Ростов-на-Дону - Екатеринбург - Самара - Новосибирск  
Киев - Харьков - Минск

2011

ББК 32.973.2-018.2

УДК 004.451

Я72

**Яремчук С. А.**  
**Я72 Linux Mint на 100 %. — СПб.: Питер, 2011. — 240 е.: ил. — (Серия «На 100 %»).**  
**ISBN 978-5-49807-803-8**

Эта книга позволит вам освоить на 100 % Linux Mint — популярнейший дистрибутив операционной системы GNU/Linux. «From freedom came elegance» («От свободы к элегантности») — девиз данного дистрибутива, который совместим с оригинальным Ubuntu и вместе с тем избавлен от многих его недостатков и недочетов, а также оснащен оригинальным интерфейсом и набором собственных приложений, таких как mintInstall, mintUpdate, mintMenu и т. д. Данное руководство рассчитано, в первую очередь, на читателей, не имеющих опыта работы в этой операционной системе. Тем не менее содержащиеся в книге обширный справочный материал и советы профессионалов будут полезны и опытным пользователям ОС Linux. Многие темы и вопросы, раскрываемые в издании, заинтересуют также пользователей дистрибутивов, родственных Linux Mint, в частности Linux Ubuntu, Debian GNU/Linux.

С помощью «Linux Mint на 100 %» вы с легкостью освоите самую элегантную ОС современности и, безусловно, полюбите свежий мятный вкус свободного ПО!

**ББК 32.973.2-018.2**

**УДК 004.451**

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

ISBN 978-5-49807-803-8

© ООО Издательство «Питер», 2011

# Краткое содержание

Введение .....	8
От издательства .....	8
Глава 1. Выбираем Linux .....	9
Глава 2. Установка Linux Mint .....	23
Глава 3. Рабочая среда Linux Mint .....	73
Глава 4. Linux Mint под капотом.....	115
Глава 5. Установка приложений в Linux Mint .....	148
Глава 6. Linux Mint в сети .....	212

# Оглавление

Введение .....	8
От издательства .....	8
Глава 1. Выбираем Linux .....	9
Что такое Linux и зачем он нужен? .....	10
Что особенного в дистрибутивах Linux? .....	14
Пакетные системы .....	17
Оптимизированные решения .....	19
Дистрибутив Linux Mint .....	20
Глава 2. Установка Linux Mint .....	23
Подготовка жесткого диска .....	24
Обозначение дисков в Linux .....	24
Обозначение разделов в Linux .....	25
Что такое UUID? .....	26
Разделы для установки Linux .....	27
Файловые системы Linux .....	36

Версии Linux Mint .....	43
Программа установки Linux Mint .....	46
Альтернативные варианты установки .....	55
Установка с флеш-носителя .....	55
Загрузка без привода компакт-диска .....	58
Загрузчик GNU GRUB .....	59
Восстановление загрузчика после установки Windows .....	68
Удаление Linux .....	71
Глава 3. Рабочая среда Linux Mint .....	73
Рабочий стол GNOME .....	76
Утилиты настройки .....	84
Настройка соответствия приложений .....	88
Обустроиваем Рабочий стол .....	91
Настраиваем клавиатуру .....	96
Автозапуск приложений .....	104
Продвинутые настройки .....	111
Редактор GConf .....	112
Глава 4. Linux Mint под капотом .....	115
Файлы и каталоги .....	116
Права доступа .....	123
Монтирование разделов .....	130
Команда mount .....	131
Автоматическое монтирование устройств — /etc/fstab .....	134
Перенос раздела /home .....	135
Управление процессами .....	137
Файловая система procfs .....	140
Постоянные установки ядра .....	143
Сбор информации о системе .....	145

Глава 5. Установка приложений в Linux Mint .....	148
Особенности программ Linux .....	149
Пакетные системы .....	151
Утилита aptitude .....	161
Графические утилиты для установки программ .....	166
Описание репозитариев в Linux Mint .....	173
Компиляция программ .....	181
Конвертеры пакетов .....	183
Некоторые приложения, входящие в состав Linux Mint .....	184
Офисные пакеты .....	184
Растровый редактор GIMP .....	194
Менеджер фотографий F-Spot .....	195
Программы для просмотра видео .....	198
Видеопроигрыватель MPlayer .....	199
Проигрыватель VLC .....	202
Музыкальные проигрыватели .....	204
Программы для записи дисков .....	207
Редактирование ID3-тегов .....	209
Видеоредактор OpenShot .....	210
Глава 6. Linux Mint в сети .....	212
Подключаемся к сети .....	213
Настройка Ethernet-соединения .....	216
Настройка PPPoE-соединения .....	218
Подключаемся по PPTP .....	222
Межсетевой экран в Linux Mint .....	224
Программы для работы в Интернете .....	228
Браузеры .....	228
Консольный Интернет .....	229
Браузеры для графической среды .....	230

Программы для работы с почтой .....	232
Mozilla Thunderbird .....	232
Почтовый клиент Evolution .....	233
Менеджеры закачек .....	234
Закачка с командной строки .....	234
Графические менеджеры закачек .....	235
Программы для работы с ICQ, IRC и Jabber .....	236
Клиент VoIP Ekiga .....	238
Клиент микроблогинга Gwibber .....	239

# Введение

Сегодня очень много говорят об альтернативных операционных системах, в частности GNU/Linux. Почему пользователям интересна эта операционная система? Почему они предпочитают ее популярной Windows от Microsoft? Ответ прост. Linux — это в первую очередь свобода. Причем свобода во всех ее возможных проявлениях. Нет коммерческих лицензий, не взимается плата за использование, свобода от вирусов, свобода выбора приложений и, естественно, возможность сделать свой дистрибутив и выложить его на всеобщее одобрение. Использование Linux — это легально, безопасно, надежно, выгодно и интересно. Каждый подбирает инструмент по себе, но он может по тем или иным причинам не подойти другому человеку в определенной ситуации. Вас никто не заставляет его использовать, никто не навязывает дистрибутив, версию, программы и др. Вы сами себе выбираете, что больше подходит. Благо выбирать есть из чего, а денег здесь традиционно от вас не требуют.

А вот выбрать свой дистрибутив Linux, который удовлетворит запросы пользователя на все 100 %, как правило, не так уже и просто. Хотя бы потому, что их много, они разные и каждый хвалит свой вариант. Книга описывает один из самых популярных дистрибутивов — Linux Mint. Многие считают Linux запутанным, но поверьте — это не так. Если разобраться, Linux более логичен, чем Windows. По ходу мы разберем все его особенности: от строения файловой системы до использования популярных приложений. Но будут затронуты многие общие вопросы, поэтому, прочитав книгу, вы не только сможете освоить Linux Mint, но и без проблем научитесь работать в любом другом дистрибутиве, в частности в «родственных» Ubuntu, Debian. Увидите, чем различаются и чем похожи разные дистрибутивы Linux. Добро пожаловать в мир свободных программ!

## От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты [gromakovski@minsk.piter.com](mailto:gromakovski@minsk.piter.com) (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На сайте издательства <http://www.piter.com> вы найдете подробную информацию о наших книгах.

# Глава 1

## Выбираем Linux

Что такое Linux и зачем он нужен?

Что особенного в дистрибутивах Linux?

Дистрибутив Linux Mint

Слово Linux ассоциируется у большинства пользователей с понятием «операционная система». На самом деле все гораздо сложнее и проще одновременно.

## Что такое Linux и зачем он нужен?

«Привет всем, кто использует миникс — Я делаю (бесплатную) операционную систему (это всего лишь хобби, она не будет большой и профессиональной, как GNU) для клонов 386 (486) AT...» Именно это сообщение, появившееся в августе 1991 г. в новостной группе Usenet конференции comp.os.minix (<http://groups.google.com/group/comp.os.minix/msg/b813d52cbc5a044b>), считается точкой отсчета в Linux. Поэтому если изучать историю ее возникновения, то выясняется, что Linux — название именно ядра, предложенного финским студентом Линусом Торвальдсом, а не всей операционной системы, и это ядро само по себе практически бесполезно. Остальное — это труд тысяч и тысяч независимых программистов со всего мира, которые даже в глаза друг друга не видели. Linux не имеет единого центра разработки, хотя каждый проект координируется одним человеком или группой лиц, которые и определяют его дальнейшее развитие. Весь исходный код доступен, поэтому каждый желающий может принимать участие в тестировании и предлагать свои изменения или исправления ошибок. В состав любого дистрибутива входит большое количество программ и библиотек, разрабатываемых сторонними разработчиками, в том числе и в рамках проекта GNU ([www.gnu.org](http://www.gnu.org)) (рис. 1.1).

Translations of this page

GNU Operating System

Sign up for the Free Software Supporter  
A monthly update on GNU and the FSF

you@example.com

GNU/Linux? FSF?

Support! Click a button on your unused Microsoft Windows license  
Sign This Petition to stop software patents in Europe  
Support the efforts on net neutrality in Europe, in the USA and in Canada.

### What is GNU?

The GNU Project was launched in 1984 to develop a complete Unix-like operating system which is **free software** — the GNU operating system.

Unix-like operating systems are built from a collection of **libraries, applications and developer tools**, plus a kernel to allocate resources and talk to the hardware — **that, GNU's kernel** is actively developed, but is still some way from being ready for daily use, so GNU is often used with the kernel Linux.

### Support GNU!

A strong **free software** movement focused on the principled issues of **software freedom** — and a strong FSF in particular — will determine what freedoms the next generation of computer users enjoy. At stake is no less than the next generation's autonomy — **Benjamin Mako Hill**, writer, technologist and FSF board member.

The FSF's work represents the most important work for freedom that this culture has seen in many generations, because it takes the idea of freedom and it removes it from the ivory tower and places it in a community — **Professor Lawrence Lessig**.

[View all the latest appeal videos](#)

### GNUstep

GNUstep is a fully-functional object-oriented development environment.

The combination of GNU and Linux is the **GNU/Linux operating system**, now used by millions.

Sometimes this combination is incorrectly called **Linux**. There are many variants or "distributions" of GNU/Linux.

We recommend the **GNU/Linux distributions** that are 100% free software, in other words.

Рис. 1.1. Сайт проекта GNU

Поэтому главный идеолог движения GNU Ричард Столлман в каждом своем интервью говорит, что правильно называть операционную систему не Linux, а GNU/Linux.

#### ПРИМЕЧАНИЕ



Название Linux получилось практически случайно. Изначально планировалось название FREAХ, но администратор FTP-сервера Анри Лемке отвел под исходные коды новой операционной системы каталог pub/OS/Linux. Впоследствии именно это название и закрепилось за данной системой. Официальный ресурс размещен по адресу <http://www.linux.org/> (рис. 1.2).

The screenshot shows the official Linux website. At the top left is the Linux Online logo with the URL <http://www.linux.org>. Below it are links for Register, Login, Applications, Documentation, Distributions, Download, General Info, and Book Store. A sidebar on the left contains links for X-Win32 LIVE vs. Cygwin/X, Courses, News, and People. The main content area features an article titled "Some facts about downloading Linux" with a sub-header "Last Updated: Thursday January 10, 2008". The article text explains that Linux is an operating system, not a single program, and discusses Live CD versions. It lists requirements for downloading Linux, such as a high-speed Internet connection and the need for ISO or DVD images.

Рис. 1.2. Официальный сайт Linux

Особенность децентрализованного создания программ является еще одним феноменом Linux. Некоторые критики говорят, что в таком случае невозможно разработать действительно эффективное решение, но время показывает, что они не правы. И сегодня подобную схему разработок уже используют многие крупные компании — Oracle, Borland, SAP, Mozilla Foundation и др.

#### ПРИМЕЧАНИЕ



Официальным символом Linux с мая 1996 г. является пингвин Тух, который отличается от «обычных» пингвинов желтым цветом клюва и лап. До этого долго решали, как должен выглядеть талисман, пока Линус на одной из конференций не ответил, что ему нравятся пингвины. Вопрос был решен. Имя TUX (читается «такс», сокращенное от Torvalds UniX) пингвину дал Джеймс Хьюз.

Я не буду подробно излагать всю историю создания Linux, в этом нет необходимости, а при желании нужную информацию можно легко найти в Интернете.

Первое время пользователями Linux были лишь энтузиасты, которые одновременно работали в этой операционной системе и принимали участие в ее развитии. По мере того как Linux становился более функциональным, количество пользователей увеличивалось. Сегодня доля установок Linux на серверах достаточно высока. Так, половина всех веб-серверов, доступных в Интернете, и 86 % ТОП 500 суперкомпьютеров работают под управлением этой операционной системы.



#### ПРИМЕЧАНИЕ

**GNU** — это рекурсивная аббревиатура от **GNU's Not UNIX**, что переводится как «GNU - не UNIX».

Процент использования этой операционной системы на клиентских системах ниже (по разным источникам — около 1 %) и занимает третье место. Linux только начал использоваться на настольных системах, и, очевидно, процент будет расти. Так, по данным NetMarketshare (<http://marketshare.hitslink.com/>), подсчитывающей информацию, собранную о посетителях веб-ресурсов, в 2008 г. Linux использовали 0,87 %, а в начале 2010 г. их было уже 1,06 % (рис. 1.3).

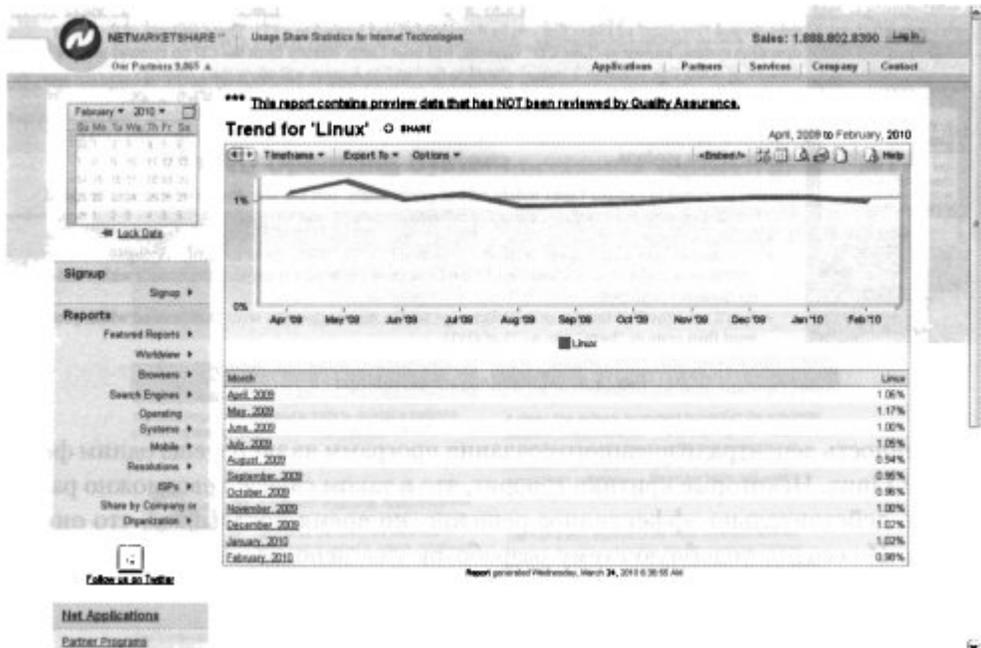


Рис. 1.3. График изменения популярности операционной системы Linux

Росту популярности способствует поддержка крупных производителей (HP, Dell), к тому же во многих государствах и международных организациях приняты про-

граммы поддержки Linux. Следует также отметить популярность Linux на нетбуках, ведь применение свободной операционной системы уменьшает стоимость такого решения. Linux популярен у пользователей, занимающихся той или иной деятельностью профессионально (программисты, системные администраторы, специалисты по безопасности, художники и др.), ведь в этом случае сохраняется лицензионная чистота конечного продукта. Обычного, то есть домашнего, пользователя кроме возможности работы с бесплатным программным обеспечением привлекает изначально безопасность Linux. Для этой системы известен лишь десяток вирусов, и то они эффективны только при наступлении определенных факторов, которые не так легко получить. Разработчики знают, как уберечь систему от взлома, поэтому о вирусных эпидемиях в Linux мы ничего не слышим. Еще один фактор — возможность работы с современным программным обеспечением на компьютере, так сказать «не первой свежести». Ведь Linux легко адаптировать самому под имеющиеся системные требования либо, как вариант, выбрать уже готовую сборку для таких компьютеров.

Важно также понимать, что Linux очень отличается от популярной сегодня Windows как внутренним устройством, принципами работы и настройками, так и взаимодействием с пользователем и лицензией. Например, в отличие от Windows в Linux не используются обозначения дисков C, D и т. д., а любой компонент, в том числе и графическую среду, можно безболезненно выбросить и заменить другим, более легким вариантом или вариантом с другой функциональностью. Кроме того, одной из первых проблем, с которой столкнется любой пользователь, — порядок установки, обновления и удаления программ. Здесь выяснится, что просто скачать файл и дважды щелкнуть на нем кнопкой мыши в большинстве случаев явно недостаточно. В дистрибутивах предлагается даже более простой путь.

Поначалу все «нововведения» покажутся неудобными и непонятными, но на самом деле они более логичны. По ходу прочтения книги все эти тонкости будут раскрыты.

Еще один важный фактор — лицензия. Не секрет, что, покупая программу и операционную систему, пользователь часто приобретает не сам продукт, а лишь право на его использование на условиях, описанных в лицензионном соглашении. Возможно, из-за виртуальности самого продукта многие просто не обращают внимания на этот факт. Но представим, что вы купили, например, фотоаппарат, потратили свои кровные, а стоит дать его соседу, как у вас его отбирают. Нонсенс. Но производители проприетарного (то есть коммерческого) программного обеспечения делают все возможное, чтобы, купив программу, вы не смогли передать или перепродать ее другому. Даже в том случае, если сама программа вам уже не нужна.

В Linux также есть лицензия GNU GPL (GNU Public License), но она гарантирует то, что программное обеспечение, которое она защищает, всегда будет открытым

и доступным для любого пользователя. Ни одна компания, ни один человек не вправе присвоить себе исключительное право собственности. Любой из нас может свободно использовать такие программы, изменять их код, продавать, оказывать услуги по внедрению и т. д., но при одном условии: все продукты, произведенные на основе программы, распространяемой по лицензии GNU GPL, должны также быть открытыми и распространяться по условиям этой лицензии.



#### ПРИМЕЧАНИЕ

Хотелось бы отметить, что у GPL есть и противники, которые считают, что открытость лицензии не идет ей на пользу. Например, многие компании, желающие скрыть свои собственные наработки (например, драйвера), не могут внедрить код в ядро Linux или другие компоненты. Они предпочитают более лояльную лицензию BSD, разрешающую на основе продуктов создавать свои закрытые решения. Наличие двух лицензий GNU GPL и BSD разделило лагерь на две идеологии Open Source и Free Software. Именно GNU GPL является одной из причин (не самой главной) отсутствия драйверов для Linux под некоторые устройства.

Кроме того, хочется внести ясность в вопрос продажи Linux и услуг, с ним связанных. Поскольку английское слово free в русском языке можно перевести не только как «свободный», но и «бесплатный», то многие, говоря о Linux и других решениях, подразумевают именно бесплатность. На самом деле это не так, если быть точнее, то бесплатность — это не основная суть распространения Linux. На сайте проекта GNU (<http://www.gnu.org/philosophy/free-sw.ru.html>), объясняя значение слова free в GNU GPL, уточняют, что, «говоря о свободе, следует подразумевать свободу слова, а не бесплатное пиво». Хотя, действительно, большинство дистрибутивов Linux можно свободно скачать с сайта разработчика и использовать без всяких отчислений.

Собственно, лицензия GNU GPL — это одна из сильных сторон Linux, и благодаря ей мы сегодня имеем такое большое количество дистрибутивов.

## Что особенного в дистрибутивах Linux?

Некоторое время пользователи Linux сами создавали себе систему, используя доступные исходные тексты ядра и добавляя необходимые компоненты. Это требовало наличия определенных знаний, ведь не всегда все сразу работало, и, главное, такой подход не способствовал распространению системы. Через некоторое время нашлись люди, которые посчитали данный способ неудобным, в результате появились первые комплекты, имеющие стабильные компоненты и примитивные средства установки. Первым популярным дистрибутивом Linux, сумевшим завоевать популярность пользователей, был SLS Linux (Softlanding Linux System, <http://www.ibiblio.org/pub/historic-linux/distributions/sls/>). Его разработчики предложили уже готовый набор для установки, не требующий компиляции, плюс понятную программу уста-

новки и возможность простой инсталляции необходимых приложений. Идею быстро подхватили, и, хотя развитие SLS по разным причинам было прекращено, он дал дорогу двум самым популярным дистрибутивам Linux — Slackware (<http://www.slackware.com>) и Debian (<http://www.debian.org>).

Давайте кратко систематизируем дистрибутивы, ориентированные на применение на домашних компьютерах и рабочих станциях.

Большинство дистрибутивов Linux собраны на сайте Distrowatch.com (рис. 1.4), здесь же ведется рейтинг, и любой может узнать, насколько популярен тот или иной дистрибутив. Рейтинг дистрибутивов на русскоязычном пространстве, к сожалению, пока не ведется, но как попытку можно назвать ресурс LinuxForge (<http://linuxforge.ru/>).

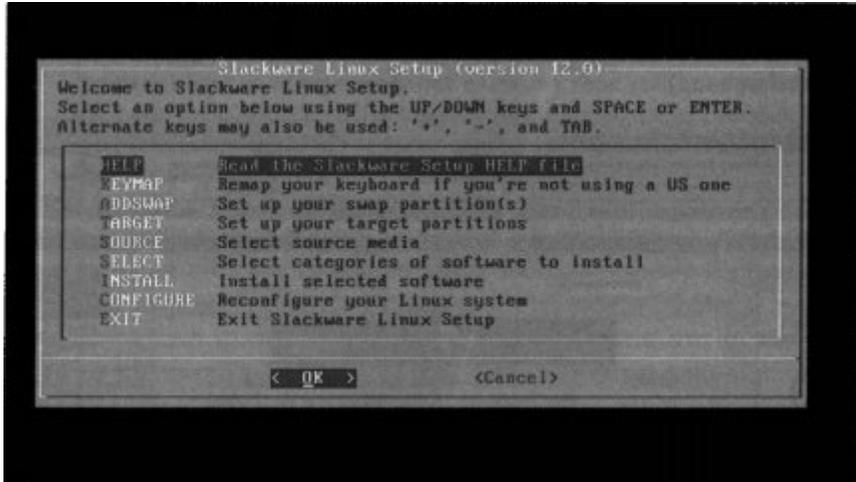


Рис. 1.4. Сайт Distrowatch.com

Выбирая дистрибутив, следует знать, для каких целей он ориентирован и каков уровень подготовки пользователя. Так, на сайте Distrowatch.com дистрибутивы, предназначенные для использования на рабочих станциях, отмечаются как Desktop. Уровень подготовки пользователя задается от начального (Beginners) до подготовленного (Power user). К последним принадлежат дистрибутивы CRUX, Gentoo и Arch Linux.

Косвенно об уровне подготовки пользователя, на которого ориентирован дистрибутив, говорит и программа установки. Чем она проще, тем больше дистрибутив подходит неподготовленному пользователю. Замечу, что можно встретить самые разные инсталляторы: от простых текстовых, которые требуют знания основных

команд Unix и понимания всего процесса, до удобных и красивых графических программ. Первые позволяют полностью контролировать установку, вторые понятны даже неподготовленному пользователю. Хотя дистрибутивы, имеющие только консольный вариант программы установки, практически вымерли, сегодня популярен некий промежуточный вариант — дистрибутивы с псевдографическими инсталляторами, в которых при помощи системы меню задаются вопросы и пользователь лишь выбирает необходимый пункт ответа. Но у него всегда есть возможность выйти в консоль и взять бразды правления в свои руки. Как пример можно привести программу установки Slackware Linux (рис. 1.5).



**Рис. 1.5.** Программа установки Slackware Linux ориентирована на подготовленных пользователей

Дистрибутивы с графической программой установки изначально ориентированы на широкий круг пользователей, поэтому разработчики сделали все, чтобы максимально упростить задачу. Это Ubuntu, Linux Mint, openSUSE, Mandriva Linux и многие другие. В некоторых дистрибутивах возможна установка как в графической, так и в псевдографической среде. Последний вариант очень полезен, когда есть проблемы с оборудованием, в частности с определением видеокарты. Здесь в качестве примера можно привести Debian.

В последнее время стал популярным принцип установки с использованием LiveCD. Пользователь загружает с CD или DVD рабочую систему, тестирует свое оборудование и, если дистрибутив ему подходит, не перегружаясь, устанавливает его прямо в графической среде за 5-7 шагов мастера (рис. 1.6).



#### ПРИМЕЧАНИЕ

Если дистрибутив использует текстовую программу установки, он наверняка рассчитан на опытного пользователя, новичку работать в нем будет сложнее.

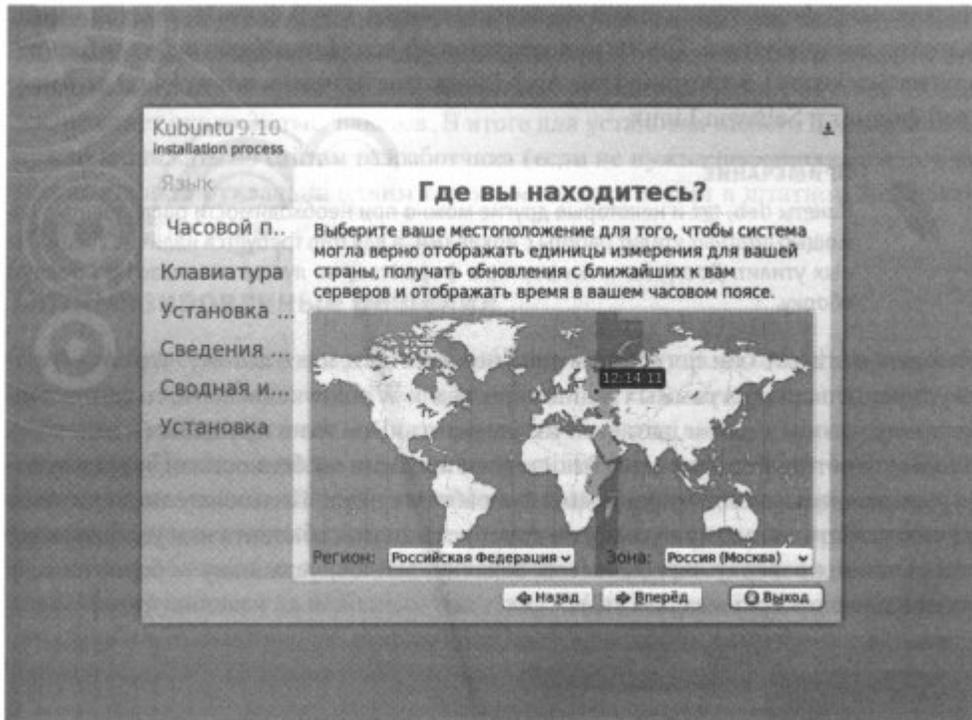


Рис. 1.6. Программа установки Kubuntu Linux

## Пакетные системы

Итак, дистрибутив выбран и установлен. Теперь выясняется, что программы устанавливаются иначе, чем в Windows. Согласно лицензии GNU GPL программы распространяются в исходных текстах, но самостоятельное компилирование не всегда заканчивается удачей, и справиться с возникшими проблемами может только специалист. Чтобы сделать более удобной установку и удаление программ, нужно использовать пакет. Он представляет собой уже откомпилированную программу, которую при помощи специальных менеджеров распаковывают по каталогам и регистрируют в системе.

Форматов пакетов сегодня ненамного меньше, чем дистрибутивов. Посудите сами. Это rpm (RedHat Packet Manager) — формат пакетов, появившийся впервые в дистрибутиве RedHat, а сегодня используется в его многочисленных клонах: CentOS, Fedora, AltLinux, ASPLinux и Mandriva. Причем из этого списка совместимы только первые три. В Debian и его клонах, вроде Ubuntu и Linux Mint, применяется другой формат — deb. И наконец, tgz (он же tar.gz) — формат пакета, используемый в Slackware.

Названными форматами список не исчерпывается. Свой формат пакетов имеют и другие дистрибутивы. Так, `tlz` используется в Vector Linux (дистрибутив базируется на Slackware), в дистрибутиве Arch Linux пакеты называются `pkg.tar.gz`, имеет свой формат и Sabayon Linux.



#### ПРИМЕЧАНИЕ

Пакеты `deb`, `tgz` и некоторые другие можно при необходимости распаковать с помощью штатных утилит работы с архивами, а для `rpm` требуется наличие специальных утилит. Доступны и конвертеры форматов, но лучше использовать родную сборку.

Новичок от такого обилия пакетов явно не в восторге, и все потому, что он пытается устанавливать программы традиционным для Windows способом, то есть скачивает программы и затем пытается их запустить. Все это в прошлом (в далеком). Чтобы упростить борьбу с зависимостями и прочими особенностями, используются репозитории пакетов и программы для работы с ними. Пользователю достаточно только указать, какую программу он хочет установить, обновить или удалить, а все остальное — проверку зависимостей, загрузку файлов, установку — берет на себя специальная программа (рис. 1.7).

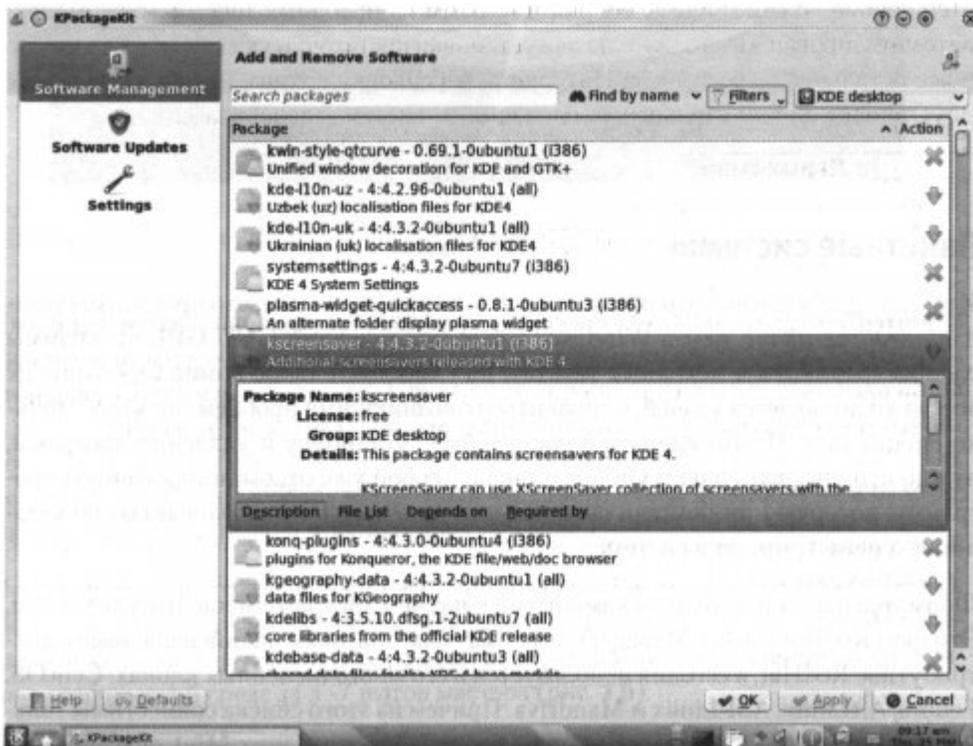


Рис. 1.7. Программа установки приложений в Kubuntu Linux

Каждый дистрибутив поддерживает свой репозиторий, к тому же сообщество пользователей обычно добавляет к нему один-два, а то и более альтернативных, в которых можно найти все, что угодно. Например, репозиторий дистрибутива Ubuntu насчитывает около 28 тыс. пакетов. В итоге для установки любого приложения не нужно его искать по сайтам разработчика (если не нужна последняя версия), это можно сделать буквально одним щелчком кнопки мыши в штатном менеджере пакетов дистрибутива.

## Оптимизированные решения

Но у пакетов есть один недостаток. Пакет компилируется под определенный тип процессора, и, чтобы охватить как можно большее количество компьютеров, разработчики дистрибутивов компилируют систему под минимально возможный процессор. Наиболее часто это архитектура i586, но нередко встречается и i386. Иными словами, возможности современных процессоров используются не полностью. В этом нет ничего страшного, но многим такая ситуация не нравится, и для решения проблемы были созданы так называемые дистрибутивы source-based (от англ. «базирующиеся на исходных текстах»). В них система пакетов не используется вообще, и сам процесс сборки программ из исходных текстов максимально автоматизирован. Пользователю для установки достаточно указать, что ему нужно, а все остальное — загрузку из Интернета, распаковку архива, конфигурирование, доустановку дополнительных приложений — система сделает сама.



### ПРИМЕЧАНИЕ

Технология автоматизированной установки при помощи исходных текстов уже много лет применяется в BSD-системах — FreeBSD, OpenBSD, NetBSD и др. В Linux она появилась относительно недавно.

В дистрибутивах source-based программы (и сама система) собираются под конкретное оборудование, что существенно повышает производительность системы. Яркие представители — Gentoo или Lunar-Linux. Есть и промежуточные решения между source-based и пакетными дистрибутивами — CRUX и Arch Linux.

Кроме того, дистрибутивы Linux различаются между собой:

- поставкой — от небольших решений, включающих только базовую систему и занимающих одну дискету, до многодисковых, размещающихся на двух-трех DVD;
- специализацией — есть специальные сборки, которые предназначены для какой-то конкретной аудитории, например: игровые (linuX-gamers Live), образования (Edubuntu) — для студентов и инженеров, занимающихся электрическими схемами Fedora Live FEL (Fedora Electronics Lab); есть дистрибутивы для работы с мультимедиа (Ubuntu Studio), создания «умного» дома (Linux MCE) и т. д.;

- внутренним устройством — сценарии загрузки (System V или BSD), соответствующие стандартам Linux Standard Base (<http://www.linux-foundation.org/en/LSB>) и Filesystem Hierarchy Standard (<http://proton.pathname.com/fhs/>).

Отдельно хотелось бы сказать о локализации Linux. Сегодня доступны дистрибутивы или локализованные сборки, которые разрабатываются в России, и поэтому их рабочая среда русифицирована изначально — Calculate Linux (<http://www.calculate-linux.ru/>), ALT Linux (<http://www.altlinux.ru/>), ASPLinux (<http://www.asplinux.ru/>), Runtu (<http://runtu.org/>), MOPSLinux (<http://www.mopslinux.org/>), Puppy Rus (<http://www.puppylinux.org/>) и Russian Fedora Remix (<http://www.russianfedora.ru/>). Но проблем с локализацией не будет и в большинстве других решений.

## Дистрибутив Linux Mint

История появления Linux Mint (<http://linuxmint.com/>, рис. 1.8) выглядит даже несколько банально. Ирландец Клемент Лефевр, проработавший в Linux более 10 лет и помогавший начинающим пользователям на различных форумах, принял решение создать свой дистрибутив, в котором максимально решались бы все возможные проблемы, с которыми обычно сталкиваются новички.



Рис. 1.8. Сайт проекта Linux Mint

За основу был взят набиравший в то время популярность дистрибутив Ubuntu (<http://www.ubuntu.com/>). Сегодня Ubuntu стабильно занимает первое место по рейтингу сайта Distrowatch.com, но, очевидно, сделать массовый дистрибутив, который бы устраивал всех — от новичка до пользователя с опытом, сложнее.

У каждого свои требования и запросы. Вот здесь и появляется Linux Mint (от англ. mint — «мята»).

Первый релиз, построенный на базе Kubuntu Dapper 6.06 LTS и называвшийся Ada, появился в августе 2006 г., прошел практически незамеченным (как и следующий). Первые версии Linux Mint представляли собой дистрибутив Kubuntu (версия Ubuntu с рабочей средой KDE) и имели все необходимые мультимедиа кодеки, популярные плагины вроде Flash для браузеров, поддерживали воспроизведения зашифрованных DVD, плюс UNRAR, Java, Real Player и др., то есть все, что разработчики Kubuntu не включали в свой дистрибутив по лицензионным соображениям (все легко устанавливалось из репозитория). Проприетарные драйвера в комплект не включают, так как это противоречит принципам разработчиков, но их легко установить самостоятельно. В итоге пользователю уже нужно меньше прилагать усилий для дальнейшей доводки дистрибутива. Со временем о Mint заговорили, но популярность пришла не сразу. Возможно, он так бы и затерялся среди прочих клонов Ubuntu, которые в то время появлялись как грибы после дождя, но собравшаяся к тому времени группа разработчиков написала несколько оригинальных приложений (mintInstall, mintUpdate, mintBackup, mintConfig, mintAssistant, mintMenu, mintDesktop и др.), упрощающих настройку и использование системы. Саму рабочую среду максимально упростили, сделав ее менее запутанной, плюс устранили некоторые недочеты.

В итоге в обзоре сайта Distrowatch.com за июль 2007 г. дистрибутив Linux Mint назван самым неожиданным решением 2007 г. Девиз проекта From freedom came elegance, который переводится как «из свободы приходит элегантность», полностью соответствует подходу разработчиков.

Еще один несомненный плюс Mint заключается в том, что разработчики не стали изобретать велосипед, а в качестве основы используют наработки Ubuntu. В частности, Linux Mint использует те же репозитории, что и релиз Ubuntu, на кодовой базе которого он построен. В итоге разработчикам не нужно беспокоиться о поддержке своего репозитория пакетов, а все усилия можно направить на доводку и тестирование системы. Основной репозиторий пакетов Linux Ubuntu уже насчитывает более 20 тыс. пакетов, и их список постоянно растет. Кроме того, в Интернете доступно несколько альтернативных репозитариев для Ubuntu (например, Medibuntu (Multimedia, Entertainment & Distractions In Ubuntu, <http://www.medibuntu.org/>)), которые также может подключить пользователь Linux Mint и затем устанавливать приложения.

Дистрибутив развивается очень динамично, не в последнюю очередь благодаря тому, что разработчики активно контактируют с пользователями и прислушиваются к их мнению. Поэтому идей для дальнейшего улучшения много, а такая обратная связь хорошо сказывается на имидже дистрибутива.

**ПРИМЕЧАНИЕ**

Если какие-то термины непонятны, ничего страшного, по ходу книги они будут раскрыты подробнее.

Дистрибутив Ubuntu выходит два раз в год (обычно в апреле и октябре), изначально планировалось, что между релизами Ubuntu будет несколько выпусков Linux Mint. Но разработчики отказались от такого плана выпуска, и сегодня релизу Linux Ubuntu соответствует один релиз Linux Mint. Начиная с версии 5 релизы обозначаются целым числом. Кроме того, каждый релиз получает женское имя, при этом имена следуют в алфавитном порядке. Текущая версия Linux Mint 9, вышедшая в мае 2010 г., носит имя Isadora.

## Глава 2

# Установка Linux Mint

Подготовка жесткого диска

Версии Linux Mint

Программа установки Linux Mint

Альтернативные варианты установки

Загрузчик GNU GRUB

Программа установки любого дистрибутива Linux, ориентированного на конечного пользователя, создана с учетом того, что установить систему на жесткий диск придется новичку. Весь процесс упрощен, и многие операции, обычно вызывающие проблемы, скрыты, поэтому в самом простом случае достаточно правильно отвечать на вопросы мастера, не вникая в подробности. Но обычно на компьютере уже установлена операционная система Windows, и пользователь хочет сохранить ее и работать, используя двойную загрузку. Чтобы быть готовым к любым неожиданностям, вначале следует рассмотреть особенности размещения Linux на жестком диске.

## Подготовка жесткого диска

У опытных пользователей Windows жесткий диск разбит на два и более раздела. Обычно все системные данные размещаются на диске C, а пользовательские — на других разделах жесткого диска. Это позволяет в случае необходимости легко переустановить операционную систему, ведь важная информация, а также видео, аудиофайлы, рисунки и т. д. находятся на разделах, которые не нужно форматировать при переустановке. Кроме того, такое разделение упрощает навигацию, ведь при современных объемах дисков без какой-либо систематизации найти что-либо будет невозможно. Сами разделы формируются в одну из файловых систем — FAT или NTFS.

Файловая система Linux настолько сильно отличается от Windows, что новичок, впервые столкнувшись с Linux, некоторое время буквально испытывает шок. В дистрибутивах, ориентированных на начинающего пользователя, максимально стараются хоть как-то сгладить эти отличия, имитируя структуры каталогов Windows и пряча все настройки за графическими утилитами. Но такие косметические ухищрения не помогут, если пользователю самостоятельно приходится администрировать компьютер, в этом случае все равно придется вникать в особенности файловой системы Linux.

В Linux также возможно располагать части операционной системы на разных разделах жесткого диска, но принцип их размещения абсолютно не такой, как в Windows. Кроме того, ядро поддерживает на порядок большее количество файловых систем. С этим и попробуем разобраться.

## Обозначение дисков в Linux

Разделы жесткого диска в Linux обозначаются по-другому. Если в Windows пользователь имеет дело с буквенными обозначениями C и D, то при добавлении нового диска их количество увеличивается. Каждый из таких каталогов существует как бы сам по себе. Это удобно, поскольку отсутствие строгой иерархии позволяет

пользователю хранить данные в любом месте. Но при некоторых обстоятельствах могут появляться и проблемы. Например, при добавлении нового диска иногда бывает, что буквы разделов перемешиваются и раздел, который назывался **D**, может стать **E**. Иными словами, программы и ярлыки, которые ссылались на такой раздел, будут недействительными.

В Linux такой проблемы нет. Традиционно в этой системе ATA-диск обозначается в соответствии с тем, к какому из интерфейсов он подключен. Например, диск, подключенный к первому IDE-диску, на первом канале всегда обозначается как `/dev/hda`, на втором канале — `/dev/hdb`. Соответственно диск, подключенный ко второму IDE-диску, на первом канале имеет обозначение `/dev/hdc`, а на втором — `/dev/hdd`. Причем называться он так будет независимо от того, есть ли диск в устройстве на данный момент или нет. Так обозначается весь диск целиком. Диски, имеющие интерфейс SCSI, который редко использовался на настольных системах, отличались первой буквой, то есть вместо **h** стояла **s**, а полное название первого SCSI-диска — `/dev/sda`. Такое обозначение стало использоваться и в дисках, имеющих SATA-интерфейс (англ. Serial ATA). В итоге одна часть дисков могла обозначаться как `/dev/hd*`, другая — `/dev/sd*`.



#### ПРИМЕЧАНИЕ

В различной литературе компьютерной тематики авторы для краткости используют шаблоны, применяемые в регулярных выражениях. Так, знак «\*» обозначает всю возможную совокупность символов. Хотя правильной было бы записать `/dev/sd[a-d]`, то есть после `sd` могут представляться все символы, которые входят в диапазон в скобках.

Начиная с 2007 г. в связи с внедрением библиотеки `libsata` все дистрибутивы перешли на одинаковое обозначение для ATA, SATA, USB и прочих дисков — `/dev/sd*`. Поэтому сегодня никакой путаницы нет.

## Обозначение разделов в Linux

Как и принято, чтобы пользователю было удобно работать, диск обычно разделяют на разделы меньшего размера. Согласно спецификации жесткий диск может иметь не более четырех первичных (англ. primary) разделов. В Linux за первичными разделами закреплены цифры от 1 до 4, то есть первый первичный раздел первого диска обозначается как `/dev/sda1`. В современных условиях четырех разделов может не хватать, поэтому для увеличения количества разделов в одном (и только одном) из первичных создают так называемый расширенный (extended) раздел, который использует свою собственную таблицу разделов. В итоге это позволяет создать на расширенном разделе еще несколько логических. Логические разделы в Linux начинают обозначаться цифрами, начиная от 5, которые следуют по порядку. Например, первый логический раздел будет всегда называться `/dev/sda5`, следующий — `/dev/sda6` и т. д.

**ПРИМЕЧАНИЕ**

В некоторых операционных системах Unix — FreeBSD, OpenBSD и др. — логические разделы (BSD Partitions) можно создать внутри каждого первичного, который именуется как *слайс* (англ. *slices*).

Итого в идеальном случае мы можем иметь на диске три первичных раздела, например `/dev/sda1` — `/dev/sda3`, и несколько логических, начиная от `/dev/sda5` и до теоретических 63-х разделов (в SCSI-дисках — максимум 15), которые размещаются на четвертом первичном.

**ПРИМЕЧАНИЕ**

Это не единственная система наименования, с которой сталкивались пользователи Linux. В ядре Linux 2.4.\* появилась новая файловая система устройств `devfs`, изобретавшая разработчиков от многих неудобств, но использовавшая новую систему наименования. Так, второй раздел первого жесткого диска имел длинное название `/dev/ide/host0/bus0/target0/lun0/part2`, и, чтобы упростить задачу, пользователю создавались ссылки со старыми названиями `/dev/sda2`.

## Что такое UUID?

Предположим, Linux установлен и работает, но возникла идея разделить еще один раздел, чтобы смонтировать его в свою точку подключения, или подключить еще один жесткий диск, который система упорно хочет видеть первым. В таком случае система может вообще не загрузиться, ведь все конфигурационные файлы, ядро и т. д. были раньше размещены, например, в разделе, отмеченном как `/dev/sda2`, который теперь стал `/dev/sda3`, то есть придется менять настройки в других конфигурационных файлах, чтобы нормально работать. Нашли простой выход. Каждый раздел получает свой уникальный постоянный идентификатор UUID (Universally Unique Identifier), и именно идентификатор, а не имя раздела используется часто в конфигурационных файлах. В случае каких-либо изменений нужный раздел будет найден по его UUID, хотя при желании можно переписать все по-старому.

Узнать соответствие UUID разделу можно, прочитав содержимое каталога `/dev/disk/by-uuid/`.

```
$ ls -l /dev/disk/by-uuid/
```

```
lrwxrwxrwx 1 root root 10 2010-05-03 11:24 0DC1-0000 -> ../../sda1
```

```
lrwxrwxrwx 1 root root 10 2010-05-03 11:24 8aeec190-857a-4d7b-9065-a3de316f72f6 -> ../../sda2
```

```
lrwxrwxrwx 1 root root 10 2010-05-03 11:24 ff8becfc-d3c4-4956-afb0-c4f4 55bb0ba3 -> ../../sda5
```

Например, разделу `/dev/sda5` соответствует UUID — `ff8becfc-d3c4-4956-afb0-c4f455bb0ba3`.

Файловые системы FAT и NTFS не поддерживают UUID, хотя для них вполне возможно образование уникального имени, но оно получается коротким. В примере оно показано первым — `0DC1-0000`.

## Разделы для установки Linux

Итак, с обозначением разобрались. Теперь разберем минимальное количество самих разделов, которое необходимо для работы операционной системы Linux. В Linux файловая структура связана в логическое дерево. Путь каждого файла и каталога начинается от корня файловой системы, который обозначается знаком «/». При обращении к файлу указывается полный путь относительно корня файловой системы или относительный — от текущего местонахождения. Например, запись `/boot/vmlinuz` указывает в нашем случае на файл `vmlinuz`, размещенный в каталоге `/boot`.



### ПРИМЕЧАНИЕ

Каталог — это файл, содержащий список связанных с ним файлов.

На уровень ниже от корневого находится большая группа каталогов, каждый из них играет свою роль в системе, их названия и назначения стандартизированы, хотя в некоторых случаях разработчики дистрибутивов отходят от стандарта. В терминале список файлов и каталогов можно просмотреть командой `ls` название каталога. Чтобы просмотреть список подкаталогов, находящихся в верхушке корня (рис. 2.1), используем

```
$ ls /
```

Вот некоторые из таких каталогов — `/boot`, `/bin`, `/sbin`, `/etc`, `/lib`, `/home`, `/root`, `/mnt`, `/usr`, `/var`, `/dev`, `/opt`, `/proc` и `/tmp` (см. рис. 2.1).

Чтобы было понятно, разберем назначение основных каталогов Linux. Для сравнения: в операционной системе Windows после установки программы все конфигурационные и исполняемые файлы могут находиться где попало: в каталоге, в который установлена программа, в одном из системных каталогов, в текущем каталоге, откуда произведен первый запуск, и т. д.



### ПРИМЕЧАНИЕ

В терминологии Linux процедура подключения устройства к дереву каталогов называется монтированием, соответственно отключение — размонтированием.

```

mint@mint ~$ ls -al /
total 2
drwxr-xr-x 30 root root 260 2010-05-01 18:06 .
drwxr-xr-x 30 root root 260 2010-05-01 18:06 ..
drwxr-xr-x 2 root root 2207 2010-01-14 10:44 bin
drwxr-xr-x 5 root root 60 2010-01-31 11:31 boot
drwxr-xr-x 8 root root 2048 2010-01-31 12:18 cdrom
drwxr-xr-x 15 root root 3620 2010-05-01 18:07 dev
drwxr-xr-x 150 root root 720 2010-05-01 18:07 etc
drwxr-xr-x 3 root root 60 2010-05-01 18:06 home
drwxr-xrwx 1 root root 33 2009-12-17 08:37 initrd.img -> boot/initrd.img-2.6.31-16-generic
drwxr-xr-x 20 root root 4193 2010-01-14 18:44 lib
drwxr-xr-x 2 root root 48 2009-10-28 21:39 media
drwxr-xr-x 2 root root 3 2009-10-20 00:04 net
drwxr-xr-x 2 root root 3 2009-10-20 21:32 opt
drwxr-xr-x 139 root root 6 2010-05-01 18:06 proc
drwxr-xr-x 20 root root 317 2009-12-17 09:04 rafs
drwxr-xr-x 5 root root 156 2010-01-31 11:21 root
drwxr-xr-x 2 root root 3165 2010-01-14 17:22 sbin
drwxr-xr-x 2 root root 3 2009-10-19 23:05 selinux
drwxr-xr-x 2 root root 3 2009-10-20 21:32 srv
drwxr-xr-x 12 root root 0 2010-05-01 18:06 sys
drwxrwxrwt 11 root root 240 2010-05-01 18:16 tmp
drwxr-xr-x 14 root root 120 2009-10-20 21:32 usr
drwxr-xr-x 20 root root 160 2009-10-20 21:37 var
drwxr-xrwx 1 root root 30 2009-12-17 08:37 vmlinuz -> boot/vmlinuz-2.6.31-16-generic
mint@mint ~$

```

Рис. 2.1 Дерево файловой системы Linux Mint

Причем порядок не является устойчивым и зависит от конкретного разработчика. В итоге даже при удалении программы нельзя точно сказать, что где находится. В Linux, как и в прочих UNIX-подобных системах, расположение основных файлов четко определено, основные исполняемые файлы размещаются в одних каталогах, конфигурационные — в других, библиотеки — в третьих и т. д.

Чтобы избежать путаницы, разработан стандарт, определяющий имена и назначение основных каталогов и подкаталогов, а также некоторых важных файлов, — **Filesystem Hierarchy Standard** (FHS, Стандарт иерархии файловой системы).



#### ПРИМЕЧАНИЕ

Последнее описание стандарта FHS можно найти на сайте проекта <http://www.pathname.com/fhs/>, перевод доступен на сайте OpenNET <http://www.opennet.ru/docs/RUS/file-sys/>.

Текущая версия стандарта 2.3 вышла в январе 2004 г. В данный момент большинство UNIX-подобных систем, в том числе и дистрибутивов Linux, в той или иной степени следуют этим правилам. Конечно, в дистрибутивах встречаются исключения и варианты, но они не настолько радикальны, чтобы в них не разобраться.

Да и часто ради совместимости разработчики используют символические ссылки. Некоторые рекомендации стандарта, например размещение домашних каталогов серверов в `/srv`, так и не нашли широкого распространения. Есть дистрибутивы, которые полностью отвергают рекомендации FHS, например GoboLinux (<http://gobolinux.org/>). Но повторюсь, исключений не так уже и много. Поэтому, меняя дистрибутив, версию системы, создавая свой скрипт и т. д., пользователь всегда будет находить все на своих местах.

Прежде чем разбираться с тем, куда и как подключаются устройства, думаю, не лишним будет разобраться с назначением основных каталогов. При разделении каталогов разработчики стандарта руководствовались простым принципом — собрать в одном месте файлы по назначению или типу использования. Нетрудно заметить, что изначально разделены системные и пользовательские файлы, файлы, которые остаются неизменными, и изменяющиеся самостоятельно в процессе работы системы, разделяемые и неразделяемые файлы. Это дает возможность подключить разделы с системными файлами, которые редко изменяются в режиме для чтения, выделив минимально необходимый объем. Такой подход позволяет сэкономить дисковое пространство и уберечь файлы от ошибок на случай сбоев и прочих неприятностей (например, атаки вирусов). Специальные программы могут следить за любыми попытками вмешательства в такие каталоги и выявлять попытки взлома или вирусной атаки без применения антивирусных программ. Каталоги, в которых помещены разделяемые ресурсы, можно использовать совместно несколькими компьютерами. Администратору легче следить за обновлениями, да и место на жестких дисках не расходуется зря.

Каталог `/bin` содержит основные исполняемые файлы, которые доступны как обычному пользователю, так и системным администраторам. Отсюда и название, которое произошло от англ. *binaries* — «двоичный, исполняемый». Некоторые из таких команд используются стартовыми скриптами для установки системных параметров даже в том случае, если система работает в однопользовательском (спасательном) режиме, когда остальные файловые системы не смонтированы. Поэтому FHS устанавливает список обязательных файлов, которые должны присутствовать здесь. Если в дистрибутиве применяется аналог одной из подобных программ, обязательно должна присутствовать ссылка с таким именем. Например, файл `/bin/sh` в большинстве случаев является ссылкой на исполняемый файл командной оболочки Борна `/bin/bash`.

Аналогично в `/sbin` (*system binaries*) содержатся системные файлы, доступ к которым имеет только пользователь с правами администратора. Здесь находятся важные системные программы, предназначенные для настройки системы и сети, работы с дисками, учетными записями пользователей и др.

В каталоге `/boot` собраны файлы, необходимые для загрузки ядра, и, конечно, само ядро. Данный каталог меняется только при обновлении ядра, поэтому часто

его помещают на отдельном разделе жесткого диска небольшого размера (до 100 Мбайт). В системах, использующих загрузчик GRUB, обычно находятся конфигурационный файл загрузчика и некоторые файлы, необходимые для его работы.

Файлы устройств собраны в каталоге `/dev`, название которого и произошло от англ. `device` — «устройство». Постепенно мы будем знакомиться с некоторыми важными файлами этого каталога. Как раз на примере каталога `/dev` видно, что все в Linux является файлами. Например, файл `/dev/dsp` является устройством вывода звука. Если звуковая подсистема уже настроена, это можно легко проверить. Выполните, например, команду

```
$ cat /boot/vmlinuz-2.6.32-21-generic > /dev/dsp
```

где `/boot/vmlinuz` — это файл, содержащий ядро Linux, и вы услышите, как «играет» ядро. Конечно, на самом деле все файлы не являются устройствами как таковыми, а обеспечивают лишь прием и передачу информации, скрывая от пользователя внутреннюю структуру. Есть здесь и «черная дыра» — файл `/dev/null`. Вся информация, отправленная в этот файл, пропадает бесследно. Его часто используют в командах, например, чтобы не засорять вывод на терминал ненужными сообщениями. Так, чтобы убрать сообщения об ошибках, выводимых при копировании файлов, нужно использовать команду

```
$ cp -R folder/* folder2 2>/dev/null
```

или, чтобы удалить содержимое файла, но при этом сохранить сам файл со всеми его атрибутами (владелец, права доступа), команду

```
$ cat /dev/null > /var/log/messages
```

Есть еще интересное псевдоустройство — файл `/dev/zero`, отличие которого от `/dev/null` состоит в том, что он содержит двоичные нули. Информация, отправленная в `/dev/zero`, исчезает бесследно, попытка прочитать ее может привести к некоторым трудностям. Его часто используют для создания пустого файла заданного размера. Например, чтобы создать файл размером 1 Мбайт:

```
$ dd if=/dev/zero of=file bs=1024 count=1000
```

Если нужен поток случайных чисел, следует обратиться к псевдоустройствам `/dev/random` и `/dev/urandom`.

Основные конфигурационные файлы системы, установленных сервисов и некоторых программ, доступных ресурсов, учетные базы пользователей и др. находятся в каталоге `/etc`. Новые программы, устанавливаемые при помощи пакетов, также будут размещать конфигурационные файлы в каталоге `/etc`.

**ПРИМЕЧАНИЕ**

Если программа компилируется самостоятельно, то место, где разместятся конфигурационные файлы, будет определено при конфигурировании в зависимости от установок переменных `--sysconfdir` и `--prefix`.

Спецификация определяет и некоторые каталоги внутри `/etc`. Например, в `/etc/X11/` находятся настройки видеоподсистемы Linux XOrg или XFree86, о настройках которой мы поговорим в следующих главах.

Настройки и личные файлы обычных пользователей находятся в подкаталогах, которые, как правило, соответствуют названию учетной записи и размещаются в каталоге `/home`. При правильных настройках системы обычный пользователь не сможет подняться выше по дереву каталогов из своего домашнего каталога. Часто для `/home` выделяют отдельный раздел жесткого диска, который можно подключать в разных дистрибутивах.

Аналогично `/root` является домашним каталогом пользователя `root`. Учитывая, что этой учетной записью в повседневной работе не пользуются, каталог `/root` имеет небольшой размер, и по возможности он должен находиться на том же разделе жесткого диска, что и корневой каталог.

Важные системные библиотеки находятся в каталоге `/lib` (от англ. libraries — «библиотеки»). В файлах реализованы стандартные функции (например, ввод/вывод данных), необходимые в основном для работы программ из каталогов `/bin` и `/sbin`.

**ПРИМЕЧАНИЕ**

Часто используемые функции обычно выносят в отдельную библиотеку, что позволяет использовать их разными программами, экономить место на диске и упростить их обновление и сопровождение.

В каталоге `/lib/modules/`uname -r`` находятся установленные модули ядра.

**ПРИМЕЧАНИЕ**

На самом деле каталог называется по-другому, и в каждом дистрибутиве по-разному, поэтому в примере использована возможность подстановки одной команды в другую. Команда `uname -r`, выполненная в консоли, как раз и покажет имя текущего ядра, то есть в нашем случае — `2.6.32-21-generic`. Поэтому во многих примерах, чтобы упростить поиск, используют конструкцию с выводом этой команды.

Для монтирования временных файловых систем предназначен каталог `/mnt`, а в версии FHS сменные носители рекомендуется монтировать в каталог `/media`. Именно в использовании этих двух каталогов в разных дистрибутивах есть расхождения. В идеальном случае, когда Linux установлен в качестве единственной системы, в `/media` должен быть только каталог для монтирования CD/DVD-привода. Но на домашнем компьютере Linux часто устанавливается в качестве второй системы, поэтому после инсталляции в `/media` или `/mnt` помещаются ссылки для

монтирования остальных разделов жесткого диска, на которых помещены данные операционной системы Windows. Поскольку стандарт четко не определяет для таких случаев точку монтирования, решение этого вопроса ложится на разработчиков. Например, в Ubuntu сторонние файловые системы и сменные носители по умолчанию монтируются в /media, а в Slackware — в /mnt. В KNOPPIX каталоги для монтирования можно найти и в /mnt, и в /media, хотя по умолчанию монтирование производится в /media. Попадаются и решения, когда точка для монтирования привода компакт-диска вынесена в корневой каталог — /cdrom. Хотя большой путаницы обычно это не вызывает.

Псевдофайловой системы /proc на самом деле нет на жестком диске, она расположена в виртуальной памяти и монтируется во время работы операционной системы. Большинство системных команд берет информацию именно отсюда. Например, команда `lsmod`, предназначенная для просмотра загруженных модулей ядра, на самом деле выводит результат команды

```
$ cat /proc/modules
```

в более удобной форме. Аналогично и `lspci`, показывающая список PCI-устройств, получает информацию из файла /proc/pci. Если просмотреть список каталогов внутри /proc, можно заметить, что большая часть имеет цифровое обозначение. На самом деле эти имена соответствуют идентификационному номеру процесса (process ID, PID). Проверим.

Запустим консоль и посмотрим связанный с ней PID.

```
$ ps a | grep /bin/bash  
5800 pts/1 Rs 0:00 /bin/bash
```

Первая цифра соответствует PID процесса. Теперь при помощи команды `ls` проверим, что находится в подкаталоге /proc с таким номером.

```
$ ls /proc/5800
```

Вывод покажет большое количество файлов, в которых содержится та или иная информация о процессе. Например, в `cmdline` находится командная строка запуска этого процесса:

```
$ cat /proc/5800/cmdline  
  
/bin/bash
```

Временные файлы, в которых программы хранят промежуточные данные, находятся в каталоге /tmp. В большинстве случаев после корректного завершения работы программы все временные файлы удаляются, а сам каталог автоматически очища-

ется при загрузке системы. Для ускорения доступа часто пользователи монтируют этот каталог в оперативную память. Как это делается, разберем чуть позже.

В каталоге `/usr` размещены прикладные программы, то есть сюда вынесено все, что не нужно при загрузке системы. Именно в `/usr` при помощи систем управления пакетами устанавливаются приложения, серверы и прочие программы и библиотеки. Структура подкаталогов в `/usr` немного напоминает корневой раздел. Здесь есть подкаталоги `bin`, `sbin`, `etc` и `lib`, назначение которых совпадает с одноименными, только относятся они к прикладным, а не к системным программам. С учетом большого количества программ, доступных в современных дистрибутивах, этот каталог желательно размещать на отдельном разделе жесткого диска. Иначе возникнет ситуация, когда, установив новые программы и перезагрузившись, вы не сможете зарегистрироваться, так как все место на корневом разделе будет занято.

Внутри `/usr` есть ряд подкаталогов, которые вы часто будете встречать. Например, `/usr/src` используется для распаковки исходных текстов ядра. В каталоге `/usr/X11R6/` находятся файлы и библиотеки видеосервера Linux — X Window System. Каталог `/usr/local` предназначен для установки дополнительных программ. Именно сюда попадают программы при компиляции с параметрами по умолчанию. Он в основном повторяет структуру каталогов `/usr`.

В каталог `/opt` (от англ. *optional* — «опциональный») устанавливаются опциональные программы. Как правило, это программы, представленные сторонними разработчиками и не устанавливаемые при помощи системы пакетов конкретного дистрибутива. Обычно подкаталоги имеют собственную структуру, и, чтобы удалить программу из `/opt`, достаточно просто удалить соответствующий каталог.

И наконец, последний, но не менее важный каталог — `/var` (от англ. *variable* — «переменные»). В него помещаются данные, которые постоянно изменяются в ходе работы системы, но в отличие от `/tmp` они нужны для дальнейшей работы после перезагрузки или по окончании работы программы. Именно здесь находятся журналы работы программ, кэш-файлов и база системы управления пакетами, файлы баз данных, почтового и веб-сервера, очереди печати. Например, файлы журналов системы и отдельных программ и сервисов находятся в `/var/log`. В подкаталоге `/var/run` размещаются файлы с расширением `.pid`. Имя файла обычно указывает на создавший его процесс. При завершении работы программы `pid`-файл удаляется. Таким образом, наличие или отсутствие одноименного файла позволяет легко узнать, работает ли необходимая программа. Нужно это, например, для блокировки запуска ее второй копии или проверки состояния необходимых сервисов, от работоспособности которых зависит запускаемая программа.

Несколько иное назначение каталога `/var/lock`, появление здесь файла говорит о блокировке ресурса или программы. Например, происходит автоматическое обновление списка пакетов в репозитории, а в это время администратор пробует установить

новую или обновить имеющуюся программу. В процессе обновления список программ может поменяться, и в результате пользователь, вполне вероятно, окажется с нерабочей программой или системой. Чтобы избежать возможных разногласий в подобной ситуации, используется файл блокировки. Пока будет производиться обновление базы пакетов, все другие попытки обращения к ней будут блокироваться.

#### ВНИМАНИЕ



В Ubuntu и Linux Mint ранних версий раздел `/var/gup` обязательно должен находиться на корневой файловой системе, иначе некоторые приложения не смогут стартовать.

Наверное, сначала такая схема покажется непривычной, неудобной, непонятной, но не спешите с выводами, вскоре вы поймете ее простоту. Тем более что в большинство каталогов, которые описаны выше, вы, скорее всего, так и не заглянете. Большую часть настроек вы будете производить в домашнем каталоге пользователя и `/etc`.

#### СОВЕТ



Чтобы не повторять настройки при переустановке, обновлении системы или переходе на другой дистрибутив, сохраняйте резервную копию каталога `/etc` и используйте готовые файлы. Если формат совместим, то вы быстро получите работоспособную систему.

Кроме этого вы сами можете добавить свой каталог в абсолютно любое место файловой системы, если обладаете соответствующими правами доступа и если такое действие имеет смысл. Так, домашний каталог пользователя является подкаталогом в `/home` и обычно соответствует логину, который он использует при регистрации в системе. Например, домашний каталог пользователя, имеющего логин `user`, будет находиться в `/home/user`. Именно здесь пользователь хранит свои данные.

#### ВНИМАНИЕ



При обращении к каталогу или файлу всегда указывается путь относительно корня. Например, чтобы обратиться к каталогу `boot`, следует писать `/boot`. Исключением является использование специальных переменных. Например, чтобы обратиться к файлу `file`, находящемуся в домашнем каталоге пользователя `grinder`, можно указывать полный путь `/home/grinder/file`. На домашний каталог пользователя можно указать при помощи переменных `$HOME` или `~`, то есть в нашем примере для пользователя `grinder` запись будет выглядеть так: `$HOME/file` или `~/file`.

Главная особенность Linux состоит в том, что любой из каталогов дерева файловой системы не обязан находиться на одном разделе жесткого диска, и чтобы части, размещенные на других разделах жесткого диска, были доступны операционной системе или пользователю, они монтируются (англ. *mounting*), то есть фактически подключаются к дереву каталогов. При этом существует полная свобода выбора:

дисковый раздел можно смонтировать в любую точку файловой системы. Хотя на первый взгляд все выглядит сложно, на самом деле монтирование — простая операция, а гибкость, которая предоставляется пользователю в построении своего рабочего пространства, поверьте, будет вами оценена. По ходу прочтения книги мы разберем все эти моменты более подробно, и вопросов не останется.

Итак, для нормальной работы Linux необходимо создать как минимум два раздела: системный (Linux native) и раздел подкачки (Linux Swap). Для установки Linux Mint под системный раздел необходимо выделить не менее 3 Гбайт, но это с учетом того, что пользователь будет хранить данные в этом же разделе. Иными словами, если в домашнем каталоге планируется хранить пользовательские данные, например коллекцию аудио- и видеозаписей, то следует сразу предусмотреть объем и для них.

Раздел подкачки желательно расположить как можно ближе к началу диска для увеличения скорости обмена данными, а идеальный вариант — на другом физическом диске, чтобы данные и кэшированная информация считывались с разных носителей. Это увеличит скорость работы. Можно использовать еще один вариант: разместить раздел подкачки на разных дисках, указав одинаковый приоритет в специальном файле `/etc/fstab`.

```
/dev/sda1 swap swap defaults,pri=1 0 0
```

```
/dev/sdc5 swap swap defaults,pri=1 0 0
```

Размер раздела подкачки зависит от количества оперативной памяти, установленной в системе, и задач, которые планируется выполнять на компьютере. При небольшом количестве оперативной памяти, приблизительно 256 Мбайт, размер раздела подкачки можно выбрать кратный трем. Если количество оперативной памяти больше 1 Гбайт, то, как правило, трехкратный размер раздела не имеет смысла.

#### ВНИМАНИЕ



Жесткий диск имеет меньшую производительность по сравнению с оперативной памятью, поэтому частое обращение системы к разделу подкачки скажется на скорости выполнения операций. В таком случае следует увеличить количество оперативной памяти или оптимизировать систему, отключив лишние сервисы.

Для корректной работы режима сна (гибернации) требуется, чтобы размер раздела подкачки был на 10 % больше количества оперативной памяти, установленной на компьютере. Этой рекомендации и нужно придерживаться.

#### ПРИМЕЧАНИЕ



В Linux кроме раздела подкачки можно создать и файл подкачки (англ. `swap file`). Если в некоторый момент размера оперативной памяти с разделом подкачки не будет хватать, то такой файл можно создать и подключить в работу по мере необходимости.

Два раздела — самый простой вариант, который часто используют новички, желающие познакомиться с системой. Опытные пользователи в домашних системах, как правило, используют три раздела. В третьем разделе располагаются домашние каталоги пользователей `/home`. В этом случае не возникнет ситуации, когда из-за переполненности диска (например, был скопирован большой фильм, занявший все свободное место) система не смогла загрузиться в обычном режиме. При таком разбиении в случае переустановки дистрибутива (конечно, если раздел, отведенный под `/home`, не форматировался) все пользовательские установки будут сохранены. Кроме этого все новые программы устанавливаются в каталог `/usr`, в него же копируются исходные тексты ядра, если будет спланирована самостоятельная переборка. Поэтому следует либо предусмотреть больше места в корневом разделе, либо сразу отвести под `/usr` отдельный раздел.

#### ВНИМАНИЕ



В старых BIOS было еще одно ограничение — ядро Linux можно было загрузить только с первых 1023 цилиндров жесткого диска. Тогда эту проблему решали выносом раздела `/boot` ближе к началу. Теперь об этом можно не беспокоиться.

Вынос загрузочного каталога `/boot` (не более 50-100 Мбайт) на отдельный дисковый раздел также может преследовать другую цель. Чтобы обеспечить безопасность и сохранность ядра, раздел `/boot` монтируют при загрузке в режиме «только чтение». Если же понадобится установить новое ядро, то раздел затем очень просто перемонтировать в режим «чтение — запись».

Минимальным требованием для корневого раздела является 3 Гбайт, рекомендуется — 8 Гбайт. Если планируется использовать отдельно `/home` с учетом того, что на нем размещаются пользовательские данные, под него следует выделить как можно больше места.

## Файловые системы Linux

Еще один момент, с которым придется столкнуться во время установки Linux, — выбор файловой системы. В отличие от Windows, которая предлагает отформатировать диск в NTFS или FAT32, а в последних версиях — только в NTFS, в Linux выбор гораздо больше. Современное ядро поддерживает следующие файловые системы, на которых можно установить Linux, — ext2, ext3, ext4, ReiserFS, XFS и JFS. А список всех файловых систем, разделы которых может монтировать Linux, еще больше, в том числе в этом списке находятся и файловые системы, считающиеся родными для других операционных систем, в частности Windows.

#### ПРИМЕЧАНИЕ



На заре своего развития Linux работал с файловой системой Minix, на замену которой и пришел ext2. Minix до сих пор поддерживается ядром, но использовать ее нет смысла.

Просмотреть полный список можно в конфигурационном файле ядра (рис. 2.2):

```
$ grep -i fs /boot/config-`uname -r`
```

#### ПРИМЕЧАНИЕ



В командах командной оболочки можно использовать символы подстановки или выходы других команд, в данном примере в кавычки заключена команда, которая получит и подставит номер версии ядра Linux.

```
mint@mint: ~$ grep -i fs /boot/config-2.6.31-16-generic
# CONFIG_SYSFS_DEPRECATED_V2 is not set
CONFIG_INITRAMFS_SOURCE=""
CONFIG_PAGE_OFFSET=0xc0000000
CONFIG_ACPI_PROCFS=y
CONFIG_ACPI_PROCFS_POWER=y
CONFIG_ACPI_SYSFS_POWER=y
# CONFIG_NCA_PROC_FS is not set
CONFIG_NET_SCH_HFSC=y
CONFIG_CFG80211_DEBUGFS=y
CONFIG_WIRELESS_EXT_SYSFS=y
CONFIG_MAC80211_DEBUGFS=y
CONFIG_PARPIDFS_PROC_FS=y
CONFIG SCSI_PROC_FS=y
CONFIG SCSI_LPF_DEBUG_FS=y
# CONFIG_RT2800_LIB_DEBUGFS is not set
CONFIG_I80N_CAPI_CAPIFS_BLOCK=y
CONFIG_I80N_CAPI_CAPIFS=y
CONFIG_GPIO_SYSFS=y
CONFIG_SENSORS_FSCHEM=y
CONFIG_SENSORS_FSCHEM=y
CONFIG_SENSORS_FSCHEM=y
CONFIG_VIDEO_PVRUSB2_SYSFS=y
CONFIG_RADIO_TYTHON_PROC_FS=y
CONFIG_SND_VERBOSE_PROCFS=y
CONFIG_USB_DEVICEFS=y
# CONFIG_USB_GADGET_DEBUG_FS is not set
# CONFIG_USB_GADGET_FSL_USB2 is not set
# CONFIG_USB_GADGET_FSL_QC is not set
CONFIG_USB_GADGETFS=y
CONFIG_RTC_INTF_SYSFS=y
CONFIG_POWERS=y
# CONFIG_POWERS_DEBUG is not set
CONFIG_POWERS_CRYPTON=y
```

Рис. 2.2. Список файловых систем, поддерживаемых Linux Mint

Кроме этого для тех, кому нужна более быстродействующая или гибкая в настройках файловая система, можно создать программные RAID-массивы (идентификатор fd) или системы управления логическими томами — LVM (англ. Logical Volume Manager) (идентификатор 8e). Причем каждая из указанных файловых систем, которую можно выбрать во время установки, имеет свои особенности и, можно сказать, специализацию. И хотя разработчики обычно предлагают оптимальный вариант, подходящий в большинстве случаев, следует знать об этих отличиях — возможно, вы предпочтете другую файловую систему.

#### ПРИМЕЧАНИЕ



Под файловой системой понимается физический способ организации данных на дисковом разделе, то есть возможность хранения, нахождения и манипулирования (запись, перезапись и удаление).

Я думаю, такого простенького определения достаточно, чтобы понять, какие требования предъявляются к файловой системе. Теперь кратко разберем особенности основных файловых систем. Если описание покажется вам скучным, можете перейти к следующей части главы и вернуться сюда, когда возникнет проблема выбора файловой системы для работы.

## Файловая система ext2

Файловая система ext2 (сокр. Extended File System) долгое время была стандартной для Linux, ее характеризует высокая надежность и высокое быстродействие, которое, в свою очередь, достигается очень эффективным механизмом кэширования дисковых операций. По мере роста размеров раздела и количества файлов ext2 перестала удовлетворять современным требованиям по управлению такими большими объемами, устойчивости к сбоям и восстановлению после сбоя. Например, чтобы указать расположение блоков для больших файлов применялся метод тройной косвенной адресации. Выглядит это примерно так: если файл маленький, то в его метаданных содержится прямая ссылка на ячейки (логические блоки), в которых хранятся данные — прямая адресация. При увеличении объема файла отведенного места в метаданных уже не хватает, чтобы указать на занимаемое пространство, поэтому блоки метаданных указывают уже на косвенные блоки, в которых содержатся адреса с данными, определенными в файле, или, опять же, указатели на следующие косвенные блоки. И так до утроения. В случае сбоя, когда данные уже записаны, но изменения в адресации не перераспределены, файл можно потерять. Ситуация усугубляется тем, что данные часто записываются в новое место, а не поверх старых. Это уменьшает фрагментацию (увеличивает скорость доступа), но увеличивает вероятность потери данных, так как истинная информация в метаданных не будет соответствовать реальному расположению данных.



### ПРИМЕЧАНИЕ

Фрагментация данных на жестком диске появляется в том случае, когда при записи недостает нужного количества последовательных блоков данных, поэтому файл можно записать в разных частях жесткого диска. Однако при считывании головке жесткого диска приходится затрачивать большее время, что сказывается на быстродействии.

Если данные заносились в каталог (а это тоже файл), то после перезагрузки мы можем недосчитаться целого каталога.

Конечно, ситуация не так плачевна, как я обрисовал. Вероятность такого исхода очень мала, но самое главное — за время существования ext2 утилита проверки файловой системы fsck (сокр. FileSystem ChecK) была доведена почти до совершенства и практически безошибочно устраняет все проблемы. Кроме этого ext2 создает копии критически важных метаданных, поэтому вероятность полной потери данных чрезвычайно мала.

**ПРИМЕЧАНИЕ**

Чтобы не проверять при каждой перезагрузке системы все диски, установленные на компьютере, нашли такой простой выход. После того как все данные согласованы, непосредственно перед самым размонтированием раздела устанавливается бит чистого размонтирования (англ. `clean bit`), наличие которого и проверяет программа `fsck`. При наличии `clean bit` делается предположение, что файловая система в порядке, и проверка не производится.

Если бит чистого размонтирования не найден, то производится полная проверка раздела файловой системы. Вот здесь-то и кроется очевидное неудобство: чем больше файловая система, тем дольше длится процесс проверки. Во время загрузки процедура проверки метаданных на дисковом разделе размером в несколько сотен гигабайт, с большим количеством файлов и каталогов может очень сильно затянуться.

Но `ext2` все еще пользуется популярностью, особенно там, где важна скорость считывания данных в разделах, которые монтируются в режиме «только чтение». Например, в каталоге `/boot` находятся ядро и некоторые другие файлы, обеспечивающие загрузку системы. Содержимое этого раздела изменяется очень редко, некоторые администраторы Linux выносят его на небольшой раздел, который форматируется в `ext2` и монтируется в режиме «только чтение».

Осталось добавить, что `ext2` поддерживает файлы максимального размера в 2 Тбайт и раздел размером до 32 Тбайт. Обычно этого хватает для домашнего применения.

## Файловая система `ext3`

Основной недостаток `ext2` — долгое время проверки после сбоя — исправили в файловой системе `ext3`, которая является дальнейшим развитием `ext2`. Разработана `ext3` в компании Red Hat доктором Стивеном Твиди. Чтобы не изобретать колесо, в данном случае поступили просто: добавили к стандартной `ext2` журнал. Таким образом, основные характеристики `ext2` и `ext3` аналогичны, за исключением того, что `ext3` является журналируемой файловой системой. Утилиты проверки файловой системы для `ext2` поддерживают и `ext3`. К тому же идентичность файловых систем позволяет оперативно переходить как с `ext2` на `ext3`, так и наоборот, просто изменив параметры монтирования.

Вся магия журнала заключается в механизме транзакций, и в итоге вместо отслеживания всех модификаций операция записи на диск рассматривается как атомарная, а не разделенная на несколько этапов. Это позволяет отследить, прошла ли запись вообще, и, в свою очередь, гарантировать, что все или ни одно изменение файловой системы не сделано. Для этого перед выполнением операций файловая система создает описание того, что собирается сделать. Как только транзакция будет

зарегистрирована, файловая система приступает непосредственно к изменению метаданных. В случае системного сбоя файловая система будет восстановлена до непротиворечивого состояния путем повторного запуска журнала и отката к предыдущему состоянию. И главное — при использовании журнала файловая система осматривает только те участки диска, в которых изменялись метаданные, то есть она уже «знает», где произошел сбой. Процесс восстановления после сбоя происходит быстрее, чем при традиционной проверке с помощью `fsck`. И что самое существенное — время восстановления совсем не зависит от размера раздела, а скорее зависит от интенсивности операций на момент сбоя.



#### ПРИМЕЧАНИЕ

Существует два варианта работы журналируемых файловых систем: журналирование метаданных, что гарантирует целостность структуры, и журналирование самой информации, что позволяет восстановить ее после сбоя. Файловая система `ext3` поддерживает оба варианта.

Платить за удобство теперь приходится быстродействием, так как самая медленная операция в компьютере — это операция ввода/вывода на диск, и количество таких операций возросло, особенно при использовании варианта с журналированием данных. Решают вопрос разными ухищрениями, например размещают журнал на другом физическом диске. Но фактически время работы с журналом намного меньше, чем работа непосредственно с данными. Что интересно, за счет улучшения алгоритма движения головки жесткого диска в некоторых тестах файловая система `ext3` показывает даже лучшую производительность, чем `ext2`.

Некоторый полезный объем теперь приходится отводить под сам журнал, но он, как правило, не занимает много места.

#### ВНИМАНИЕ



Журналируемые файловые системы предназначены не для восстановления всех данных любой ценой, а для поддержания непротиворечивости метаданных файловой системы на момент сбоя.

Файловая система `ext3` поддерживает следующие три режима работы.

- `data=writeback` — режим, при котором выполняется только журналирование метаданных, сами данные не журналируются. Сохранность данных после сбоя не гарантируется, но скорость работы выше по сравнению с другими режимами. Журнал позволяет уменьшить время начальной загрузки системы.
- `data=ordered` — используется по умолчанию, похож на предыдущий, то есть журналируются только метаданные, но блоки данных записываются первыми. В большинстве случаев такой режим гарантирует сохранность данных, особенно если данные дописывались в конец файла, чего в принципе достаточно.

- `data=journal` — режим полного журналирования, все новые данные вначале записываются в журнал и только после этого переносятся на свое постоянное место. В случае аварийного отказа журнал можно повторно перечитать и привести данные и метаданные в непротиворечивое состояние.

Выбранный отличный от `ordered`, который, как уже говорилось, используется по умолчанию, указывается помощью параметра `-o`. Например:

```
$ sudo mount -o data=journal -t ext3 /dev/hda5 /home
```



#### ПРИМЕЧАНИЕ

Конвертировать `ext2` в `ext3` просто, достаточно использовать утилиту `tune2fs` с параметром `-j`. Например, `tune2fs -j /dev/hda5`.

Именно `ext3` разработчики рекомендуют как подходящую для большинства случаев, хотя по умолчанию предлагается `ext4`.

## Файловая система `ext4`

Файловая система `ext3` заслужила репутацию надежной, ей доверяют свои данные многие пользователи и корпорации. Но наиболее существенным ограничением является максимальный размер хранилища 16 Тбайт. Чтобы снять это ограничение, в середине 2006 г. была выпущена серия заплаток (patch), которые, в свою очередь, нарушали обратную совместимость. Поэтому было принято решение на основе заплаток создать новую файловую систему, которая получила название `ext4`.

Главная задача разработчиков — не просто повысить производительность и улучшить характеристики, но и предоставить простой путь перехода пользователей, работавших с `ext2` или `ext3`, на новую версию файловой системы.

Размер раздела увеличен до 1 эксабайта, а максимальный размер файла — 16 Тбайт (в будущем планируется увеличить это число), в каталоге может быть до 4 млн файлов (в `ext3` — 32 тыс.). В `ext4` предложен механизм пространственной (англ. extent) записи файлов, когда новые данные добавляются в конец заранее выделенной по соседству области файла, что уменьшает фрагментацию и повышает производительность. Кроме этого механизм отложенного размещения максимально затягивает запрос на запись файла, и в итоге при сбрасывании данных на диск драйвер файловой системы, зная размер файла, старается отвести под него рядом расположенные блоки. Это увеличивает производительность и уменьшает фрагментацию.

И главное, чтобы перейти с `ext3` на новую файловую систему, достаточно просто смонтировать раздел с `ext3` как `ext4`. Старые данные будут обрабатываться, как принято в `ext3`, новые получают все преимущества `ext4`.

## Файловая система ReiserFS

Проект начат в конце 1990-х гг., первый прототип носил название TreeFS. Основным разработчиком являлся Ханс Райзер и основанная им компания Namesys. Задача стояла весьма глобальная — создать единую общедоступную среду наименования (namespace). По многим параметрам ReiserFS на голову превосходила ext3, являющуюся базовой для Linux на то время. Так, размер файлов может достигать 8 Тбайт, максимальный размер раздела — 16 Тбайт. Использование оптимизированных b\* сбалансированных деревьев, кроме увеличения производительности, фактически снимает ограничение на количество каталогов. Поддерживается журналирование, но только метаанных. Однако главное преимущество ReiserFS проявляется в работе с маленькими файлами. Несколько маленьких файлов можно упаковать в один блок (англ. tail packing) или сохранить в inode. По необходимости для файла может ассигноваться точный размер. В итоге упрощается поиск данных, уменьшается фрагментация и экономится место на диске, занятом большим количеством файлов небольшого размера.



### ПРИМЕЧАНИЕ

Как известно, большинство файловых систем оперируют только целыми блоками данных, часто последний блок не заполнен. Таким образом теряется некоторое дисковое пространство. В ReiserFS эта проблема решена.

Режим упаковки хвостов предусмотрен в ReiserFS по умолчанию, для повышения быстродействия его можно отключить. Хотя показатели ReiserFS при работе с большими файлами довольно высоки, именно работа с маленькими файлами (меньше 1 Кбайт) и обслуживание большого их количества выделяет эту файловую систему среди прочих. Устойчивость к сбоям ReiserFS проверена временем.

На смену ReiserFS версии 3 планировался следующий релиз Reiser4, но, к сожалению, после того как Ханс Райзер сел в тюрьму по обвинению в убийстве жены (история вызвала большой ажиотаж в обществе из-за того, что у обвинения не было доказательств), разработки файловой системы замедлились. Файловая система ReiserFS поддерживается всеми современными дистрибутивами.

## Файловая система XFS

Основа файловой системы XFS была создана в начале 1990-х гг. фирмой Silicon Graphics Inc. (сокр. SGI) для выпускаемых мультимедийных компьютеров с операционной системой Irix с целью замены уже не удовлетворявшей требованиям времени EFS. Файловая система была ориентирована на работу с очень большими файлами по 9 эксабайт и файловыми системами такого же размера. Особенностью данной файловой системы является устройство журнала: в журнал пишется часть метаанных самой файловой системы, поэтому весь процесс восстановления сводится к копированию этих данных из журнала в файловую систему. Тесты на производительность показывают бесспорное преимущество XFS, особенно при работе

с большими и в большинстве случаев средними файлами. Кроме того, эту файловую систему характеризует прямолинейность падения производительности при увеличении нагрузки и предсказуемость, дополнительно она не генерирует излишнюю дисковую активность, поскольку пытается кэшировать как можно больше данных, и «основанием» для сброса на диск является заполнение памяти, а не интервал времени, как это принято в других файловых системах. В XFS реализована очень хорошая возможность, позволяющая избежать фрагментации файлов, называемая *delayed allocation*. При этом файловая система, получая данные для записи, по началу лишь резервирует под них необходимое свободное место, откладывая саму запись до момента фактического сброса данных. Когда же такой момент наступает, XFS решает, куда необходимо их поместить. Если осуществляется дозапись, то подбираются соседние сектора. Но наибольший эффект в такой задержке получается еще за счет того, что если создается временный файл с малым временем жизни, то он вообще при таком случае на диск не записывается.



#### ПРИМЕЧАНИЕ

В Linux доступна еще одна файловая система JFS (Journaled File System), изначально разработана фирмой IBM, по своим характеристикам она схожа с XFS, но при установке редко когда предлагается. Поэтому подробно останавливаться на ней не будем.

## Версии Linux Mint

Так же, как и Ubuntu Linux, дистрибутив Linux Mint 9 Isadora является Live-дистрибутивом, то есть его можно запустить без предварительной установки на жесткий диск. На момент написания этих строк на официальном сайте проекта были доступны четыре варианта дистрибутива с Рабочим столом GNOME (GNU Object Model Environment, <http://www.gnome.org/>):

- Live CD Standart Edition (ранее Main Edition) — поставляется в виде образа LiveCD размером 700 Мбайт, в котором доступны все необходимые кодеки и несвободные части, но интерфейс Рабочего стола в Live-режиме изначально только английский (любой другой язык будет автоматически установлен при инсталляции дистрибутива на жесткий диск);
- Live DVD — образ чуть большего размера (757 Мбайт), в который включены дополнительные пакеты (Java, VLC, F-Spot, OpenOffice.org-base, Samba, ttf-dejaVu), интерфейс в Live-режиме также только английский (рис. 2.3);
- OEM installation disks — специальный установочный диск для фирм, занимающихся продажей компьютеров, позволяет установить систему без создания учетной записи пользователя;
- USA/Japan distribution disks — специальный образ диска для распространения на территории США и Японии и в других странах, имеющих ограничения на распространение некоторого программного обеспечения.

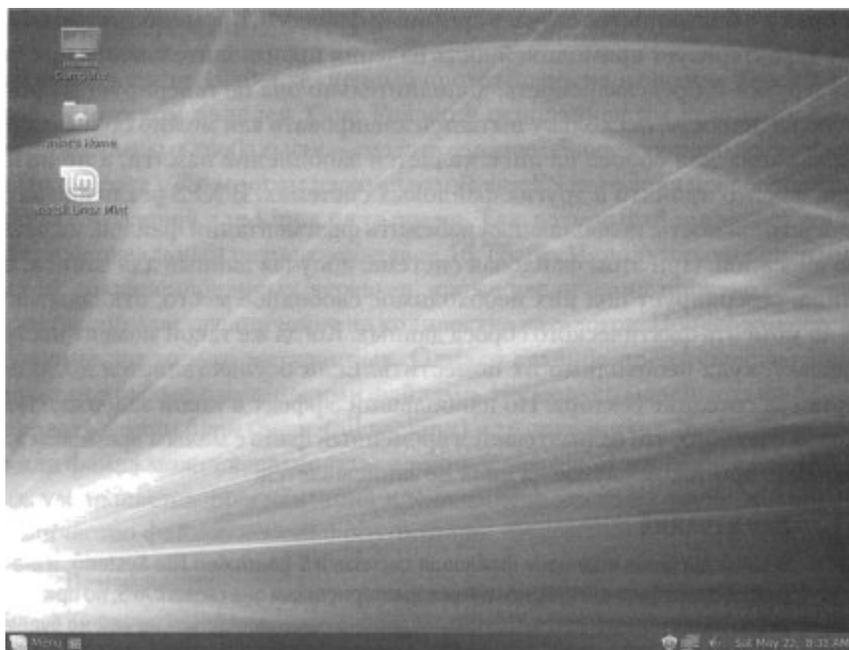


Рис. 2.3. Рабочий стол GNOME в Linux Mint

Пока в списках для загрузки не значится LiveDVD Universal Edition, образ чуть большего размера (1,3 Гбайт), в системе присутствуют только свободные компоненты, но есть возможность выбора языка, включая русский и украинский. Он обычно выходит чуть позже официального релиза дистрибутива, и, возможно, на момент появления книги в печати Universal Edition уже будет находиться на FTP серверах. Кодеки в Universal Edition убраны, чтобы у пользователей, проживающих в странах со строгой лицензионной политикой, не было проблем. Затем установить нужный кодек очень просто.

Обе версии собраны в двух версиях с оптимизацией под 32- и 64-битные процессоры.



#### ПРИМЕЧАНИЕ

Оптимизация под определенный процессор указывает минимальную конфигурацию системы, при которой можно запустить дистрибутив. В компьютерах с более поздними процессорами такие дистрибутивы работают без проблем. Тесты показывают, что в оптимизированной 64-битной системе производительность выше. Кроме того, снимаются ограничения, присущие 32-битным системам. Главное из них — возможность использования не более 4 Гбайт оперативной памяти.

Через некоторое время после релиза выходит 64-битная версия системы (X86 64), рекомендуемая пользователям современных компьютеров. Сообщество пользователей Linux Mint предлагает и версии системы с Рабочими столами KDE (рис. 2.4),

Xfce (рис. 2.5), Fluxbox и LXDE (Openbox). Хотя они обычно выходят с некоторым запаздыванием.



Рис. 2.4. Рабочий стол KDE 4 в Linux Mint



Рис. 2.5. Рабочий стол Xfce

**ПРИМЕЧАНИЕ**

Поддержка Linux Mint 9 заявлена до апреля 2013 г., то есть до этого времени будут доступны обновления для всех компонентов системы, а также к этому времени обычно выходит следующий релиз с долгим периодом поддержки.

Все СЕс синхронизированы и совместимы с основным релизом и используются теми, кто по разным причинам хочет работать с другим оконным менеджером (системные требования или предпочтения). Так, версия с LXDE основана на Linux Mint Main Edition и имеет в своем составе легкий оконный менеджер Openbox и файловый менеджер PCManFM. Версия с Xfce основана на Xubuntu, в котором также используется Рабочий стол Xfce. В минимальных системных требованиях для Linux Mint с Рабочими столами Xfce и LXDE указан компьютер с 256 Мбайт оперативной памяти и графическая карта с поддержкой разрешения 800 x 600 и выше, 3 Гбайт свободного места на диске. Обе среды показывают практически идентичные результаты по производительности и потреблению системных ресурсов.

Чтобы не путать читателя, за основу возьмем версию Standart Edition, которую и буду описывать по ходу книги. Об особенностях других версий будет сказано отдельно. Хотя различий, по сравнению с Main Edition, не так уже много. При необходимости в другом Рабочем столе его легко установить самостоятельно, что мы и разберем по ходу изложения материала.

Программа установки в различных версиях дистрибутива схожа.

## Программа установки Linux Mint

Дистрибутив поставляется в виде загрузочного ISO-образа, который необходимо записать на CD (Standart Edition) или DVD, установить в BIOS вариант загрузки с привода компакт-дисков и перезапустить компьютер. При загрузке появится меню диска (рис. 2.6), в котором необходимо выбрать свой вариант.

Чтобы загрузиться в Live-режиме, выбираем вариант, предлагаемый по умолчанию — Start Linux Mint.

**ПРИМЕЧАНИЕ**

В Universal Edition для загрузки системы с русифицированным меню и окружением следует выбрать Start Linux Mint in Russian [ru].

Если при загрузке происходят сбои, выводятся сообщения об ошибках или вместо изображения получаем черный экран, то следует выбрать загрузку в режиме совместимости Start Linux Mint (compatibility mode), в которой система будет использовать простые видеодрайвера VESA (сокр. Video Electronics Standards Association) и отключит поддержку APIC (сокр. Advanced Programmable Interrupt Controller — «улучшенный программируемый контроллер прерываний»).

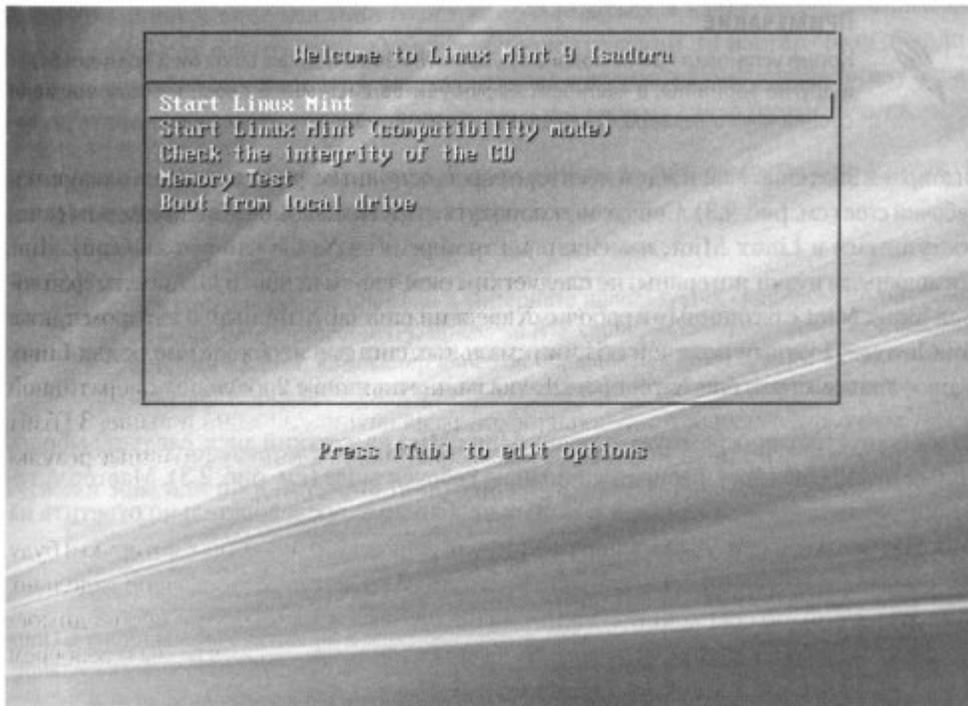


Рис. 2.6. Загрузочное меню Linux Mint

Загрузка в Live-режиме позволяет протестировать оборудование и познакомиться с системой перед установкой.



**ПРИМЕЧАНИЕ**

В версии 8 был дополнительный пункт меню Start Linux Mint (OEM installation mode), выбор которого сразу же запускал программу установки, без запуска живой системы.

Случается, что скачанный ISO-образ получился с ошибками или такие ошибки появились при записи на носитель, во время установки это приведет к проблемам. Чтобы сразу определить целостность, перед установкой следует выбрать пункт Check the integrity of the CD. Если по результатам проверки получаем сообщение Check finished: no errors found, можно начинать установку с выбранного диска.

Иногда сбой в компьютере может появляться из-за неправильной работы оперативной памяти. Проверить установленную в системе оперативную память можно, выбрав пункт Memory Test, запущенная утилита memtest проведет тесты, выполнение которых потребует некоторого времени.

И наконец, последний пункт меню Boot from local drive позволяет загрузиться с жесткого диска.

**ПРИМЕЧАНИЕ**

Кроме установки при помощи загрузочного компакт-диска Linux Mint поддерживает и другие варианты, в частности загрузку по сети. Но такой способ требует наличия специального сервера. В домашней обстановке такой вариант используется редко.

Выбираем Start Linux Mint и ждем некоторое время, когда загрузится система и появится Рабочий стол (см. рис. 2.3). Сейчас можно запустить и познакомиться с приложениями, доступными в Linux Mint, изменить настройки. Учтывая, что все данные будут после перезагрузки потеряны, не следует бояться что-то испортить. Кстати, Рабочий стол Linux Mint с различными рабочими средами стандартизирован и выполнен «под Windows». Поэтому новичок обычно сразу находит все необходимое для работы, первое знакомство с системой происходит на интуитивном уровне.

Чтобы запустить программу установки Linux Mint на жесткий диск, следует выбрать ярлык Install Linux Mint, расположенный на Рабочем столе (см. рис. 2.3). Мастер установки состоит из семи шагов, в которых необходимо последовательно ответить на простые вопросы или указать параметры.

**ПРИМЕЧАНИЕ**

Мастер установки Linux Mint полностью схож с аналогичным мастером в Linux Ubuntu/Kubuntu, поэтому, установив Linux Mint, вы без труда разберетесь с установкой и работой не только в этих дистрибутивах, но и в производных от них.

На первом шаге мастер позволяет выбрать язык, на котором будет производиться установка дистрибутива (рис. 2.7).

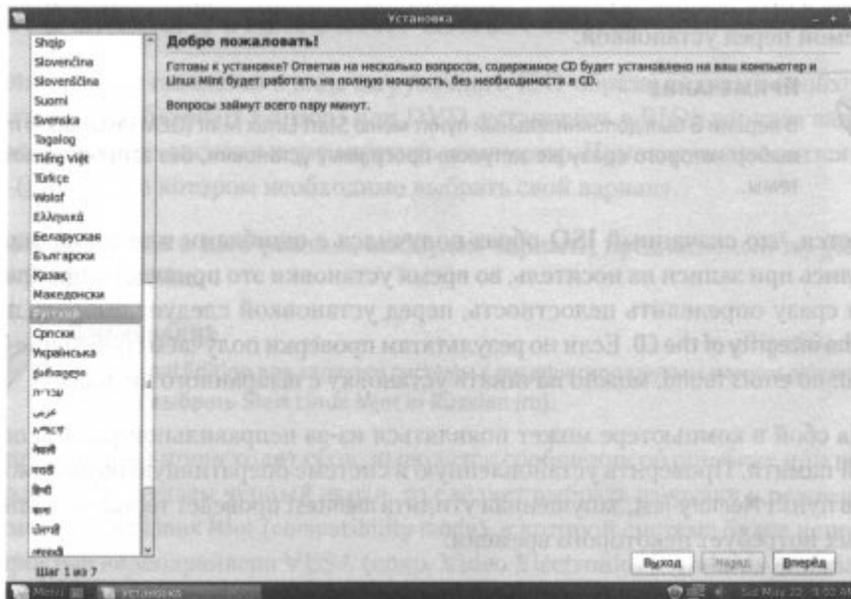


Рис. 2.7. Шаг первый: выбор языка установки и системы

Выбранный язык впоследствии будет использоваться в качестве системного по умолчанию. Если в загрузочном меню выбирался русский, то мастер сразу предложит использовать его дальше. Ссылка Сведения о релизе, размещенная внизу страницы, позволяет просмотреть информацию о системе на сайте разработчика.

На втором шаге настраиваем часовой пояс. Вначале мастер пытается получить точное время с сервера времени.



#### ПРИМЕЧАНИЕ

Для синхронизации времени в Интернете используется специальный протокол Network Time Protocol (NTP). Время на NTP-серверах синхронизируется с высокоточными часами, компьютер клиента, обращаясь к NTP-серверу, автоматически получает точное время и при необходимости подстраивает системные часы.

Если нужно, корректируем часовой пояс при помощи раскрывающихся списков Регион и Зона или просто указав точку нахождения на карте (рис. 2.8).

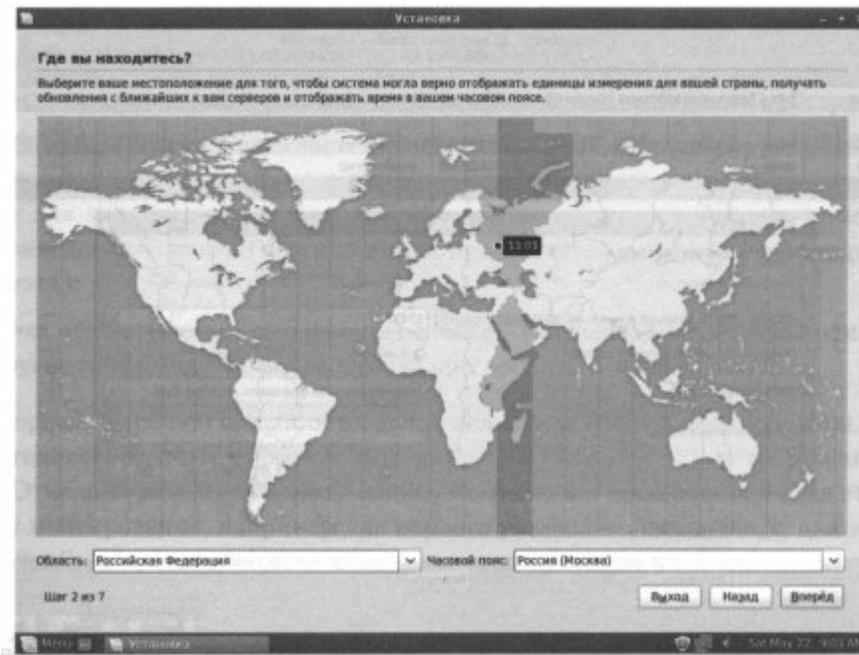


Рис. 2.8. Шаг второй: выбираем часовой пояс

Третий шаг — выбор клавиатурной раскладки. В зависимости от выбора пользователя на предыдущих двух шага установки мастер предложит свой вариант раскладки, который и является оптимальным в большинстве случаев. Проверить ее можно в поле внизу, щелкнув в нем мышкой и введя символы с клавиатуры. Кроме этого для русской раскладки предлагается еще два десятка вариантов, в обычных условиях их использовать нет смысла.

**ПРИМЕЧАНИЕ**

В Linux раскладка клавиатуры соответствует файлу в текстовом формате, при желании можно легко создать свой вариант раскладки.

Но при необходимости устанавливаем переключатель в положение **Выбрать свою** и отмечаем нужную раскладку.

Четвертый шаг — **Подготовка жесткого диска** — самый важный, здесь необходимо указать, на какие разделы будет установлен Linux Mint. По умолчанию выбор установлен в **Стереть и использовать весь диск**. Если Linux Mint будет единственной системой, можно выбрать этот вариант, мастер в этом случае подготовит разделы для установки самостоятельно. Иначе выбираем **Указать разделы вручную (расширенно)** и переходим к следующему окну **Подготовка разделов диска** (рис. 2.9).

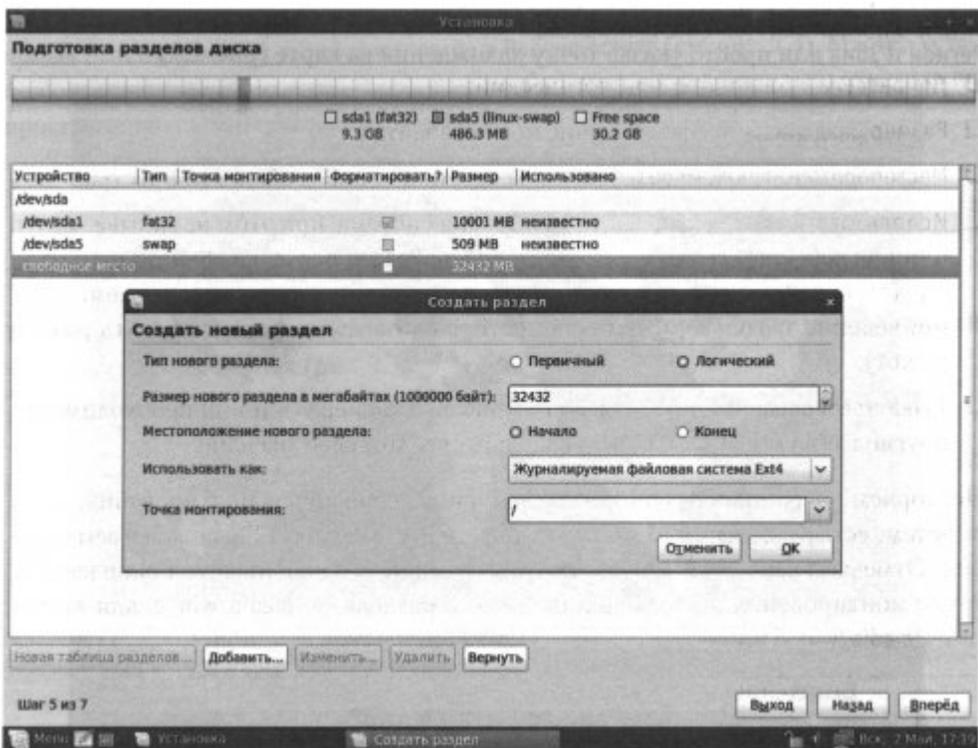


Рис. 2.9. Подготовка разделов жесткого диска

**ВНИМАНИЕ**

Программа подготовки разделов позволяет лишь добавлять, удалять разделы, а также форматировать их в определенную файловую систему. Изменять размер раздела она не может (для этого его вначале придется удалить, а затем создать новый с требуемым размером).

Для установки Linux Mint заранее следует подготовить раздел требуемого размера, например освободив один из разделов в Windows. Предположим, у нас есть раздел NTFS, освобожденный для установки Linux.



#### ПРИМЕЧАНИЕ

Хотя программа установки не имеет возможности изменять разделы жесткого диска, в репозитории находится программа GParted, которая умеет изменять размеры разделов. Установив GParted в «живой» системе, мы получим необходимый инструмент. Для установки следует открыть консоль и ввести `sudo apt-get update` и затем `sudo apt-get install gparted` или использовать для установки менеджер пакетов Synaptic.

Щелкаем на нем кнопкой мыши и нажимаем кнопку Удалить, в результате получаем свободный раздел. К примеру, нам теперь необходимо создать три раздела — подкачки, корневой и домашний. Выбираем свободное место, нажимаем кнопку Добавить и в появившемся окне выбираем:

- **Тип раздела** — основной или логический;
- **Размер раздела** — указывается число в мегабайтах;
- **Местоположение нового раздела** — с начала или конца свободного пространства;
- **Использовать как** — указывается файловая система, при этом не забываем, что один из разделов должен указываться как **Раздел подкачки**, файловую систему для корневого или домашнего разделов выбираем исходя из описания, приведенного ранее (по умолчанию предлагается отформатировать раздел в ext4);
- **Точка монтирования** — выбираем из списка / (корневая), и при необходимости другие точки монтирования, как вариант вводим свое значение.

Повторяем операцию создания разделов для каждой точки монтирования. Если в системе есть разделы, отформатированные в FAT32 или NTFS, не забываем и про них. Отмечаем раздел, нажимаем кнопку **Изменить** и в появившемся окне вводим точку монтирования, например для первого раздела — `/media/win_c`, для второго — `/media/win_d` или `/windows`, если раздел один. Как вам удобнее.

#### ВНИМАНИЕ



Все производимые вами изменения пока остаются виртуальными, то есть не затрагивают жесткий диск. И пока вы не перешли к следующему шагу, опасаться за свои данные не нужно. Можно смело экспериментировать с разделами.

По окончании всех настроек нажимаем кнопку **Вперед** и переходим к пятому шагу, на котором настраиваем параметры учетной записи, которую вы будете использовать для входа в систему (рис. 2.10).

Вводим имя (оно будет выводиться в окне регистрации), логин, дважды пароль, в поле **Имя этого компьютера?** указывается сетевое имя системы.

**ВНИМАНИЕ**

Для логина следует использовать только латинские буквы, без спецсимволов и знаков пробела.

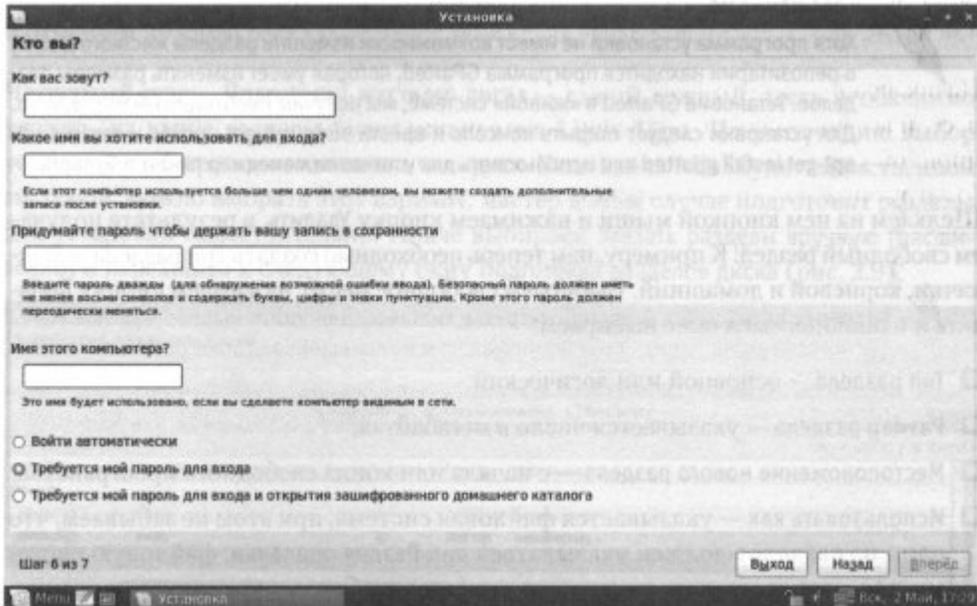


Рис. 2.Ю. Шаг шестой: настройка учетной записи

После ввода пароля в поле справа будет выведена подсказка, говорящая о его надежности. Пароль должен состоять не менее чем из 6 символов (лучше 8) и содержать не только буквы (заглавные и прописные), но и цифры и спецсимволы. Рекомендуется, чтобы надежность, показанная программой установки, была не менее чем **Достаточная**. Кроме этого переключатель в самом низу позволяет выбрать один из следующих трех вариантов входа пользователя в систему.

- **Войти автоматически** — после загрузки системы пользователь сразу же попадает в рабочее окружение без ввода учетных данных. Удобен с точки зрения простоты использования, но менее безопасен, так как любой может получить доступ к личной информации.
- **Требуется мой пароль для входа** — после загрузки системы следует ввести свой логин и пароль. Вариант, предлагаемый по умолчанию, его и нужно использовать в большинстве ситуаций. В таком случае в системе легко создать несколько учетных записей с разными правами, которые будут работать, не получая доступа к персональным данным друг друга.
- **Требуется мой пароль для входа и открытия зашифрованного домашнего каталога** — похож на предыдущий, только данные в домашнем каталоге пользователя сохра-

няются в зашифрованном виде, поэтому любой пользователь, получивший доступ к диску, не сможет их прочесть. Полезен при установке Linux Mint на ноутбук.

Затем первые два пункта можно будет переопределить в рабочей системе, выбрав в меню Администрирование • Экран входа в систему.

На последнем, шестом, шаге (рис. 2.11) предлагается еще раз проверить настройки и, если все нормально, собственно начать процесс установки. Для загрузки операционной системы Linux следует установить специальную программу-загрузчик (о загрузчике читайте чуть ниже).

#### ПРИМЕЧАНИЕ

Установить другую файловую систему, предлагаемую по умолчанию, можно также, передав параметр ядру при загрузке системы. Для этого, выбрав нужный пункт, нажимаем клавишу «е» и дописываем в появившейся строке `partman/default_filesystem=ext3` (для `ext3`).

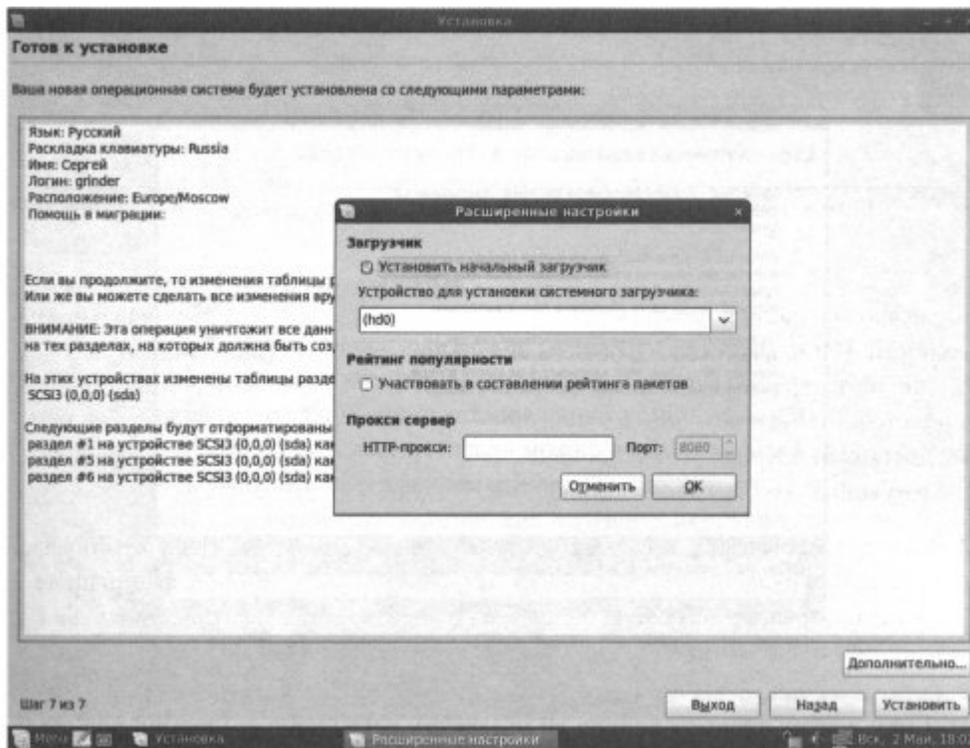


Рис. 2.11. Дополнительные настройки загрузчика

По умолчанию загрузчик устанавливается в нулевой сектор первого жесткого диска, который отмечен как `hd0`. В большинстве случаев это единственно правильный

вариант, иначе Linux просто нельзя будет загрузить. Обратите внимание на кнопку **Дополнительно:** нажав ее, мы получаем возможность изменить раздел, в который будет установлен загрузчик, указать параметры прокси-сервера, если подключение к Интернету производится через такой сервер. Установка флажка **Участвовать в составлении рейтинга пакетов** даст возможность затем отсылать на сервер разработчиков данные об установленных в системе пакетах, что позволит составлять рейтинг.

Все настройки закончены, нажимаем кнопку **Установить** и ждем, пока завершится процесс установки системы на жесткий диск. Мастер будет информировать о ходе установки и выводить информацию о некоторых особенностях самого Linux Mint (рис. 2.12). Чтобы упростить дальнейшие настройки пользователю, мастер автоматически загружает из Интернета языковые модули. Если канал связи с Интернетом не достаточно быстр, это может занять некоторое время. В таком случае можно пропустить данный этап, нажав кнопку **Пропустить**, а сами модули загрузить вручную при помощи пакетного менеджера.

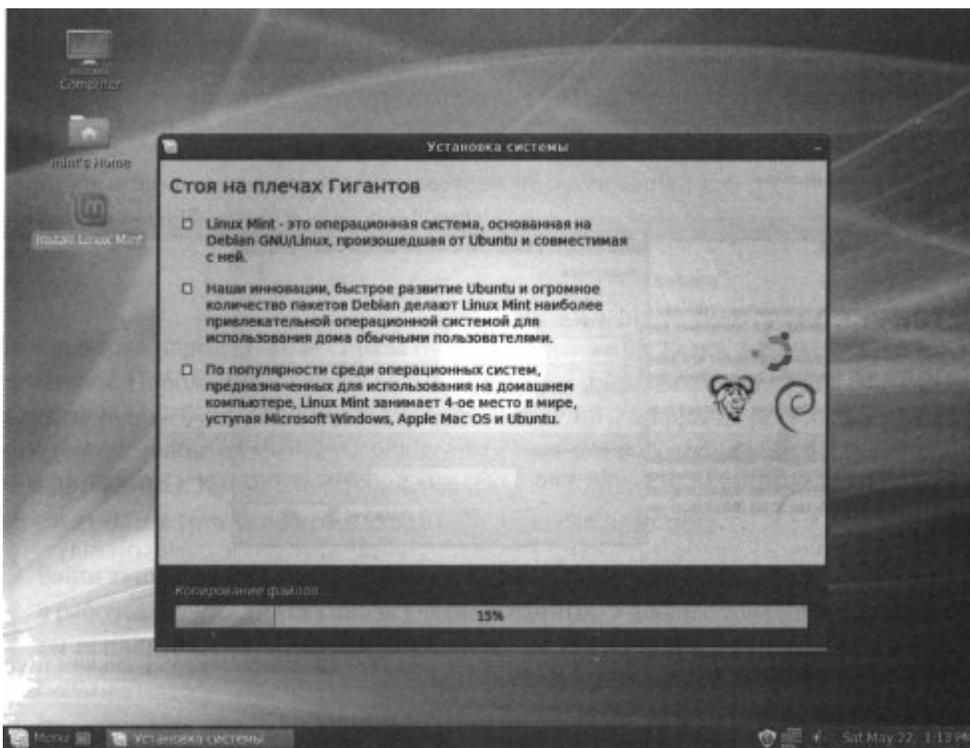


Рис. 2.12. Идет установка Linux Mint

По окончании установки мастер предложит продолжить работу или перезагрузить систему. Перезагружаемся, регистрируемся, используя учетные записи, созданные на шестом шаге установщика.

## Альтернативные варианты установки

Сегодня многие компьютеры подключены к локальным сетям и Интернету, что упрощает обмен файлами между пользователями. К тому же объем флеш-накопителей давно уже превысил возможности DVD. Все это приводит к тому, что часто при покупке нового компьютера пользователи отказываются от приобретения привода компакт-дисков. В нем просто нет необходимости. Как же установить Linux в таком случае? Разработчики Linux Mint, а также Ubuntu Linux и других дистрибутивов предлагают несколько вариантов — сетевая загрузка, создание загрузочного флеш-накопителя. Кроме этого возможен вариант загрузки скачанного ISO-образа прямо с жесткого диска. В новой версии 9 появилась еще возможность установки из Windows, для этого достаточно записать диск, вставить его в привод и запустить находящийся внутри файл `mint4win.exe`, далее указываем раздел диска, на который будет установлен Linux Mint, учетную запись, создаваемую в процессе установки, и язык системы.

Как я уже говорил ранее, сетевая загрузка требует наличия отдельного сервера, поэтому рассматривать ее мы не будем.



### ПРИМЕЧАНИЕ

В Ubuntu Linux разработчики также предлагают вариант установки прямо из Windows, для этого задействуется специальное приложение Wubi (<http://wubi-installer.org/>), которое размещено на установочном диске.

## Установка с флеш-носителя

Создать загрузочный флеш-носитель можно разными способами. В Интернете доступно с десяток программ, в которых реализована необходимая функциональность. При наличии в вашем распоряжении установленного Linux Mint или Linux Ubuntu для создания загрузочного флеш-носителя можно воспользоваться утилитой `usb-creator`, или с графическим интерфейсом `usb-creator-kde`, или `usb-creator-gtk`. Установить ее можно из репозитория пакетов, для чего достаточно ввести команду

```
$ sudo apt-get install usb-creator
```

После этого запускаем из меню программу Создание загрузочного USB-диска (Live USB creator) (рис. 2.13).

Далее при помощи кнопки Обзор указываем на привод компакт-дисков или ISO-образ, с которого будет создаваться загрузочная флешка. Вставляем сам флеш-накопитель в USB-разъем и, после того как он определится системой и появится в окне программы, нажимаем кнопку Создать загрузочный диск. По окончании устанавливаем в приоритете загрузки в BIOS загрузку с внешнего носителя и перезагружаем компьютер. В дальнейшем все действия аналогичны установке с компакт-диска.

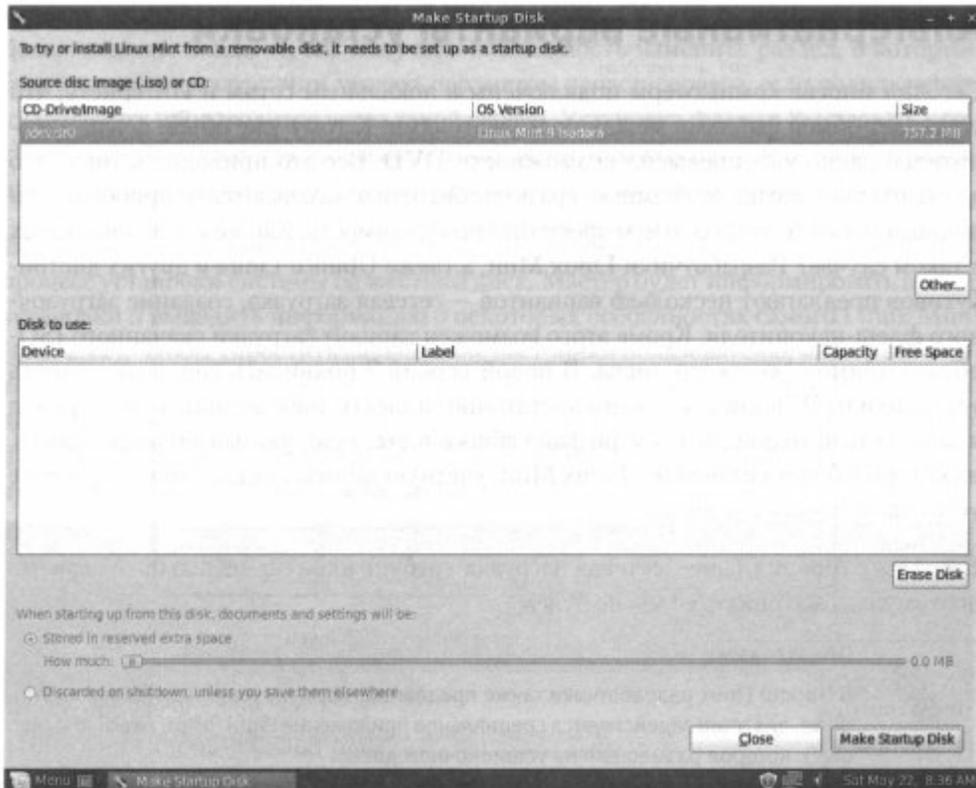


Рис. 2.13. Интерфейс программы Live USB-Creator

Но такой вариант требует наличия установленного Linux. Если в вашем распоряжении есть только Windows, следует выбрать программу, написанную под эту операционную систему.

Одна из таких программ Universal USB Installer (рис. 2.14), которая разрабатывается в рамках проекта Pendrivelinux. Скачать ее можно со страницы <http://www.pendrivelinux.com/downloads/Universal-USB-Installer>. Программа бесплатная и благодаря наличию простого интерфейса очень проста в использовании. Запускаем скачанный файл **Universal-USB-Installer-v1.5.1.exe** и в появившемся окне выбираем **дистрибутив Linux**, для которого будет создаваться загрузочный флеш-носитель.

В списке нет Linux Mint, но учитывая его происхождение от Ubuntu Linux выбираем эту систему. В моем случае это **Ubuntu 10.04 Desktop i386**. Далее при помощи кнопки **Browse** указываем на образ, полученный с сайта Linux Mint. Вставляем в разъем USB флеш-накопитель, ждем когда он определится системой, выбираем его в окне и нажимаем **Create**.

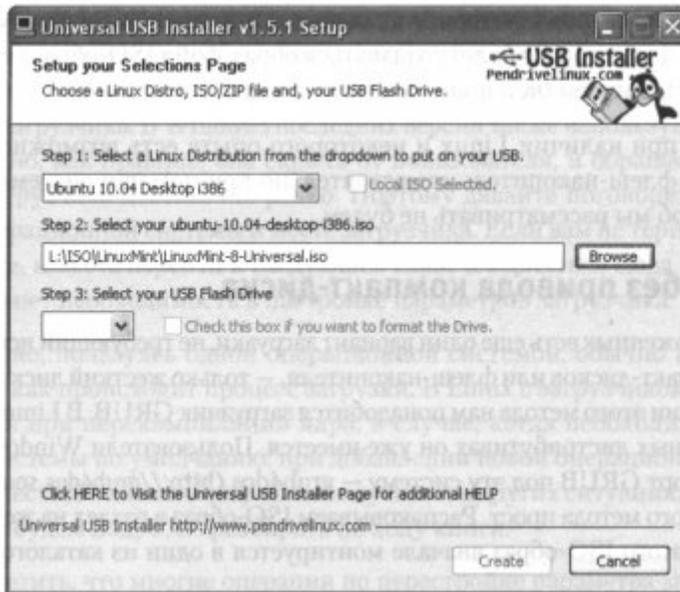


Рис. 2.14. Окно программы Universal USB Installer

Еще одна популярная программа UNetbootin (рис. 2.15), позволяющая установить на компьютер или создать загрузочный флеш-диск с популярными Linux/BSD дистрибутивами, работает в Windows 2000/XP/Vista/7 или Linux. Официальный сайт проекта — <http://unetbootin.sourceforge.net/>.

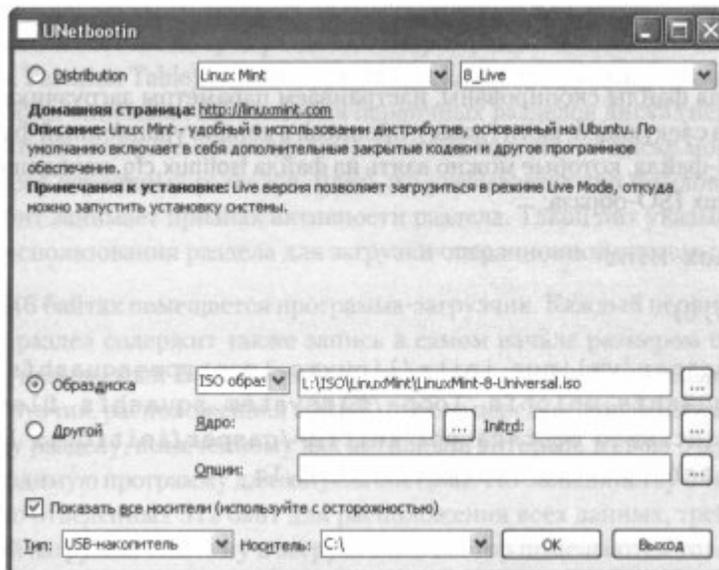


Рис. 2.15. Программа UNetbootin

Принцип использования UNetbootin аналогичен Universal USB Installer: указываем дистрибутив, для которого будет создаваться образ, файл ISO-образа и собственно накопитель. Нажимаем ОК и получаем загрузочный флеш-диск.

Кроме этого при наличии Linux и некоторого опыта есть возможность создать загрузочный флеш-накопитель самостоятельно при помощи системных команд. Но этот способ мы рассматривать не будем.

## Загрузка без привода компакт-диска

Кроме предложенных есть еще один вариант загрузки, не требующий использования привода компакт-дисков или флеш-накопителя, — только жесткий диск компьютера. Для реализации этого метода нам понадобится загрузчик GRUB. В Linux Mint и других современных дистрибутивах он уже имеется. Пользователи Windows могут использовать порт GRUB под эту систему — grub4dos (<http://grub4dos.sourceforge.net>). Принцип такого метода прост. Распаковываем ISO-образ в раздел на жестком диске. В Linux для этого ISO-образ вначале монтируется в один из каталогов, например /mnt:

```
$ sudo mount -o loop /home/grinder/iso/LinuxMint-Universal.iso /mnt
```

В результате в /mnt появится содержимое ISO-образа. Затем монтируем один из разделов жесткого диска и копируем в него файлы из /mnt:

```
$ sudo mount /dev/sdal /linuxmint
```

```
$ sudo cp -vR /mnt/* /linuxmint
```

Теперь, когда файлы скопированы, настраиваем параметры загрузчика (о GRUB я расскажу в следующей части главы), прописав в конфигурационный файл данные ядра и initrd-файла, которые можно взять из файла isolinux.cfg, находящегося в каталоге isolinux ISO-образа:

```
title Linux Mint

root (hd0,0)

kernel /casper/vmlinuz init=/linuxrc looptype=squashfs unionfs
doload=squashfs,unionfs loop=/filesystem.squashfs file=/cdrom/
preseed/mint.seed boot=casper initrd=/casper/initrd.lz locale=ru
quiet splash -initrd /casper/initrd.lz

quiet
```

Теперь перезагружаем систему и в загрузочном меню выбираем пункт Linux Mint. Далее производим установку обычным образом.

# Загрузчик GNU GRUB

Во время загрузки системы первым пользователем встречается меню специальной программы-загрузчика. В Windows последних версий также используется подобная технология, но, как правило, она скрыта от пользователя, и обращается он к настройкам загрузчика достаточно редко. Поэтому давайте поговорим о процессе загрузки операционной системы и месте загрузчика. Если вам не терпится приступить к работе, можете перейти к следующей главе и вернуться сюда в том случае, когда возникнет необходимость в настройке параметров загрузчика.

Действительно, пользуясь одной операционной системой, обычно не задумываешься о том, как происходит процесс загрузки. В Linux с загрузчиком приходится сталкиваться при перекомпиляции ядра; в случае, когда необходимо изменить поведение системы по умолчанию; при добавлении новой операционной системы или нового жесткого диска; в аварийных случаях и в других ситуациях. Некоторые моменты мы будем подробно разбирать по ходу книги.

Следует отметить, что многие операции по перестройке параметра загрузчика автоматизированы. Например, при обновлении ядра из репозитория пакетов в конфигурационный файл загрузчика сразу же заносится соответствующая запись. Поэтому знание того, как работает загрузчик, потребуется при решении нестандартных ситуаций или для тонкой подстройки системы.

Итак, при старте компьютера происходит считывание содержания так называемого MBR (Master Boot Record), который располагается всегда в одном и том же месте — в первом секторе жесткого диска (если совсем точно — в нулевом). Занимает MBR всего 512 байт и состоит из трех разделов. Второй раздел называется таблицей разделов (англ. Partition Table). Занимает он 64 байта, состоит из четырех записей, в каждой из которых описывается геометрия первичных разделов диска (первый и последний цилиндр, занимаемый разделом). Именно поэтому на диске может быть не более четырех первичных разделов. Здесь же указывается тип файловой системы раздела, 1 бит занимает признак активности раздела. Такой бит указывает на возможность использования раздела для загрузки операционной системы.

В первых 446 байтах помещается программа-загрузчик. Каждый первичный и расширенный раздел содержит также запись в самом начале размером один сектор (512 байт), называемый Boot Record. В некоторых операционных системах программа-загрузчик, расположенная в MBR, просто передает управление по цепочке первичному разделу, помеченному как активный, который, в свою очередь, загружает необходимую программу для запуска системы. Но большинству операционных систем мало отведенных 512 байт для расположения всех данных, требуемых для нормальной загрузки. Поэтому в загрузочный сектор помещаются только некоторые инструкции, необходимые для запуска первичного загрузчика операционной системы.

В Linux наиболее популярны менеджеры загрузки LILO (Linux LOader) и GRUB (GRand Unified Bootloader). Поскольку в Linux Mint для загрузки операционной системы используется второй, о нем и пойдет речь далее. Официальный сайт проекта — <http://www.gnu.org/software/grub/>.

Проект возник в 1995 г. при разработке ядра Hurd — довольно оригинальной операционной системы, к сожалению, так и не доведенной до окончательного релиза. Разработчики, вместо того чтобы загружать ядро Hurd как-то по-особенному, предложили спецификацию мультизагрузки (англ. Multiboot Specification), позволяющую загружать операционную систему универсальным образом. Те же операционные системы, которые не были совместимы с этой спецификацией, сохранили схему загрузки по цепочке. В 1999 г. GRUB официально примкнул к проекту GNU, и уже начиная с 2000 г. производители стали включать GRUB в свои дистрибутивы. Сегодня доступна GRUB версии 2, которая и используется в качестве менеджера загрузки в большинстве современных дистрибутивов. Далее речь пойдет именно об этой версии.



#### ПРИМЕЧАНИЕ

Конфигурационный файл GRUB версии 1 назывался `/boot/grub/grub.conf` и имел несколько отличный от первой версии формат.

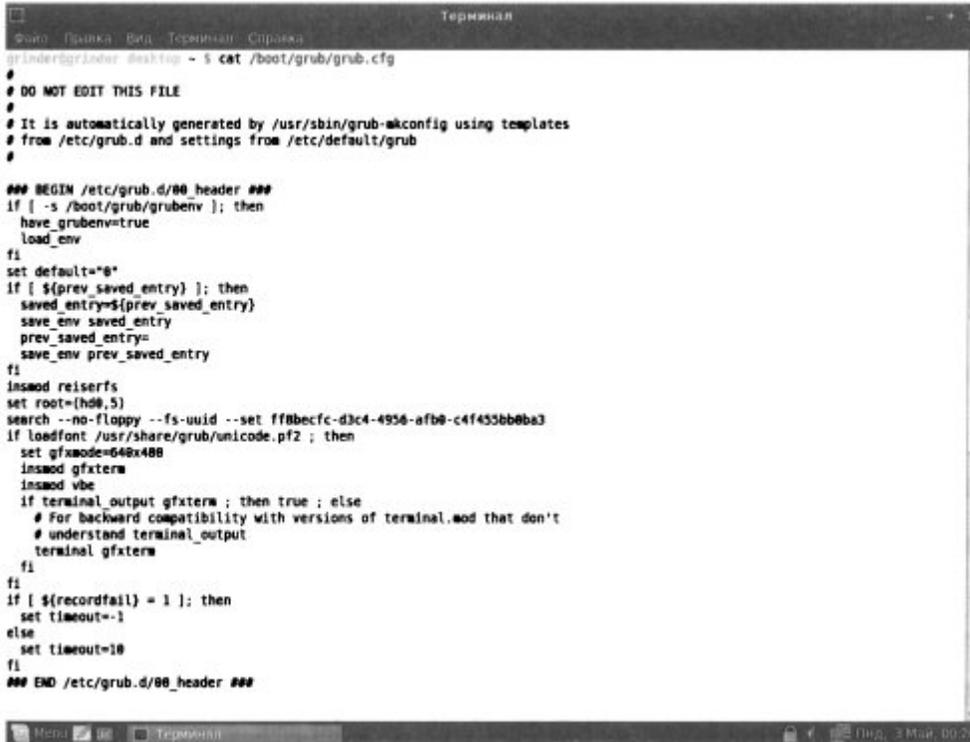
Программа установки Linux Mint сканирует все разделы жесткого диска и заносит в конфигурационный файл GRUB `/boot/grub/grub.cfg`. Просмотреть его содержимое можно, введя команду (рис. 2.16)

```
$ cat /boot/grub/grub.cfg
```

Однако главное различие в версиях GRUB состоит в том, что редактировать конфигурационный файл `/boot/grub/grub.cfg`, как это делали в версии 1 загрузчика, нельзя. Об этом красноречиво свидетельствует предупреждение DO NOT EDIT THIS FILE (от англ. «Не редактируйте файл») в самом начале файла. Файл генерируется автоматически скриптом `/usr/sbin/grub-mkconfig` на основании шаблонов, расположенных в каталоге `/etc/grub.d`, и настроек в `/etc/default/grub`. Поэтому вы можете утратить свои правки при обновлении утилиты GRUB или в других случаях, когда потребуется регенерация `/boot/grub/grub.cfg`. Файл довольно большой по содержанию, поэтому приведу лишь отдельные его части, достаточные, чтобы понимать основные настройки.

```
# DO NOT EDIT THIS FILE
```

```
# It is automatically generated by /usr/sbin/grub-mkconfig using templates
```



```
grub@grub:~$ cat /boot/grub/grub.cfg
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by /usr/sbin/grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s /boot/grub/grubenv ]; then
  have_grubenv=true
  load_env
fi
set default="0"
if [ ${prev_saved_entry} ]; then
  saved_entry=${prev_saved_entry}
  save_env saved_entry
  prev_saved_entry=
  save_env prev_saved_entry
fi
insmod reiserfs
set root=(hd0,5)
search --no-floppy --fs-uuid --set ff0becfc-d3c4-4956-afb8-c4f455bb0ba3
if loadfont /usr/share/grub/unicode.pf2 ; then
  set gfxmode=640x480
  insmod gfxterm
  insmod vbe
  if terminal_output gfxterm ; then true ; else
    # For backward compatibility with versions of terminal.mod that don't
    # understand terminal_output
    terminal gfxterm
  fi
fi
if [ ${recordfail} = 1 ]; then
  set timeout=-1
else
  set timeout=10
fi
### END /etc/grub.d/00_header ###
```

Рис. 2.16. Конфигурационный файл GRUB v.2

```
# from /etc/grub.d and settings from /etc/default/grub
```

```
### BEGIN /etc/grub.d/00_header ###
```

```
if [ -s /boot/grub/grubenv ]; then
```

```
  have_grubenv=true
```

```
  load_env
```

```
  fi
```

```
set default="0"
```

Параметр `set default` показывает, какая система будет загружаться по умолчанию, цифра 0 указывает на первую по списку запись, указанную в параметре `menuentry`. В нашем случае это Linux Mint 9, 2.6.32-21-generic (**dev/sda1**).

```
insmod reiserfs

set root=(hd0,5)
```

Эти две инструкции загружают модуль ядра для файловой системы ReiserFS и указывают на корневой раздел, в котором размещены в том числе конфигурационные файлы загрузчика GRUB.

```
if [ ${recordfail} = 1 ]; then
    set timeout=-1
else
    set timeout=10
```

```
### END /etc/grub.d/00_header ###
```

Параметр `set timeout` устанавливает время ожидания меню загрузчика, в течение которого пользователь может выбрать другую операционную систему.

Далее идут описания тем оформления меню загрузки. В качестве фоновой картинки выбран рисунок в формате PNG/`boot/grub/linuxmint.png`.

```
### BEGIN /etc/grub.d/05_debian_theme ###
```

```
set menu_color_normal=white/black
set menu_color_highlight=black/white
```

```
### END /etc/grub.d/05_debian_theme ###
```

```
### BEGIN /etc/grub.d/06_mint_theme ###
```

```
insmod reiserfs

set root=(hd0,5)

search --no-floppy --fs-uuid --set ff8becfc-d3c4-4956-afb0-
c4f455bb0ba3

insmod png

if background_image /boot/grub/linuxmint.png ; then
    set color_normal=white/black
    set color_highlight=white/light-gray
```

```
else

    set menu_color_normal=white/black

    set menu_color_highlight=white/light-gray

### END /etc/grub.d/06_mint_theme ###

И наконец, меню выбора операционной системы:

### BEGIN /etc/grub.d/10_linux ###

menuentry "Linux Mint 9, 2.6.32-21-generic (/dev/sda1)" {

    recordfail=1

    if [ -n ${have_grubenv} ]; then save_env recordfail; fi

    set quiet=1

    insmod reiserfs

    set root=(hd0,5)

    search --no-floppy --fs-uuid --set ff8becfc-d3c4-4956-afb0-c4f455bb0ba3

    linux /boot/vmlinuz-2.6.32-21-generic root=UUID=5bc61bd9-38b5-43a4-b3e8-11f9fd8ab389 ro quiet splash

    initrd /boot/initrd.img-2.6.32-21-generic

### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_memtest86+ ###

menuentry "Memory test (memtest86+)" {

    linux16 /boot/memtest86+.bin

}

### END /etc/grub.d/20_memtest86+ ###

### BEGIN /etc/grub.d/30_os-prober ###
```

```

menuentry "Microsoft Windows XP Home Edition (on /dev/sdal)" {
    set root=(hd0,1)
    chainloader +1

    ### END /etc/grub.d/30_os-prober ###

```

Все операционные системы, приведенные в параметре `menuentry`, будут доступны для выбора при загрузке. Если пользователь ничего в течение 10 секунд не выберет, то будет запущена операционная система, указанная в `set default`.

Строка

```
linux /boot/vmlinuz-2.6.32-21-generic root=UUID=ff8becfc-d3c4-4956-afb0-c4f455bb0ba3 ro quiet splash
```

указывает на файл ядра `/boot/vmlinuz-2.6.32-21-generic` и параметры, передаваемые ядру при загрузке системы. При помощи `root` указывается корневой раздел. Как видите, для обозначения раздела использован UUID, а не имя.



#### ПРИМЕЧАНИЕ

**Initrd (Initial RAM Disk)** — это образ диска, распаковываемый в оперативную память для начальной инициализации системы, который образует временную файловую систему, содержащую драйвера и утилиты, которые необходимы ядру при старте.

Так, редактировать `/boot/grub/grub.cfg` напрямую не имеет смысла. Посмотрим, откуда берутся установки для его генерирования.

Вначале выводим содержимое файла `/etc/default/grub`, после установки системы при помощи утилиты `cat` или в любом текстовом редакторе:

```

$ cat /etc/default/grub

# If you change this file, run `update-grub` afterwards to update
# /boot/grub/grub.cfg.

GRUB_DEFAULT=0

GRUB_HIDDEN_TIMEOUT=0

GRUB_HIDDEN_TIMEOUT_QUIET=true

GRUB_TIMEOUT="10"

GRUB_DISTRIBUTOR=`lsb release -i -s 2> /dev/null || echo Debian`

```

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command 'vbeinfo'
#GRUB_GFXMODE=640x480
# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true
# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

Теперь разберем некоторые параметры. Вы, наверное, обратили внимание, что файл состоит из ряда строк вида **ПАРАМЕТР=ЗНАЧЕНИЕ**. Все, что записано после знака диэз #, считается комментарием и не учитывается при регенерации нового конфигурационного файла. Закомментировав строки, разработчики уже приготовили настройки, и чтобы привести их в действие, следует просто снять комментарий. Сравнив параметры с файлом /boot/grub/grub.cfg находим совпадения. Например, строка

```
GRUB_DEFAULT=0
```

соответствует `set default="0"`, то есть при необходимости можно изменить операционную систему, которая будет загружаться по умолчанию, прописав здесь нужное значение. Причем если в качестве параметра указать `saved`, то в следующий раз будет загружаться операционная система, выбранная в прошлый раз. Кроме цифры можно сразу задать метку, как она прописана в параметре `menuentry`. Например:

```
GRUB_DEFAULT="Linux Mint 9 Isadora, linux 2.6.32-21-generic (/dev/sda5)"
```

Изменив параметр `GRUB_TIMEOUT="10"`, мы установим новое время ожидания выбора меню пользователем в секундах. Установив значение `-1`, мы отключим таймер отсчета, и меню будет отображаться до тех пор, пока пользователь не выберет нужный пункт.

Строка

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

задает параметры ядру по умолчанию. В данном случае активирована графическая заставка и «молчаливый режим» загрузки. При необходимости дописываем здесь нужные дополнительные параметры. О некоторых из них мы будем говорить по ходу книги. Если загрузка системы невозможна из-за проблем совместимости оборудования, следует попробовать добавить один из следующих параметров — `noapic`, `nolapic`, `noacpi`, `acpi=off`, `pci=noacpi`, `norcpcia` (если есть `pcmcia`-устройство), `irqpoll`, `all generic ide`. Например, для материнских плат на nVidia nForce часто следует добавлять — `noapic nolapic`, иначе загрузка прерывается черным экраном.

Аналогично, передав параметр ядру, можно указать разрешение экрана при загрузке. Например, добавив `vga=7 91`, мы установим разрешение 1024 × 768 с количеством цветов 64 Кбайт, `vga=7 90` — 1024 × 768 и 32 Кбайт цветности, `vga=787` — 800 × 600.

Параметр `ro single` переводит систему в режим загрузки в однопользовательском режиме, который обычно используется при восстановлении работоспособности системы.

В каталоге `/etc/grub.d` есть несколько файлов. Просмотреть их список можно, например, при помощи консольной команды `ls` (рис. 2.17):

```
$ ls /etc/grub.d
00_header          06_mint_theme     20_moment86+      40_custom
05_debian_theme   10_linux          30_os-prober      README
```

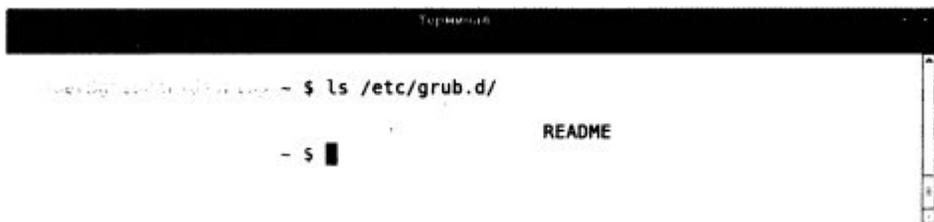


Рис. 2.17. Список файлов в каталоге `/etc/grub.d`

Параметры, которые они генерируют в файле `/boot/grub/grub.cfg`, обозначаются секциями вроде:

```
### BEGIN /etc/grub.d/30_os-prober ###
### END /etc/grub.d/30_os-prober ###
```

Файлы 10\_linux и 30\_os-prober отвечают за поиск и добавление в меню Linux и других операционных систем, установленных на жестких дисках компьютера. Файл 20\_memtest86+ добавляет в меню пункт, позволяющий произвести проверку оперативной памяти. И наконец, файл 40\_custom дает возможность пользователю самостоятельно добавить свои пункты загрузки в меню. Например, это могут быть какие-либо особые варианты загрузки системы.

**ПРИМЕЧАНИЕ**

Файл 40\_custom должен обязательно заканчиваться пустой строкой, иначе последний пункт меню не будет отображаться.

После всех правок /etc/default/grub или файлов в /etc/grub.d для генерирования нового файла /boot/grub/grub.cfg следует выполнить команду

```
$ sudo update-grub
```

**ПРИМЕЧАНИЕ**

При обновлении GRUB с версии 1 на версию 2 необходимо использовать команду `upgrade-from-grub-legacy`, которая на основании старого конфигурационного файла создаст новый.

Забегая чуть вперед, скажу, что в установленном Linux Mint имеется графическая утилита, позволяющая изменить некоторые параметры GRUB, вызвать которую можно из меню Администрирование • Менеджер загрузки (рис. 2.18).

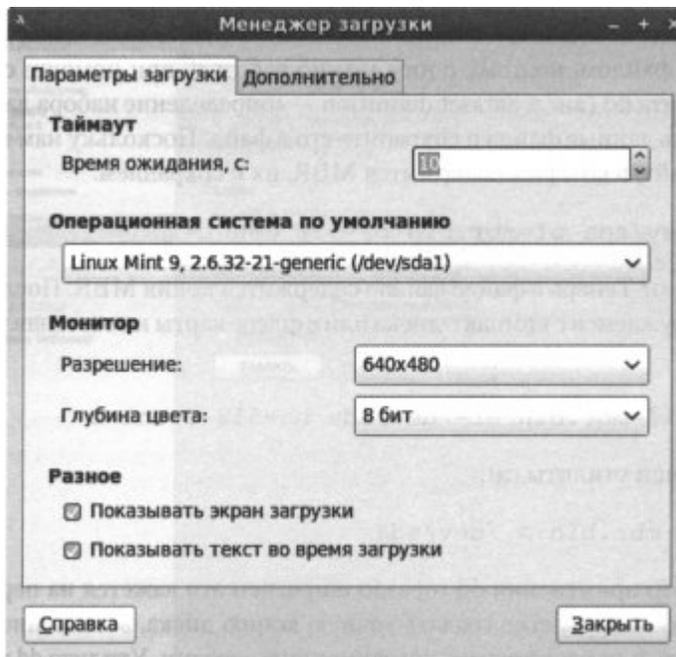


Рис. 2.18. Настройка GRUB при помощи графической утилиты

В появившемся окне можно указать время ожидания выбора системы, операционную систему, загружаемую по умолчанию, разрешение экрана во время загрузки системы. Во вкладке создается спасательная дискета и указывается разрешение экрана при показе меню загрузчика. Все просто и понятно.

## Восстановление загрузчика после установки Windows

Большинство пользователей-новичков устанавливают Linux в качестве второй системы и работают параллельно с Windows. Программа установки Linux Mint обнаруживает другие операционные системы и заносит их в свой список, позволяя выбрать при загрузке. Но в том случае, когда пользователь решает переустановить Windows, он лишается возможности загрузить Linux Mint и все потому, что программа установки Windows стирает MBR, записывая в него свои данные, и, соответственно, удаляет установленный в него менеджер загрузки GRUB. Восстановить возможность загрузки Linux можно двумя способами:

- 1) сохранить MBR с установленным в него GRUB в файл, а после установки Windows восстановить MBR;
- 2) установить GRUB.

Оба варианта требуют возможности загрузиться в Linux хотя бы в «живом» режиме с привода компакт-дисков.

Первый вариант основывается на том, что в Linux (как и в любой UNIX-системе) все является файлом, поэтому с ним можно работать при помощи стандартных утилит. Утилита `dd` (англ. dataset definition — «определение набора данных») позволяет считать данные файла и сохранить его в файл. Поскольку нам нужны всего первые 512 байт, в которых содержится MBR, их и сохраняем:

```
$ dd if=/dev/sda of=mbr.bin bs=512 count=1
```

Правда, удобно? Теперь в файле `mbr.bin` содержится копия MBR. После установки Windows загружаемся с компакт-диска или с флеш-карты и восстанавливаем MBR из файла

```
$ sudo dd if=mbr.bin of=/dev/sda bs=512 count=1
```

или при помощи утилиты `cat`:

```
$ sudo cat mbr.bin > /dev/sda
```

К слову, спектр применения `dd` гораздо шире, чем это кажется на первый взгляд. С ее помощью можно легко создать точную копию диска, вставленного в привод компакт-диска, или клонировать операционную систему. Утилита `dd` портирована и в Windows, скачать ее можно с сайта <http://www.chrysocome.net/dd>.

Теперь рассмотрим второй вариант. Во многих дистрибутивах доступен специальный режим **rescue**, или возможность создания спасательной дискеты, позволяющей при необходимости в любом случае загрузиться в корневую систему и восстановить загрузчик. Затем, обычно автоматически, производится поиск установленной Linux-системы, и, если такая будет найдена, загружается система, установленная на жестком диске. Далее загрузчик легко восстанавливается, причем разработчики часто предлагают графические утилиты. К сожалению, в загрузочном меню LiveCD диска Ubuntu такого пункта нет, как нет его и в Linux Mint. Поэтому прямого способа восстановить загрузчик GRUB не существует. Но, впрочем, это не страшно. Учитывая, что Linux Mint все-таки LiveCD-дистрибутив, умеющий работать без установки на жесткий диск, это дает нам в руки большие возможности. Итак, загружаемся с LiveCD-диска, открываем окно терминала **Menu • Терминал**. Определяем раздел, на котором установлен Linux Mint, например при помощи `fdisk`:

```
$ sudo fdisk /dev/sda
```

После чего для вывода таблицы разделов нажимаем клавишу **p** и, чтобы выйти, **q**.

Как вариант, можно использовать графическую утилиту Palimpsest (рис. 2.19), которую вызываем из меню **Menu • Администрирование • Дисковая утилита** или просто набрав в окне терминала команду `palimpsest`.

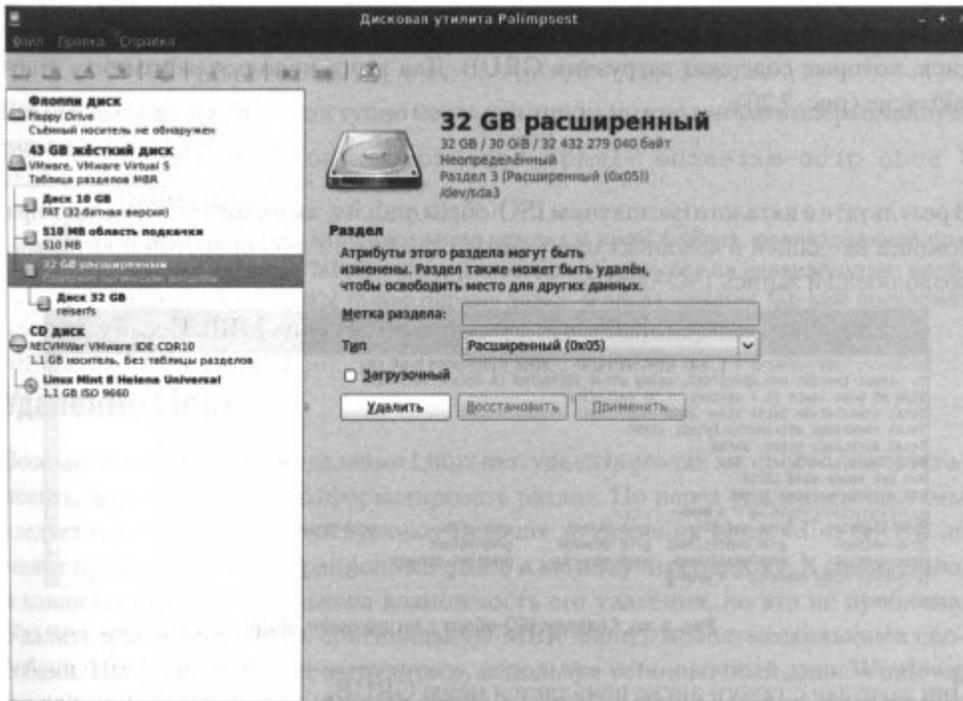


Рис. 2.19. Дисковая утилита Palimpsest

Теперь монтируем раздел, в который установлен Linux Mint как корневой, например, возьмем /dev/sda5:

```
$ sudo mount -t ext3 /dev/hda5 /mnt
$ sudo mount -t proc proc /mnt/proc
$ sudo mount -t sysfs sys /mnt/sys
$ sudo mount -o bind /dev /mnt/dev
```

И переходим в chroot-окружение

```
$ chroot /mnt /bin/bash
```

Теперь мы находимся внутри установленной системы и можем устанавливать загрузчик обычным образом. Для установки GRUB используется команда `grub-install` в качестве указания раздела, в который требуется произвести установку:

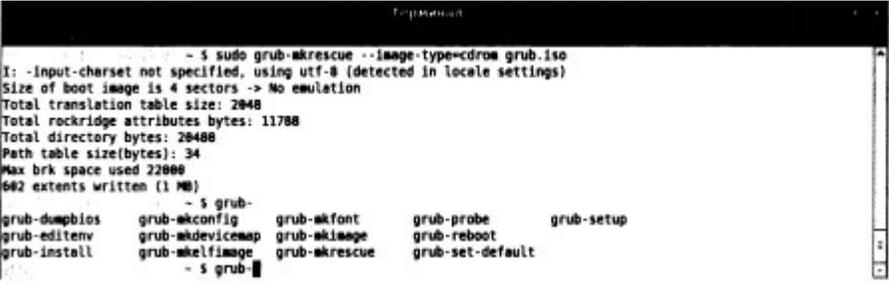
```
$ sudo grub-install /dev/sda
```

Возможно, все получится сразу, но есть вариант, что будет получено сообщение о том, что такое устройство не найдено. Просто добавьте к команде параметр `-recheck`, и все заработает.

Чтобы не загрузаться с LiveCD, можно заранее подготовить дискету или компакт-диск, которые содержат загрузчик GRUB. Для этого используем утилиту `grub-mkrescue` (рис. 2.20)

```
$ sudo grub-mkrescue -image-type=cdrom iso/grub.iso
```

В результате в каталоге `iso` получим ISO-образ `grub.iso`, записываем его на диск при помощи входящей в комплект программы `Brasero` или любой другой программы, позволяющей записать ISO-образа.



```
grub-mkrescue
- $ sudo grub-mkrescue --image-type=cdrom grub.iso
I: -input-charset not specified, using utf-8 (detected in locale settings)
Size of boot image is 4 sectors -> No emulation
Total translation table size: 2048
Total rockridge attributes bytes: 11788
Total directory bytes: 20488
Path table size(bytes): 34
Max brk space used 22000
602 extents written (1 MB)
- $ grub-
grub-dumptions  grub-mkconfig  grub-mkfont    grub-probe     grub-setup
grub-editenv   grub-mkdevicecap  grub-mkimage  grub-reboot
grub-install   grub-mkelfimage  grub-mkrescue  grub-set-default
- $ grub-
```

Рис. 2.20. Создаем ISO-образ с загрузчиком GRUB

При загрузке с такого диска появляется меню GRUB:

```
grub>
```

Необходимо сначала выбрать корневой раздел, обращая внимание на вывод команды, в котором указан тип файловой системы, чтобы быть уверенным, что используете действительно нужный:

```
grub> root (hd0,4)
```

```
Filesystem type is ext2fs, partition type 0x83
```

Если не знаете раздел, то его можно легко найти:

```
grub> find /boot/grub/stage1
```

Теперь можем загрузить конфигурационный файл:

```
grub> configfile /boot/grub/grub.cfg
```

В появившемся меню выбираем нужную операционную систему и загружаемся. Или как вариант сразу устанавливаем GRUB на свое место.

```
grub> setup (hd0)
```

По окончании установки выходим:

```
grub> quit
```

После перезагрузки будет доступно меню, в котором можно выбрать операционную систему.



#### ПРИМЕЧАНИЕ

В GRUB работает автодополнение команд и имен файлов, реализованное при помощи клавиши Tab. Поэтому полностью набирать все на клавиатуре нет необходимости, наберите только первые знаки, и после нажатия Tab вам предложат варианты.

## Удаление Linux

Вообще, ничего хитрого в удалении Linux нет, удалить его так же просто, как и установить, достаточно лишь отформатировать раздел. Но перед удалением системы следует побеспокоиться о загрузчике. Ведь после удаления Linux Mint GRUB не может прочитать конфигурационный файл, и вы получите ошибку. К сожалению, в командах GRUB не заложена возможность его удаления, но это не проблема. Удалить или восстановить оригинальную MBR-запись можно несколькими способами. Например, следует загрузиться, используя установочный диск Windows, и в процессе установки выбрать восстановление системы без форматирования раздела. Но это долгий и рискованный путь. Если под рукой имеется загрузочный

диск со старыми версиями операционных систем от Microsoft Windows 95, Windows 98 или Windows Me, то можно использовать консольную команду

```
> fdisk /mbr
```

которая восстановит MBR. В операционных системах на ядре NT, например Windows XP или Windows Vista, следует загрузиться с установочного диска и при появлении надписи «Вас приветствует мастер установки» нажать клавишу R, чтобы запустить консоль восстановления. Затем ввести команду fixmbr, которая восстановит MBR.

## Глава 3

# Рабочая среда Linux Mint

Рабочий стол GNOME

Утилиты настройки

Итак, в предыдущей главе мы установили операционную систему, теперь самое время перезагрузить компьютер и посмотреть, какой же он, Linux Mint, на самом деле. Если выбран вариант входа в систему с вводом пароля, на некотором этапе процесс загрузки остановится и появится окно графического менеджера GDM (сокр. GNOME Display Manager, сайт проекта <http://projects.gnome.org/gdm/>) — программы, используемой для регистрации пользователей в системе (рис. 3.1).

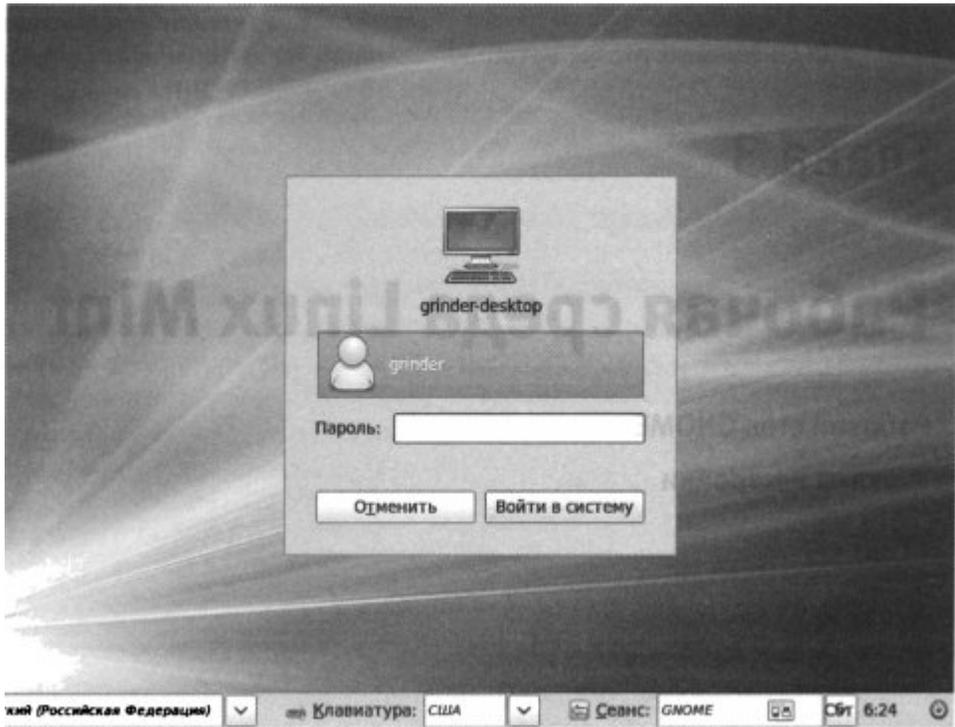


Рис. 3.1. Окно приветствия GDM



#### ПРИМЕЧАНИЕ

Внешний вид экрана приветствия можно легко изменить, позже мы рассмотрим, как это делается.

Выбираем, дважды щелкнув кнопкой мыши в списке на логине, который был создан во время работы мастера установки дистрибутива, и затем вводим пароль.

Обратите внимание на панель внизу. Здесь можно изменить язык **Рабочего стола**, раскладку клавиатуры и в поле **Сессия** — оконный менеджер. По умолчанию загрузка происходит в среде GNOME (других просто нет), если в процессе возникают проблемы, можно выбрать режим **GNOME (безопасный)**, и, наконец, вариант **xterm** загрузит просто окно терминала, запущенного в графической среде X Window (рис. 3.2).

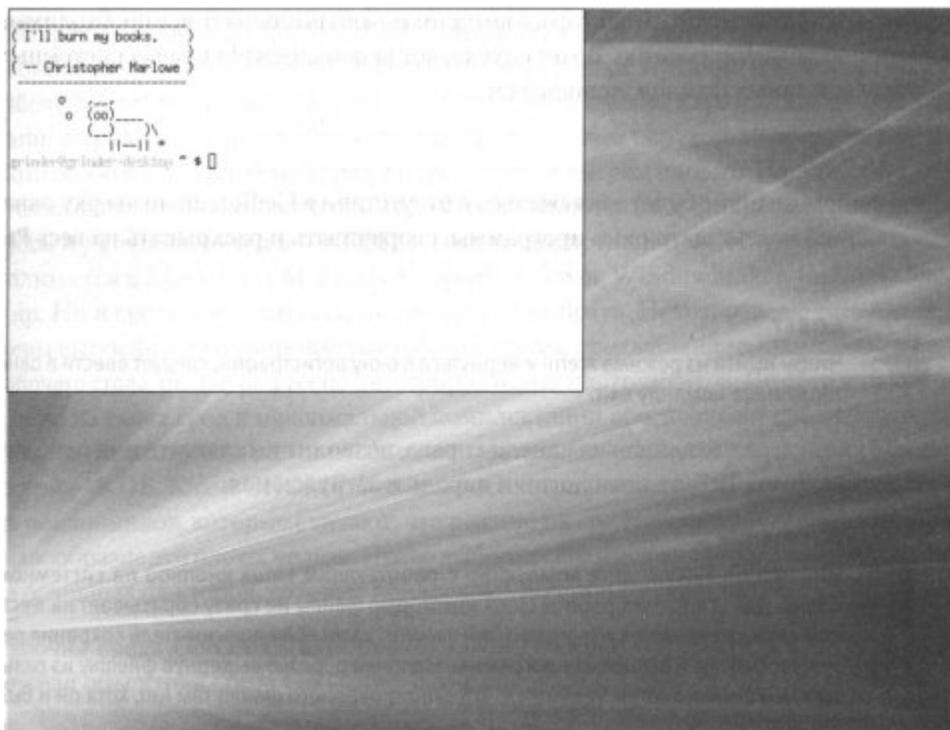


Рис. 3.2. Загрузка в режиме xterm

Такой режим загрузки может понадобиться не только в том случае, если GNOME по определенным причинам не хочет загружаться (всякое ведь бывает), но и, например, если для выполнения некоторых операций (обновление системы, установка пакетов, обработка видео, звука и т. д.) нет необходимости загружать тяжелую графическую среду.

#### ВНИМАНИЕ



Одной из частей причин, по которым не загружается графическая среда, является заполнение корневого раздела, тогда оконный менеджер не может создать в подкаталоге `/tmp` свои временные файлы и загрузка заканчивается с ошибкой. Поэтому контролируйте наличие свободного места, а для хранения своих файлов используйте отдельный раздел жесткого диска.

Причем, введя команду непосредственно в окне терминала, можно загрузить любое приложение.

Например, чтобы отредактировать текст при помощи текстового редактора Gedit, входящего в состав Linux Mint, вводим

```
$ gedit
```

Появится окно текстового редактора, в котором можно выполнять все необходимые операции по редактированию. В том случае, когда понадобится правка системных конфигурационных файлов, используем

```
$ sudo gedit
```

Единственное отличие будет заключаться в отсутствии в Gedit панели сверху окна, позволяющей перемещать окно программы, сворачивать и раскрывать на весь Рабочий стол.

#### СОВЕТ

Чтобы выйти из режима `xterm` и вернуться к окну регистрации, следует ввести в окне терминала команду `exit`.

Красная кнопка, размещенная на панели справа, позволит выключить или перезагрузить компьютер. Все, вводим логин и пароль и загружаемся.

#### ВНИМАНИЕ



Никогда не выключайте компьютер с работающим Linux кнопкой на системном блоке. Для ускорения работы Linux некоторые файлы не сразу сбрасывает на жесткий диск, а держит их в оперативной памяти, даже если пользователь сохранил результат работы и вышел из программы. Например, резко выдернув флешку из разъема USB, можно затем удивиться, что скопированного файла там нет, хотя он и был показан в окне файлового менеджера. Во избежание таких ситуаций следует использовать команду `sync`, при помощи которой данные принудительно сбрасываются на диск. При выключении компьютера эта команда подается автоматически в одном из скриптов. Выключив кнопкой компьютер, вы можете повредить данные.

## Рабочий стол GNOME

В отличие от Windows, в Linux пользователь не привязан к одному интерфейсу и может выбирать оконный менеджер (то есть рабочее окружение) в зависимости от личных предпочтений, наличия свободных системных ресурсов и настроения, а то и вовсе обойтись без графической среды. Это возможно потому, что графическая подсистема X Window System, обеспечивающая все необходимые инструменты и протоколы, не интегрирована в ядро системы, а является отдельным (и необязательным) компонентом. Более того, X Window обеспечивает лишь базовые функции графической среды, такие как отрисовка и перемещение окон на экране, взаимодействие с устройствами ввода, мышкой и клавиатурой.



#### ПРИМЕЧАНИЕ

Некоторое время одним из недостатков Linux как операционной системы, рассчитанной на применение на настольных компьютерах, считалось отсутствие единого интерфейса, сегодня, очевидно, к этому уже привыкли, а многие разработчики дистрибутивов стилизуют интерфейс так, что отличить, какой оконный менеджер сейчас работает, может только знающий пользователь.

Интерфейс и прочие украшения предоставляют специальные программы — оконные менеджеры, которые, собственно, определяют внешний вид и особенности **Рабочего стола** пользователя. В результате оконных менеджеров за два десятка лет написано более сотни, что позволяет выбрать рабочее окружение по своему вкусу. Хотя особо популярных менеджеров среди них не так уже и много. Наиболее используемыми оконными менеджерами для Linux являются kwm (используется в KDE), Metacity (используется в GNOME), twm (стандартный для XFree86), Xfwm (используется в Xfce), IceWM, Fluxbox, Openbox, fvwm, WindowMaker, Enlightenment и др. Но и среди того, что есть, не так просто выбрать. Некоторые оконные менеджеры выросли в интегрированные рабочие среды, предлагающие, кроме собственно **Рабочего стола**, целый ряд тесно интегрированных инструментов. По сути ядро Linux и одна из таких сред в максимальной комплектации обеспечивает среднестатистического пользователя всеми необходимыми приложениями и утилитами. Сегодня их три — KDE, Xfce и GNOME. Каждая имеет свои уникальные возможности и своих поклонников, которые считают, что именно их среда лучше других. Сказать, что разнообразие это плохо, нельзя. Право выбора всего и вся является, наверное, одним из основных факторов, которые привлекают пользователей в Linux.

Рабочая среда GNOME (акроним от GNU Network Object Model Environment, сетевая объектная среда GNU, рис. 3.3) развивается в рамках проекта GNU.

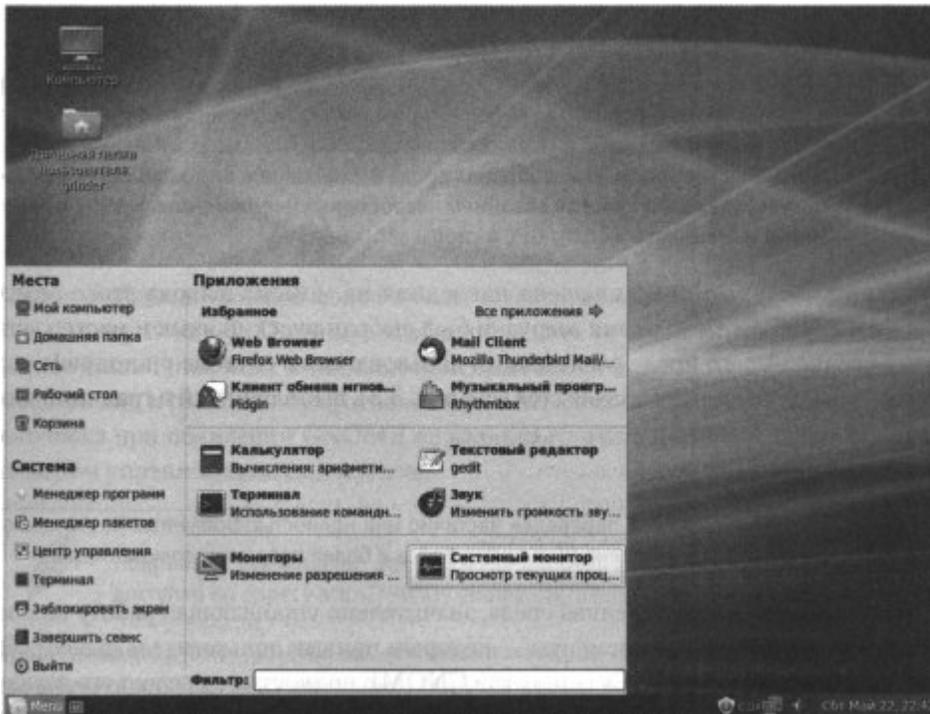


Рис. 3.3. Рабочий стол GNOME после установки системы

Проект GNOME изначально ориентировался на создание полностью свободной среды, доступной любому пользователю вне зависимости от уровня подготовки, языка и физических ограничений. По этому пути и идут разработчики от релиза к релизу. Сайт проекта, на котором доступны последние новости, приложения и ссылки на документацию, размещен по адресу <http://www.gnome.org/>.

В рамках проекта разрабатывается как множество приложений для конечных пользователей, которые интегрируются в рабочую среду, делая ее более удобной, так и инструменты для разработчиков. От другой популярной среды KDE GNOME отличает практичность и простота интерфейса. Все требования к рабочей среде изложены в документе «Руководство по созданию человеческого интерфейса GNOME» (англ. GNOME Human Interface Guidelines, сокращенно — HIG), в котором даны рекомендации разработчикам по созданию высококачественного и удобного графического интерфейса. Найти HIG можно по адресу <http://developer.gnome.org/projects/gup/hig/>. Именно HIG определяет, какие настройки должны быть доступны пользователю, а что лишнее. В результате в приложениях GNOME на порядок меньше возможностей по доводке интерфейса штатными средствами. Именно поэтому любители изменять настройки предпочитают больше KDE, хотя на сегодня доступны инструменты, позволяющие перенастроить среду GNOME практически на свой вкус или наличие ресурсов.



#### ПРИМЕЧАНИЕ

GNOME появился как ответ проекту KDE, который использовал библиотеку Qt, а в первое время Qt была доступна по закрытой лицензии, поэтому созданная среда критиковалась многими пользователями, в том числе и Линусом Торвальдсом. Они говорили, что свободная среда и входящие в ее состав программы не могут создаваться с использованием несвободных инструментов. В 1997 г. лицензия Qt изменилась на GNU GPL и споры затихли.

В поставку Linux Mint 9 включена последняя на момент выхода этого релиза GNOME 2.30.0. Если во время загрузки был выбран русский язык и мастер установил все пакеты, то после регистрации пользователя в системе он получит корректно локализованный Рабочий стол (см. рис. 3.3), выполненный в традиционном зелено-коричневом стиле.



#### ПРИМЕЧАНИЕ

Интерфейс GNOME переведен частично или полностью более чем на 60 языков, на русский переведено 99 % интерфейса и более половины документации.

GNOME — очень дружелюбная среда, значительно упрощающая работу пользователя. Она содержит все элементы, — которым привык пользователь, ранее работавший только в Windows. Суть проекта GNOME полностью передает его девиз — just work, что в вольном переводе означает «работает, и все». Иными словами, пользователь получает готовую к работе среду, которую не нужно перестраивать.

Изначально среда GNOME состоит из панели GNOME, располагающейся обычно вверху экрана, панели задач (внизу) и собственно Рабочего стола (рис. 3.4). Панель GNOME состоит из нескольких меню, каждое из которых выполняет свою функцию — доступ к приложениям, системным настройкам и основным каталогам.

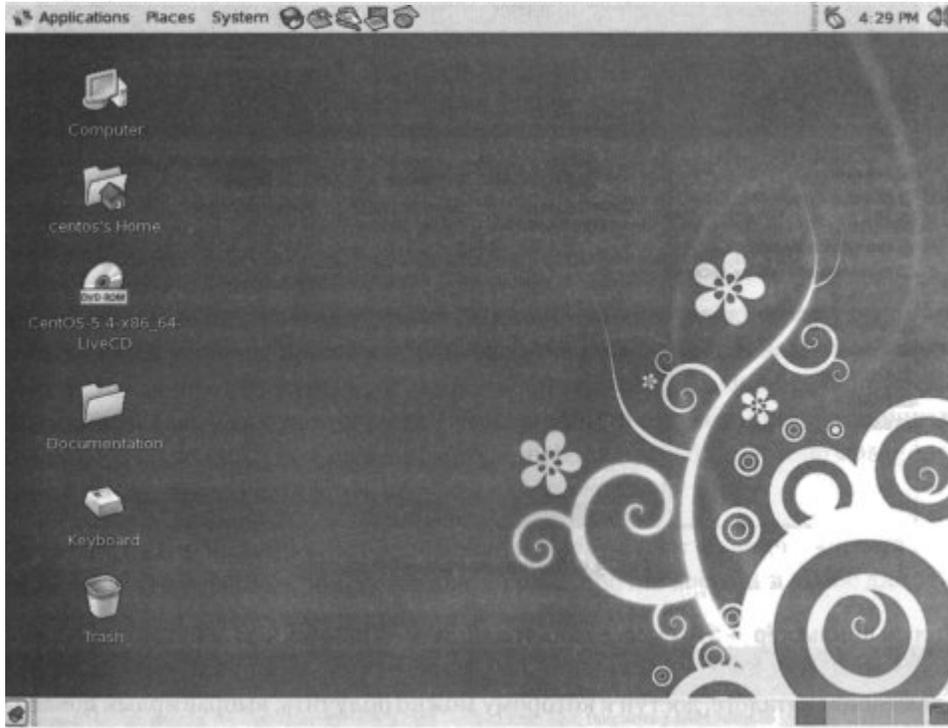


Рис. 3.4. Среда GNOME в дистрибутиве CentOS

Такое размещение панелей может показаться непривычным (так оно и есть) и сбивает с толку новичка. Поэтому, чтобы упростить процесс миграции с Windows, разработчики Linux Mint полностью изменили настройки Рабочего стола, стилизовав внешний вид и расположение основных элементов под Windows. В итоге культурного шока при появлении GNOME не возникает, пользователь находит основные элементы управления на своем месте.



#### ПРИМЕЧАНИЕ

Подробное руководство пользователя рабочей среды GNOME на русском языке доступно по адресу <http://library.gnome.org/users/user-guide/>.

При первой загрузке системы вы увидите окно приветствия Linux Mint. Приведенные в нем ссылки на документацию, форумы поддержки помогут быстрее освоиться с новой системой и получить ответы практически на любые вопросы. Чтобы окно больше не выводилось, снимите флажок Показывать это окно при загрузке системы.

На **Рабочем столе** размещены два ярлыка. После выбора ярлыка **Компьютер** откроется файловый менеджер Nautilus, в окне которого будут показаны все подключенные в настоящий момент съемные носители (рис. 3.5), привод компакт-дисков и ярлык **Файловая система**, который откроет окно корня файловой системы.

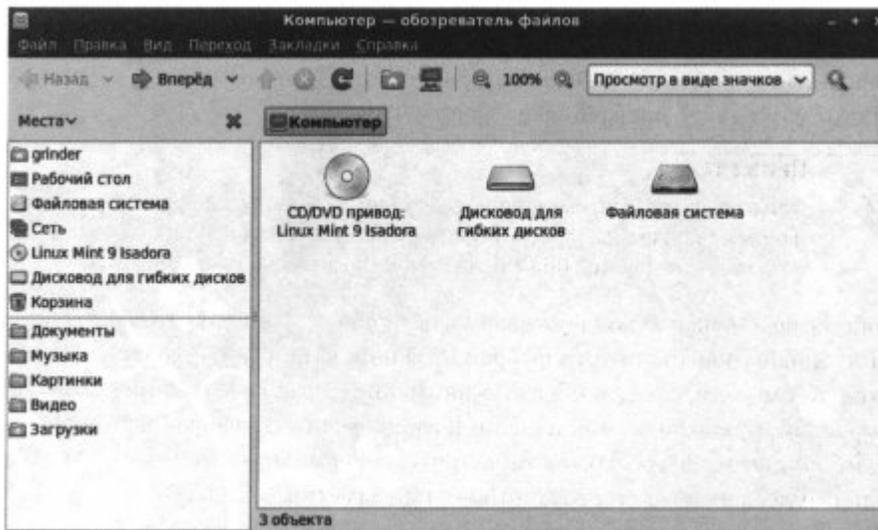


Рис. 3.5. Выбираем ярлык Компьютер

Ярлык **Компьютер** — это скорее дань традициям Windows, чем действительно необходимость. Как я уже говорил в главе 2, все свои данные пользователь хранит в домашнем каталоге, доступ к которому можно получить, выбрав ярлык **Домашняя папка пользователя**.

В боковом окне Nautilus показываются основные каталоги файловой системы и доступные ресурсы, к которым можно быстро перейти, просто щелкнув кнопкой мыши на ярлыке. Выбрав ярлык **Сеть**, вы получите доступ к ресурсам сети Windows (рис. 3.6).

Последовательно выбираем группу, конкретный компьютер и щелкаем кнопкой мыши на нужной сетевой папке, после чего она откроется в этом же окне. Если вставить флешку в разъем USB или диск в привод компакт-дисков, ярлык появится на **Рабочем столе** и в окне Nautilus. Все происходит интуитивно, и какой-либо подготовки не требуется. Кстати, в реализации этой функции в Linux Mint есть отличие от принятого в GNOME. В GNOME традиционно все каталоги, в которые переходит пользователь, открываются в новом окне. Причем представления файлов (значки, список или компактный) привязаны к конкретному окну. Месторасположение также фиксируется, и при следующем запуске окно выглядит как в прошлый раз. Такое размещение называется **пространственно-ориентированным**, и его идея состоит в том, что на настоящем **Рабочем столе** все документы также

лежат в том же месте, где их оставили, не замещая друг друга. Считается, что новичку так будет легче разобраться, учитывая, что в Linux основные перемещения пользователя происходят в домашнем каталоге и количество открытых окон обычно невелико. Однако опытные пользователи в большинстве своем предпочитают открывать каталоги в одном окне. Стандартными способами перенастроить поведение не так просто, необходимо править конфигурационный файл Nautilus, размещенный в `~/.gconf/apps/nautilus/preferences/`. Но это нам и не нужно, поскольку в Linux Mint такое поведение выбрано изначально.

#### ПРИМЕЧАНИЕ

**Файлы или каталоги, имя которых начинается с точки, считаются в Linux скрытыми и по умолчанию в файловом менеджере не отображаются. Чтобы увидеть их, нужно установить флажок Вид • Показывать скрытые файлы (Ctrl + H).**

В процессе перемещения поддерживается история, поэтому можно быстро вернуться на нужный уровень, просто выбрав название каталога на панели или в меню **Переход**. Кроме того, следует обратить внимание на раскрывающийся список, расположенный чуть выше левой панели. В зависимости от выбранного значения на панели будут показаны ресурсы компьютера, информация об отмеченном на правой панели файле, дерево каталогов, история или заметки.

Среда GNOME и большинство приложений поддерживают функцию «перетащить и бросить», что упрощает копирование файлов и прочие операции, плюс для этих же целей можно задействовать контекстное меню.

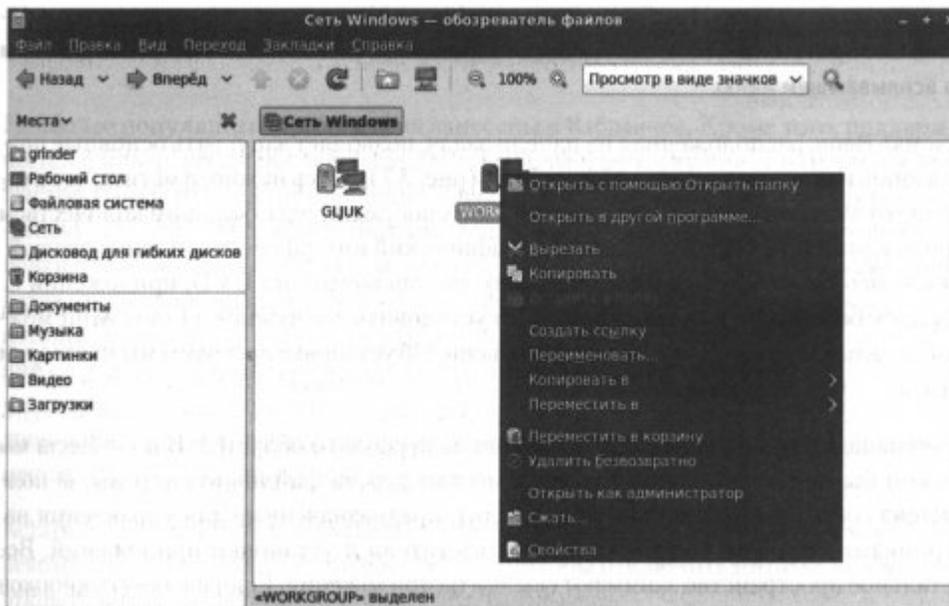


Рис. 3.6. Ресурсы Windows в окне Nautilus

**ВНИМАНИЕ**

Все сменные носители монтируются в дерево файловой системы, поэтому просто вынуть флешку или компакт-диск нельзя. И хотя Linux Mint поддерживает автоматическое размонтирование ресурсов, при обращении лучше для этих целей использовать пункт контекстного меню **Безопасно отключить носитель** и только затем вынимать флешку из разъема.

При щелчке на файле автоматически запускается приложение, связанное с этим типом данных, — аудио- или видеопроигрыватель, текстовый редактор и т. д. Если компакт-диск имеет определенный формат, например DVD, то сразу загружается видеопроигрыватель.

При щелчке правой кнопкой мыши на выбранном объекте появляется контекстное меню, пункты которого позволяют выполнить большинство стандартных операций — переместить, копировать, удалить файл, открыть в другом окне или программе, в том числе и в терминале, сжать при помощи архиватора, отправить (по электронной почте, Bluetooth, записать на компакт-диск), просмотреть и отредактировать свойства, открыть общий доступ по сети к каталогу.

**ВНИМАНИЕ**

Если права доступа не позволяют открыть файл или каталог от имени текущей учетной записи, прочитать его не получится. Такие файлы в Nautilus помечаются крестиком. В меню Nautilus доступен пункт **Открыть от имени администратора**, воспользуйтесь им.

Для настройки меню Nautilus в комплект Linux Mint входит утилита Nautilus Actions Configurations Tools, запускаемая через **Центр управления • Настройка действия во всплывающем меню**.

Кнопка **Menu**, расположенная на панели задач, позволяет запустить основные приложения, идущие в комплекте Linux Mint (рис. 3.7). Здесь нужно отметить, что в отличие от Windows любой дистрибутив Linux поставляется с большим количеством приложений и утилит, имеющих как графический интерфейс, так и запускающийся в консоли. Конечно, в дистрибутивах, поставляемых на DVD, приложений на порядок больше, но при необходимости установить все нужное в Linux Mint можно буквально одним щелчком кнопкой мыши. Об установке программ мы поговорим в гл. 5.

Появившееся меню визуально разделено на несколько областей. В поле **Места** мы можем быстро перейти к некоторым местам дерева файловой системы. В поле **Система** собраны ярлыки основных утилит, предназначенные для управления настройками системы, **Рабочего стола** пользователя и установки приложений. Все остальное пространство занимают ссылки на приложения. Если щелкнуть кнопкой мыши на заголовке вкладки, она свернется в значок и не будет занимать место. При большом количестве приложений это спасает. Для удобства выбора приложе-

ния разделены на группы — **Аудио и Видео**, **Графика**, **Интернет**, **Офис**, **Системные утилиты**, **Стандартные**, **Администрирование** и **Параметры**. В отдельной вкладке собраны ярлыки для запуска всех приложений. Порядок работы прост: выбираем подпункт и затем необходимую утилиту. Большинство ярлыков подписано и включает название программы и ее назначение, что позволяет сориентироваться среди неизвестных программ. Например: **Brasero Запись и копирование CD и DVD**.

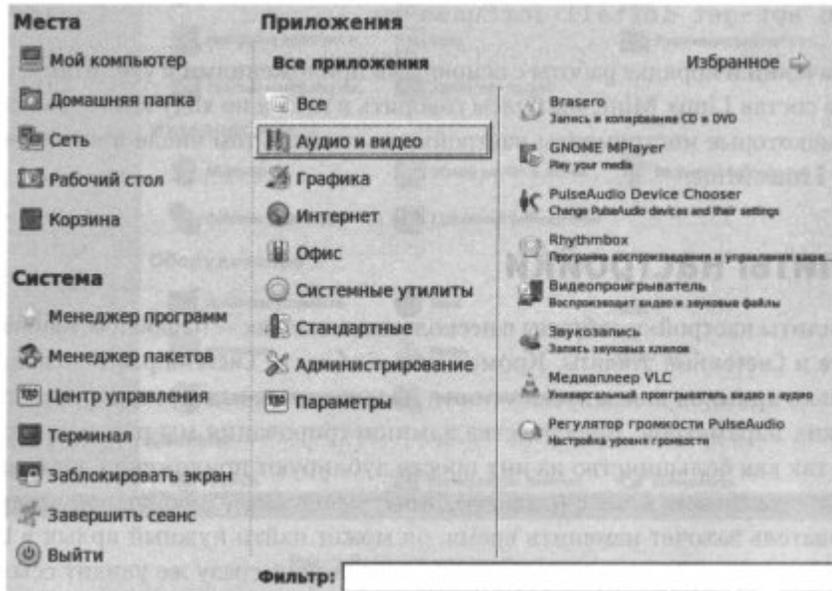


Рис. 3.7. Меню Linux Mint

Наиболее популярные приложения занесены в **Избранное**. Кроме того, поддерживается фильтр: вводя последовательно символы в полоску внизу окна, мы можем быстро найти нужное приложение. Особо хочу отметить, что в меню собраны ярлыки для запуска приложений, имеющих графический интерфейс, после установки программы обычно сами заносят сюда свои ярлыки, но возможны ситуации, когда этого не происходит. К тому же Linux включает сотни консольных утилит, которые необходимо запускать в терминале. И главное — большинство приложений с графическим интерфейсом являются по сути надстройками к консольным утилитам. Отредактировать список приложений в главном меню можно, обратившись к **Параметры • Главное меню**, где можно добавлять, удалять, скрывать, перемещать пункты, создавать новые подменю и т. д.

Кстати, если открыть окно терминала, то вас встречают небольшие цитаты (см. рис. 3.1), некоторым они мешают, некоторых веселят. Чтобы отключить цитаты, необходимо открыть в текстовом редакторе файл `/etc/bash.bashrc`:

```
$ sudo gedit /etc/bash.bashrc
```

В открывшемся файле в самом низу находится строка

```
/usr/bin/mint-fortune
```

Удаляем ее или комментируем, установив в начале знак #, сохраняем и закрываем файл. Для русификации цитат установите пакет fortunes-ru:

```
$ sudo apt-get install fortunes-ru
```

О назначении и порядке работы с основными приложениями и утилитами, входящими в состав Linux Mint, мы будем говорить в гл. 5 и по ходу книги, сейчас разберем некоторые инструменты настройки системы, в том числе и свойственные только Linux Mint.

## Утилиты настройки

Все утилиты настройки собраны в нескольких вкладках — **Параметры**, **Администрирование** и **Системные утилиты**. Кроме этого в области **Система** расположены еще несколько ярлыков для запуска утилит, предназначенных для настройки специфических параметров. Все средства администрирования мы рассматривать не будем, так как большинство из них просто дублируют приложения, вызываемые из **Центра управления GNOME** (рис. 3.8). Это сделано для удобства, например если пользователь захочет изменить время, он может найти нужный ярлык в **Центре управления** или, зайдя в подменю **Администрирование**, сразу же увидит ссылку на **Дата и время**. К некоторым настройкам будем возвращаться по ходу прочтения книги. Вообще, могу сказать, что человеку, знакомому с компьютером, понадобится не так много времени, чтобы освоиться с настройками при помощи графических утилит.



### ПРИМЕЧАНИЕ

Некоторые операции потребуют прав администратора, первый зарегистрировавшийся в системе пользователь имеет такие права. Чтобы их подтвердить, введите свой пароль на запрос системы. О правах поговорим в следующей главе.

Обратите внимание на поле **Фильтр**, вводя в него параметры, можно быстро найти нужную настройку.



### ПРИМЕЧАНИЕ

Все приложения из меню можно запустить и другим способом, просто введя название исполняемого файла в окне терминала (gnome-terminal). Например, чтобы запустить Центр управления, вводим `gnome-control-center`. В будущем я буду давать команду для некоторых приложений в скобках. Самостоятельно узнать строку запуска можно, просмотрев список процессов. Как это сделать, мы поговорим в следующей главе.

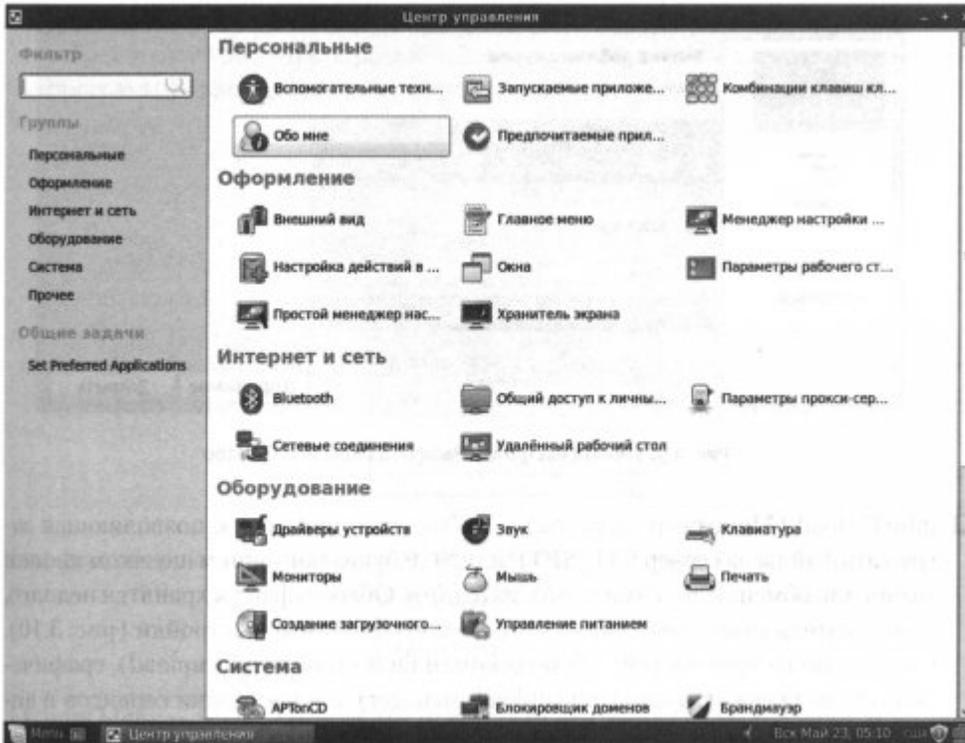


Рис. 3.8. Центр управления

Linux Mint отличается от Ubuntu наличием нескольких утилит собственной разработки, делающих эту среду удобнее и понятнее для пользователя. Учитывая то, что именно они во многом определяют выбор пользователями Linux Mint вместо любого другого дистрибутива, перечислим их отдельно.

- `mintInstall` (Менеджер программ) — очень удобный в работе менеджер программ для установки приложений, по принципу построения отличается от идущей в комплекте другой программы, используемой для этих же целей, — `Synaptic`.
- `mintUpdate` (Менеджер обновлений) — представляет собой программу, активируемую при нажатии на небольшом, имеющем вид замка апплете, который размещен на панели задач. Он подключается к репозиториям и проверяет наличие новых версий системных компонентов, после чего программа предлагает их пользователю для установки.
- `mintDesktop` — небольшая программка, позволяющая настроить внешний вид **Рабочего стола**: значки, выводимые по умолчанию, внешний вид окон и интерфейса (рис. 3.9).

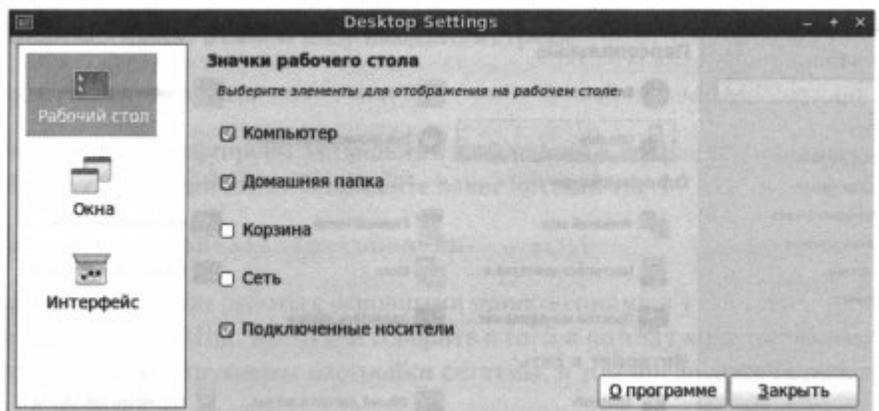


Рис. 3.9. Утилита настройки Рабочего стола mintDesktop

mintUpload (Менеджер загрузок) — небольшая программа, позволяющая загружать файлы на сервер FTP, SFTP или SCP буквально одним щелчком кнопки мыши для обмена с другими пользователями. Обычно файлы хранятся недолго, пользователь самостоятельно создает все необходимые настройки (рис. 3.10). Состоит он из трех частей: утилиты командной строки (mintupload), графического менеджера загрузок (mintupload-manager) для настройки сервисов и апплета для собственно загрузок (mintupload-file-uploader).

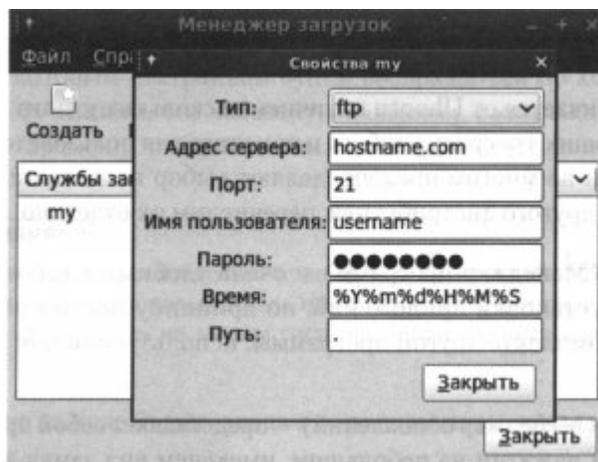


Рис. 3.10. Окно настроек Менеджера загрузок

- Mint-Space — платная версия сервиса mintUpload, предоставляющая больше места на диске и больший срок хранения файлов.
- mintWifi — программа для установки драйверов для большого количества беспроводных сетевых карт. Ее можно запустить и в консоли, введя `sudo mintwifi`.

- minNanny (Блокировщик доменов) — простенькая программа, позволяющая заблокировать доступ к определенному домену, просто введя его название в окне программы (рис. 3.11).

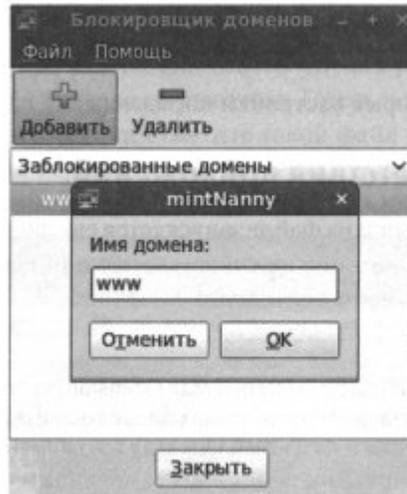


Рис. 3.11. Окно mintNanny

- mintMenu — меню, которое мы открывали для запуска приложений, и заменяющее стандартное меню GNOME. В меню также находится индикатор заполнения корзины.
- mintBackup (Средство резервного копирования) — очень простая программа, позволяющая одним щелчком кнопкой мыши создать резервную копию файлов домашнего каталога и установленных программ и также легко восстановить, если такая необходимость настанет (рис. 3.12).

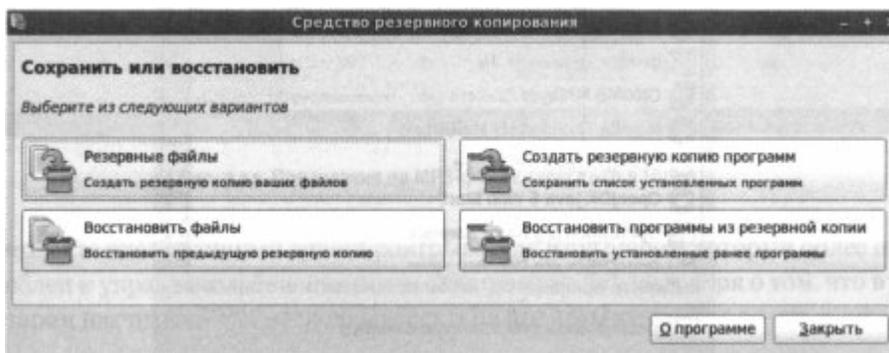


Рис. 3.12. Программа mintBackup

- mintWelcome — окно приветствия, которое встречает пользователя при входе в систему.

В ранних версиях системы были доступны и некоторые другие утилиты — `mintAssistant`, `mintDisk` и `mintConfig`, но сегодня они заменены аналогами либо в них уже нет необходимости, так как функции реализованы в штатных утилитах GNOME. Утилиты от разработчиков Linux Mint покрывают основные требования пользователей к настройкам, они просты и удобны в применении.

Теперь разберем некоторые настройки по задачам.

## Настройка соответствия приложений

При щелчке кнопкой мыши на файле запускается связанное с типом данных MIME приложение. Если приложение не сопоставлено или сопоставлено неправильно, то это придется сделать самостоятельно.



### ПРИМЕЧАНИЕ

**MIME** (англ. **M**ultipurpose **I**nternet **M**ail **E**xtension, многоцелевые расширения почты Интернета) — стандарт, изначально описывающий правила передачи по электронной почте разного типа вложений, позже его применение несколько расширили.

После установки нового приложения настройки могут автоматически измениться, и вам, наверное, захочется вернуть их обратно. Все просто. При запуске файла, расширение которого неизвестно, появится запрос на выбор приложения, нажимаем кнопку **Выбор приложения** и в появившемся окне отмечаем нужное (рис. 3.13).

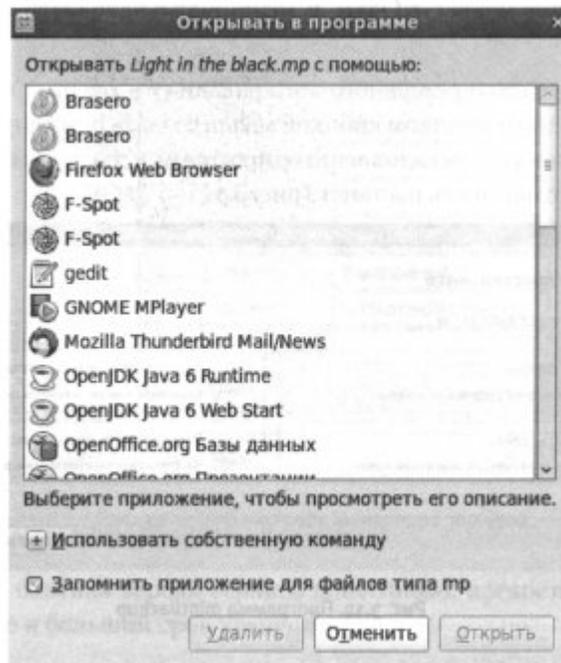


Рис. 3.13. Сопоставляем файл с приложением

Смотрим, чтобы был установлен флажок **Запомнить приложение для файлов типа**. Если приложений несколько, то можно выбрать любое при помощи пункта контекстного меню **Открыть в другой программе**, этот же пункт впоследствии позволяет сопоставить с данным типом файлов другое приложение.

Например, по умолчанию при щелчке кнопкой мыши на музыкальном файле с расширением mp3 запускается видеопроигрыватель Totem, который исправно справляется с задачей. Кроме этого, если отметить такой файл в окне Nautilus в левой части (при установленном переключателе **Сведения**), появляется сопоставленное действие в **«Видеопроигрыватель»** (рис. 3.14).

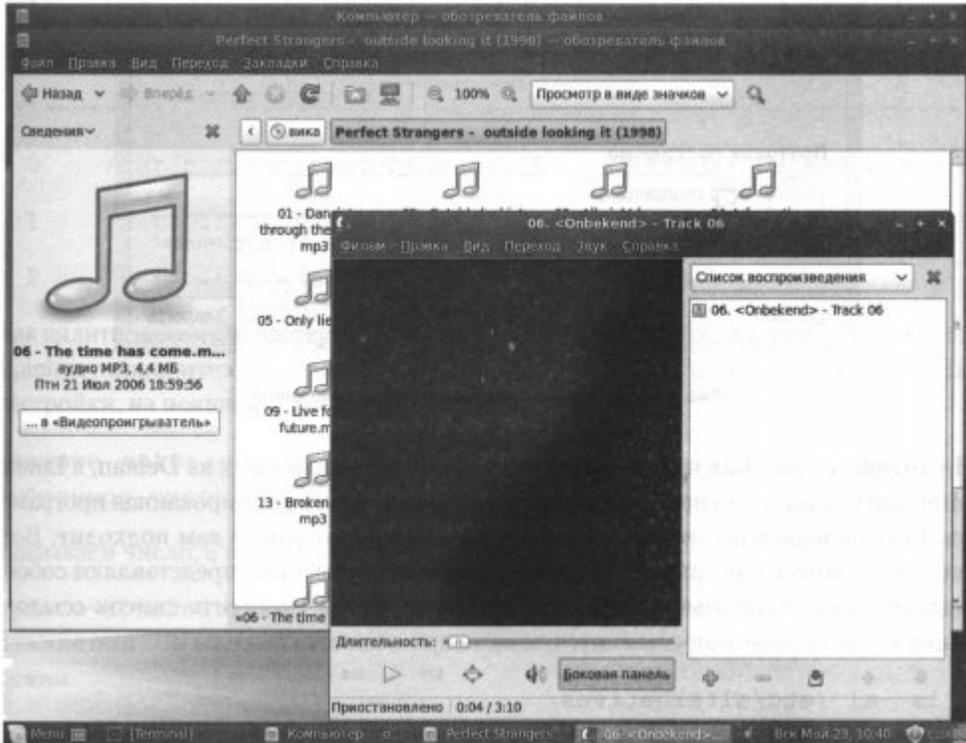


Рис. 3.14. При щелчке на MP3-файле запускается Totem

В комплекте идет отличный аудиопроигрыватель Rhythmbox, который более приспособлен к управлению большими коллекциями аудио, не говоря о том, что в репозитории насчитывается несколько десятков проигрывателей.

Кроме контекстного меню в Linux Mint доступна и другая возможность установить приложения, запускаемые по умолчанию, — **Параметры • Предпочитаемые приложения**. Появляющееся окно позволяет задать приложения, которым будет отдано предпочтение, — браузер, почтовый клиент (рис. 3.15), мультимедиа, эмулятор

терминала и некоторые функции специальных возможностей. Доступно редактирование команды для запуска и поведения приложений, например можно установить открытие веб-страницы в новом окне или новой вкладке.

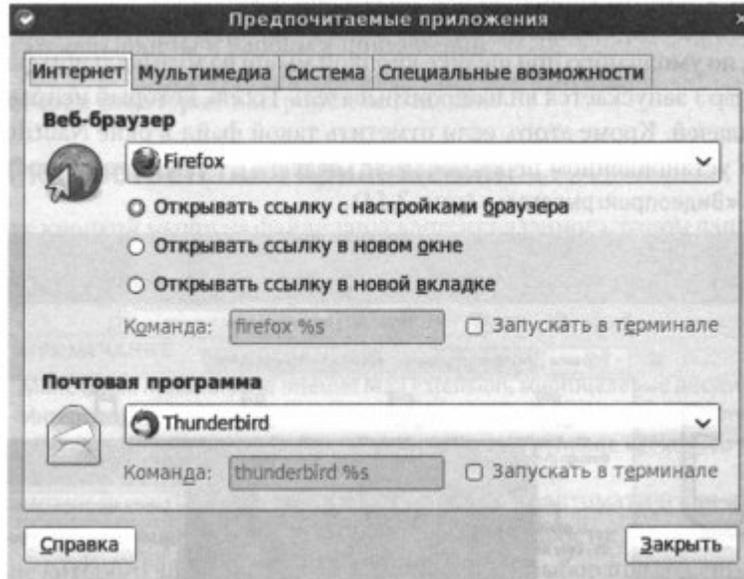


Рис. 3.15. Установка предпочитаемых приложений

Но это еще не все. Как и во многих дистрибутивах, основанных на Debian, в Linux Mint при выполнении некоторых действий вызывается ассоциированная программа. Вполне вероятно, что не всегда предлагаемая программа вам подходит. Все ассоциированные программы и соответствующие переменные представляют собой ссылки, расположенные в каталоге `/etc/alternatives/`. Получить список ссылок файлов, на которые они ссылаются, можно при помощи команды `ls`:

```
$ ls -al /etc/alternatives/

lrwxrwxrwx  1 root root    16 2010-05-22 13:13 x-www-browser
-> /usr/bin/firefox

lrwxrwxrwx  1 root root    14 2010-05-22 13:13 gnome-text-
editor -> /usr/bin/gedit
```

Приготовьтесь, вывод будет большим. О файлах и ссылках мы поговорим подробнее в следующей главе.

#### СОВЕТ



Чтобы выводить информацию построчно, используйте команду `less`, например `ls - al | less`.

В каталоге `/etc/alternatives/` некоторым предопределенным переменным указываются приложения. Например, пользователь запускает браузер, проверяется файл-ссылка `/etc/alternatives/x-www-browser`, которая в нашем случае указывает на браузер Mozilla Firefox, точнее — на исполняемый файл `/usr/bin/firefox`. При необходимости можно легко сопоставить любую переменную с другим приложением. Причем вручную делать ничего не нужно, для этих целей используем утилиту `update-alternatives`.

```
$ sudo update-alternatives --config x-www-browser
```

Есть 2 варианта для альтернативы `x-www-browser` (предоставляет `/usr/bin/x-www-browser`).

Выбор	Путь	Приор	Состояние
* 0	<code>/usr/bin/epiphany-browser</code>	85	автоматический режим
1	<code>/usr/bin/epiphany-browser</code>	85	ручной режим
2	<code>/usr/bin/firefox</code>	40	ручной режим

Как видите, имеем две альтернативы: браузеры Firefox и Epiphany. После установки Epiphany приоритет был изменен на него. Мне нравится Firefox. Чтобы изменить настройки, на появившийся запрос

Нажмите `enter`, чтобы сохранить текущий выбор[\*], или введите выбранное число:

указываем число, в нашем случае это 2.

```
update-alternatives: используется /usr/bin/firefox для предоставления /usr/bin/x-www-browser (x-www-browser) в ручной режим
```

Чтобы оставить все настройки по умолчанию, просто жмем клавишу **Ввод**.

Вызвав `sudo update-alternatives --all`, мы можем последовательно перенастроить все соответствия.

## Обустроиваем Рабочий стол

Как я уже говорил, рабочую среду можно изменить практически до неузнаваемости и сделать ее более удобной. Например, чтобы изменить фон **Рабочего стола**, достаточно щелкнуть правой кнопкой мыши и в контекстном меню выбрать пункт **Изменить фон рабочего стола** или, как вариант, вызвать **Центр управления** и перейти к области **Оформление**. Появится окно **Параметры внешнего вида**, которое состоит из

четырёх вкладок: **Тема**, **Фон** (рис. 3.16), **Шрифты** и **Визуальные эффекты**. Просто дважды щелкнув кнопкой мыши, отмечаем нужную картинку. При необходимости, используя открывающийся список **Стиль**, масштабируем картинку. По умолчанию в Linux Mint для оформления окон применяется тема Shiki-Wise, при смене обоев она может смотреться негармонично, переходим в **Тема** и выбираем наиболее подходящую.



Рис. 3.16. Настраиваем фон Рабочего стола в GNOME

Ссылка внизу страницы ведет на сайт проекта <http://art.gnome.org/>, с которого можно скачать дополнительные темы, фоновые картинки и значки для оформления **Рабочего стола**. Процесс установки обоев, полученных с сайта, очень прост, как, впрочем, и любого другого рисунка. Выбираем понравившуюся картинку, нажимаем кнопку **Go**, чтобы скачать ее на жесткий диск компьютера. Когда картинка откроется в браузере, щелкаем правой кнопкой мыши и в появившемся контекстном меню выбираем пункт **Сделать фоновым рисунком рабочего стола** (рис. 3.17) или сохраняем рисунок на жесткий диск. Как видите, все просто.



Рис. 3.17. Устанавливаем рисунок в качестве фонового на Рабочем столе

Кроме обоев на сайте <http://art.gnome.org/> можно выбрать рисунок, который будет встречать пользователя в окне для регистрации, фоновый рисунок при загрузке системы, значки и т. д. Установка так же проста. Выбираем подходящую тему и нажимаем на ссылку для ее загрузки, в появившемся окне по умолчанию будет предложено открыть ссылку при помощи **Установщик тем (по умолчанию)**, соглашаемся и через некоторое время получаем новую тему.

Кроме этого несколько тем оформления, значки и прочие украшения доступны в репозитории дистрибутива. Как устанавливать приложения при помощи пакетного менеджера, рассмотрим в гл. 5.

В меню **Параметры • Окна** устанавливается поведение открытого окна при щелчке кнопкой мыши на заголовке, получении фокуса и некоторых других действиях пользователя.

В состав Linux Mint включен композитный оконный менеджер Compiz Fusion (<http://www.compiz.org/>) с коллекцией дополнительных плагинов. Compiz Fusion использует OpenGL и 3D-компонент графической карты для отрисовки трехмерного **Рабочего стола**. Это сделано не для красоты и удобства, как многие считают

и как представляют это пользователям, а с практическим расчетом. Ведь сегодня графические карты, которые стоят в системных блоках компьютеров, по мощности не уступают центральному процессору, но при отрисовке двухмерного **Рабочего стола** используется не более 5 % их возможностей (а то и меньше), остальное задействуется только в играх. В итоге пользователь, заплативший немало денег, по сути, тратит их на нагрев помещения. Чтобы задействовать максимальный потенциал, придумали трехмерные **Рабочие столы**, которые параллельно дают больше возможностей и удобства при общении с компьютером и выглядят более футуристично. Причем в Linux реализации подобных решений являются надстройкой или частью графической подсистемы, вероятно, поэтому требования при активации 3D в Linux предъявляются только к видеокарте, а не к остальному оборудованию компьютера. При включенных трехмерных эффектах даже на относительно слабой системе каких-либо замедлений в работе не наблюдается.

Проект Compiz Fusion появился при слиянии двух проектов — Beryl (прекратил свое развитие) и дополнений к подобному композитному оконному менеджеру Compiz — Compiz Extras. Ранее активация трехмерных эффектов была не так проста и часто требовала некоторой подготовки пользователя. Сегодня достаточно, чтобы видеокарта поддерживала 3D-ускорение. Если у вас современная видеокарта, к которой доступны драйвера, то проблем с активацией не будет. Причем в большинстве случаев вам нет необходимости устанавливать драйверы самостоятельно, просто попробуйте перейти в **Визуальные эффекты** и активировать нужный уровень (рис. 3.18). Вначале будет произведен поиск доступных драйверов, и при наличии таковых они активируются автоматически.

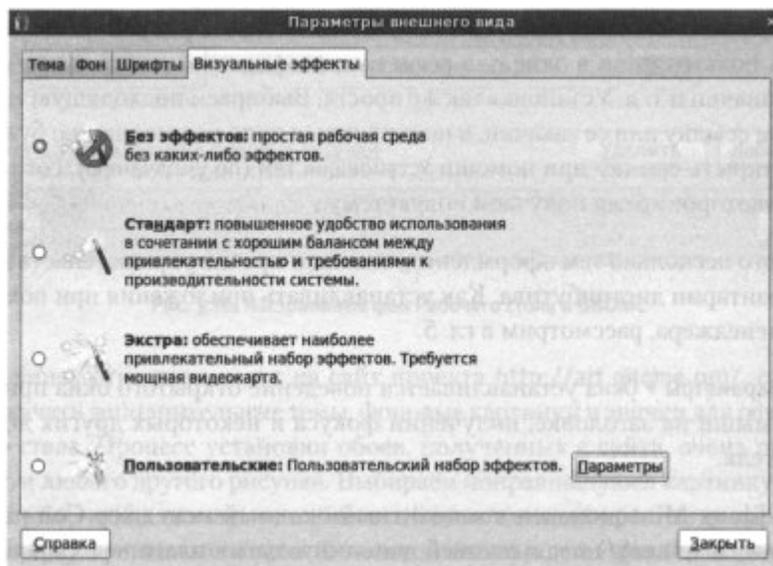


Рис. 3.18. Активируем трехмерные эффекты

Предусмотрено четыре варианта: **Без эффектов**, **Стандарт**, **Экстра** и **Пользовательский**. Выбираем нужный в зависимости от мощности видеокарты и личных предпочтений. Последний позволяет точно указать те эффекты, которые необходимо активировать. Нажав рядом кнопку **Параметры**, получим возможность произвести настройки в **Простом менеджере настройки CompiZConfig** (simple-ccsm, рис. 3.19).

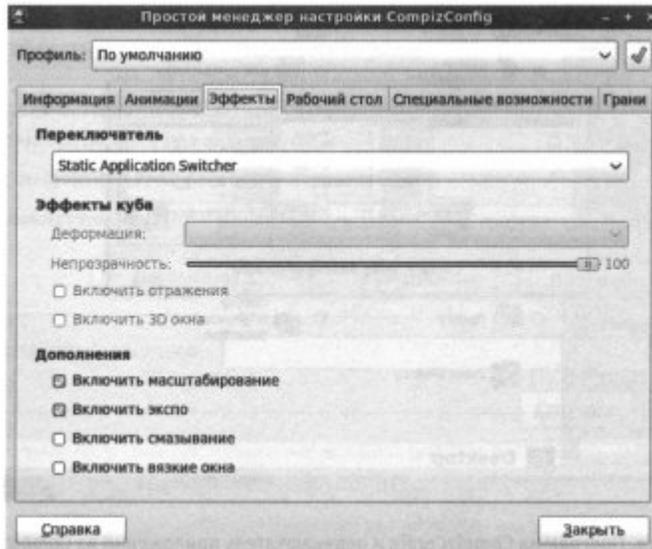


Рис. 3.19. Окно простого менеджера настройки CompiZConfig

Перемещаясь по вкладкам, настраиваем эффекты анимации при открывании и закрывании окна (скольжение, угасание и др.), прозрачность, эффект куба, специальные возможности (масштабирование экрана или области), а также настраиваем поведение при приближении мышки к одной из граней **Рабочего стола**. Кроме того, есть расширенная программа настройки спецэффектов CompiZ Fusion, вызываемая командой меню **Параметры • Менеджер настройки CompiZConfig** (рис. 3.20). Здесь уже можно настроить совместимость с различными оконными менеджерами (GNOME и KDE), комбинации клавиш, различные эффекты и дополнения.

И это только основные настройки, осталось выбрать в меню **Параметры • Хранитель экрана** заставку, которая будет отображаться в случае бездействия пользователя. При установленном флажке **Блокировать экран при активном хранителе экрана** потребуется ввести пароль, чтобы опять получить доступ к **Рабочему столу**. Это удобно в том случае, когда вы работаете в помещении, в котором могут находиться посторонние, чтобы они не имели возможности воспользоваться компьютером в ваше отсутствие. В домашней обстановке такая блокировка, как правило, только мешает.

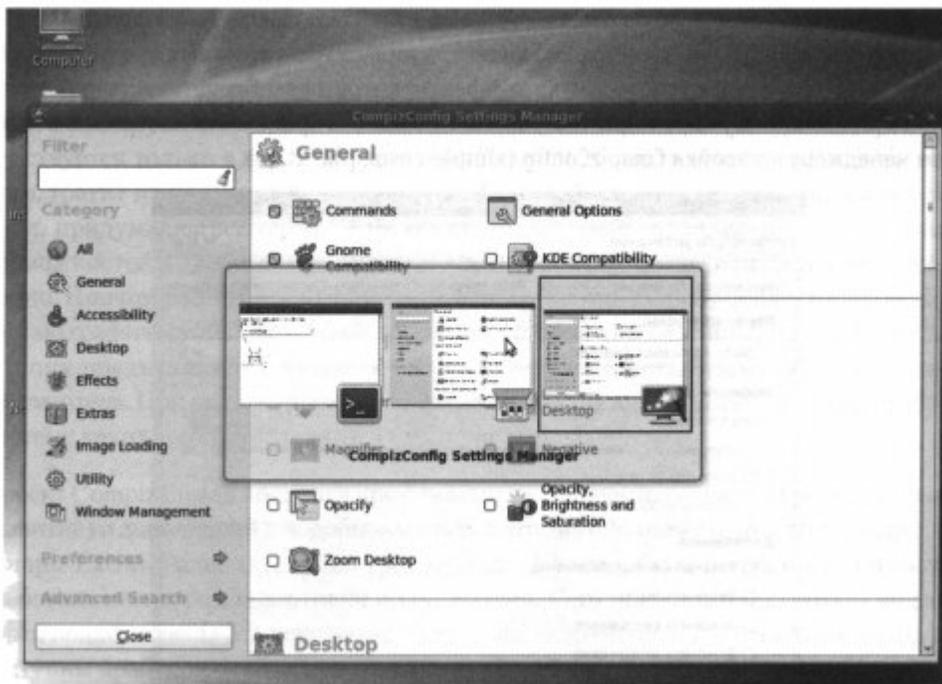


Рис. 3.20. Программа CompizConfig и переключатель приложений из CompizFusion

## Настраиваем клавиатуру

В случае выбора русского языка при установке дистрибутива на панели задач доступен индикатор, который показывает текущую раскладку. Реализуется такая функция за счет апплета GSwitchIt, который ранее развивался как отдельный проект, но теперь входит в состав рабочей среды GNOME. При щелчке правой кнопкой мыши на индикаторе появляется контекстное меню, пункты которого позволяют изменить раскладку, вызвать окно настройки программы либо просмотреть, как настроена текущая раскладка. По умолчанию комбинация клавиш, отвечающая за изменение раскладки, не назначена. Но это легко исправить. Выбираем меню **Параметры клавиатуры**, открывается окно настроек, состоящее из пяти вкладок: **Общие**, **Раскладки**, **Специальные возможности**, **Кнопки мыши** и **Перерыв в работе**. В поле внизу можно проверить текущую раскладку. Во вкладке **Общие** устанавливается периодичность автоповтора при постоянно нажатой клавише и мерцание курсора. Вкладка **Перерыв в работе** поможет делать периодические перерывы трудоголикам, блокируя клавиатуру на время. Раскладки редактируются на одноименной вкладке (рис. 3.21), по умолчанию в списке две комбинации — США и Россия; используя кнопки **Добавить** и **Удалить**, мы можем изменить этот список.

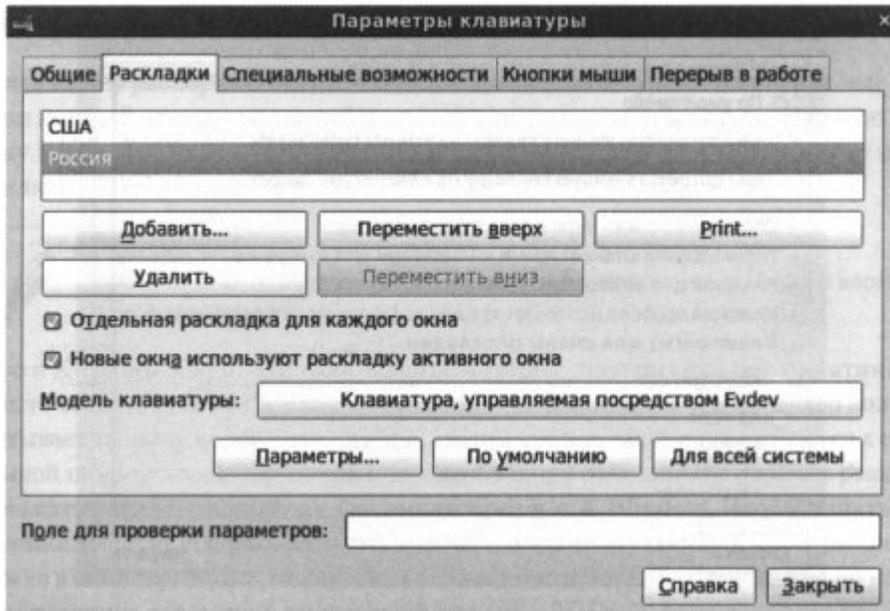


Рис. 3.21. Настройка раскладки

Обратите внимание на флажки **Отдельная раскладка для каждого окна** и **Новые окна используют раскладку активного окна**, установка/снятие которых позволяет определить глобальное изменение переключателя. Например, в терминале мы вводим команды, как правило, на английском, в текстовых документах — на русском, и если установлен флажок **Отдельная раскладка для каждого окна**, нам не понадобится менять раскладку при переходе между приложениями. Кнопка **Параметры** откроет доступ к многочисленным настройкам (рис. 3.22), позволяющим определить назначение некоторых клавиш (**Alt**, **Windows**, **CapsLock**, **Ctrl**) и их комбинаций — варианты раскладки цифровой клавиатуры, дополнительная подсветка раскладки при помощи индикаторов, ввод неразрывного пробела и др. Здесь же в подпункте **Клавиша(ы) для смены раскладки** устанавливаем нужную комбинацию (например, **Shift + Alt**). Обратите внимание, что там, где параметр установлен, он подсвечивается жирным шрифтом.

Современные клавиатуры, как правило, оснащены дополнительными клавишами, предназначенными для удобного управления мультимедийными устройствами. В Linux такие клавиши также работают, и в большинстве случаев дополнительных драйверов не потребуется. Возможно, после установки дистрибутива уже можно будет управлять мультимедийным проигрывателем без какой-либо дополнительной настройки. Если такая необходимость возникает, выбираем **Параметры • Комбинации клавиш клавиатуры** и в окне утилиты — **Комбинации клавиш клавиатуры** (gnome-keybinding-properties, рис. 3.23), в результате мы получаем возможность дополнительной настройки.

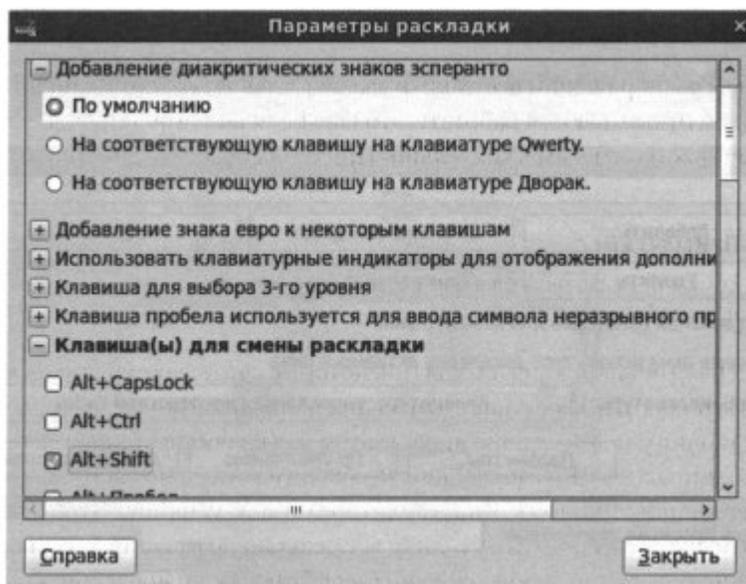


Рис. 3.22. Устанавливаем комбинацию клавиш для смены раскладки

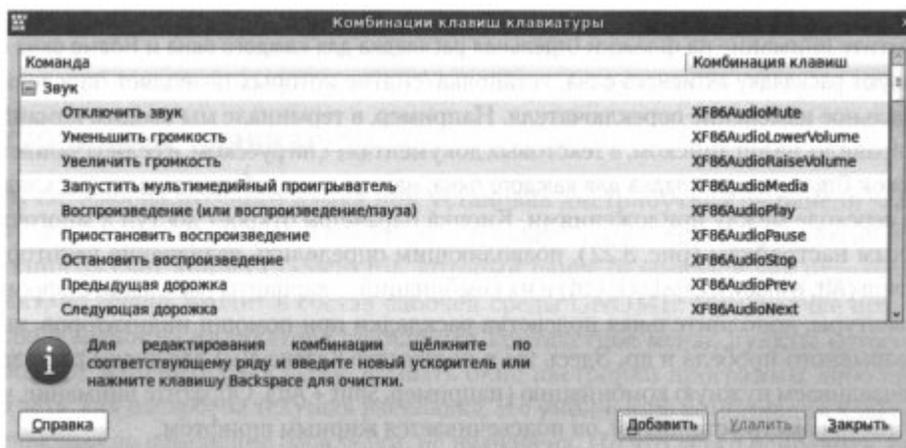


Рис. 2.23. Настройка мультимедийных клавиш

**ПРИМЕЧАНИЕ**

Все комбинации клавиш, начинающиеся с XF86, как раз и относятся к мультимедийным.

При необходимости можем очень просто добавить свои комбинации клавиатуры для выбранного действия.

## Если не помогло

Если в вашем распоряжении все-таки оказалась клавиатура, дополнительные клавиши которой отказываются работать, это еще не повод огорчаться. Нужную клавишу можно настроить самостоятельно. Причем особых трудностей это не представляет.



### ПРИМЕЧАНИЕ

Если материал покажется вам сложным, можете пропустить эту часть и вернуться при необходимости.

Ядро и X-сервер оперируют скан-кодами, которые считываются при нажатии клавиш, то есть, по сути, им все равно, что на них написано. Причем X-сервер вначале считывает таблицу кодов клавиш ядра, а затем код клавиши привязывается к собственной таблице кодов. Поэтому в Linux необходимо отдельно настраивать реакцию на нажатие кнопок в консоли (на эмуляторе) и в X Window. Чтобы узнать код клавиши, следует использовать утилиту `xev`, входящую в состав X-сервера. Запускаем ее в окне терминала, появится окно Event Tester, затем последовательно нажимаем клавиши, запоминая выдаваемый код (рис. 3.24):

```
$ xev
```

```
KeyRelease event, serial 31, synthetic NO, window 0x3e00001,
    root 0x67, subw 0x0, time 27 9734 67 6, (311, 611),
root:(1104,687),
    state 0x2000, keycode 236 (keysym 0x1008ff19, XF86Mail),
same_screen YES,
    XLookupString gives 0 bytes:
    XFilterEvent returns: False
```

И так далее.

Информации будет выдано много, так как будет отслеживаться каждое движение мыши при проходе над окном Event Tester. Клавишу описывает блок KeyRelease, в частности значения `keycode` как раз и являются скан-кодом, который мы хотим узнать. В приведенном примере нажата клавиша с кодом 236, которой соответствует код клавиши для X-сервера, указанный в `keysym`, и символьное обозначение клавиши действия XF86Mail (по умолчанию запускает почтовый клиент).



## Настройка привязки скан-кодов в X Window

Итак, скан-коды мы получили, теперь нужно указать X-серверу, что он должен делать при нажатии этой клавиши, то есть присвоить клавише символьное имя. Список символьных имен приведен в файле заголовков XF86keysym.h. По умолчанию заголовочные файлы X-сервера в современных дистрибутивах не устанавливаются, поэтому необходимо установить пакет `x11proto-core-dev`, после чего файл XF86keysym.h появится в каталоге `/usr/include/XI`. Или, как вариант, можно обратиться к GIT-серверу с исходными кодами на сайте [freedesktop.org](http://cgit.freedesktop.org/xorg/proto/x11proto/tree/XF86keysym.h) (<http://cgit.freedesktop.org/xorg/proto/x11proto/tree/XF86keysym.h>). Смотрим:

```
$ cat /usr/include/X11/XF86keysym.h

#define XF86XK_Back      0x1008FF26 /* Like back on a browser */
#define XF86XK_Forward  0x1008FF27 /* Like forward on a browser */

#define XF86XK_Mail     0x1008FF19
```

И так далее. Если сравнить последнюю строку с выводом `hex`, то увидим, что значения совпадают с клавишей с `keycode 236` — `keysym 0x1008ff19`, XF86Mail (без суффикса XK\_). Список всех доступных значений в том виде, каком они должны использоваться, вы найдете в файле `/usr/share/X11/XKeysymDB`.

Составить свой вариант раскладки можно несколькими способами: создать описание своей клавиатуры или использовать `Xmodmap`. Последний способ самый простой, поэтому о нем и будем говорить дальше. В домашнем каталоге пользователя создаем файл `~/.Xmodmap` или используем общесистемный `/etc/X11/Xmodmap`, в который и заносим все нужные значения.

```
$ gedit      Xmodmap

keycode 161 XF86Calculator
keycode 174 XF86AudioLowerVolume
keycode 176 XF86AudioRaiseVolume
keycode 162 XF86AudioPause
```



### ПРИМЕЧАНИЕ

Код клавиш можно заносить как в десятичном, так и шестнадцатеричном виде.

Как вы понимаете, теперь мы можем фактически построить свою раскладку клавиатуры. Коды большинства клавиш стандартизированы, поэтому достаточно настроить реакцию на нажатие клавиши один раз, а в случае переустановки перенести файл в новую систему.

**ПРИМЕЧАНИЕ**

Хотя использование `~/Xmodmap` и `/etc/X11/Xmodmap` является почти стандартным, в некоторых дистрибутивах Linux могут быть свои особенности. Найти, откуда запускается нужная команда, можно, проанализировав вывод команды `sudo grep -iR xmodmap /etc`.

Теперь, когда символьные имена клавишам присвоены, можно назначать им любые желаемые действия.

## Настройка мультимедийных клавиш в консоли

В консоли порядок настройки клавиш несколько отличается от графического режима. Главная особенность — клавишных команд в консоли не может быть больше 128, то есть следует выбирать значения от 0 до 127:

```
$ setkeycodes 0hex 118
```

Посмотреть свободные значения можно в файле текущей клавиатурной раскладки. По умолчанию в Linux Mint загружается файл `/etc/console-setup/cached.kmap.gz`.

**ПРИМЕЧАНИЕ**

В других дистрибутивах, базирующихся на Debian, это обычно `/etc/console-setup/boottime.kmap.gz`.

Скорее всего, будут заняты все клавиши, но часто не все из них действительно задействованы. Если после запуска проблем с клавишами нет, то заносим эту строку в один из стартовых скриптов, например в `/etc/init.d/rc.local`.

Теперь осталось задать соответствие клавиши и выполняемого действия. Здесь уже простора для творчества даже больше, чем в X. В `man keymaps (5)` процедура установки соответствия `keycode` выглядит так:

```
{ plain | <modifier sequence> } keycode keynumber = keysym
```

Иными словами, можно указать так:

```
keycode 105 = Decr_Console
```

Таким образом, мы реализовали переключение консоли на одну назад при нажатии клавиши с кодом 105.

Теперь зададим переключение консоли на одну вперед при нажатии `Alt` и клавиши с кодом 106:

```
alt keycode 106 = Incr_Console
```

Но можно создавать и свои варианты, указывая команду в переменных:

```
keycode 120 = F100
```

```
string F100 = "/sbin/shutdown -h now\n"
```

Итак, при нажатии клавиши с кодом 120 будет выполнено действие, указанное в переменной F100, в нашем случае задано выключение компьютера. Вместо F100, естественно, можно использовать другое имя.

Чтобы не прописывать эти параметры непосредственно в файл `/etc/console-setup/ cached.kmap.gz`, лучше прописать нужную строку в `/etc/init/console-setup.conf`.

## Клавиши на ноутбуке

Сегодня ноутбуки очень популярны у пользователей, которых привлекают мобильность, энергосберегающие функции, а также то, что они практически не занимают места на рабочем столе. Если дополнительные клавиши не работают, то принцип добавления событий аналогичен описанному выше. Хотя есть один прием, о котором хотелось бы сказать, — это режим гибернации (hibernate) и реакция системы на закрывание крышки.



### ПРИМЕЧАНИЕ

В режиме гибернации сохраняется состояние компьютера на момент выключения системы, после перезагрузки Рабочий стол и все документы, с которыми работал пользователь, восстанавливаются в том же положении. Но его активация требует, чтобы swap-раздел был чуть больше, чем объем оперативной памяти, установленной на компьютере.

Современные дистрибутивы поддерживают режим гибернации, в Linux Mint для этого достаточно при выключении компьютера выбрать пункт **Спящий режим**. Единственное, что каждый раз для перехода в этот режим необходимо идти в меню, а хочется просто закрыть крышку ноутбука и, вновь включив питание, обнаружить все на своих местах. Это сделать очень просто.



### ПРИМЕЧАНИЕ

Перейти в режим гибернации можно, введя команду `sudo /etc/acpi/hibernate.sh`.

Любыми событиями ACPI управляет демон `acpid`, доступ ко всем событиям можно получить через `/proc/acpi/event`.



### ПРИМЕЧАНИЕ

Сервисы в Linux называют демонами (англ. *daemons*), под которыми подразумевают программы, работающие автономно с минимальным участием пользователя.

В каталоге `/etc/acpi/events/` содержится набор конфигурационных файлов, в том числе описывающих некоторые модели ноутбуков. В каталоге `/etc/acpi` находятся скрипты, описывающие действия при нажатии определенных клавиш.



### ПРИМЕЧАНИЕ

Список ноутбуков, которые протестированы в Linux Mint, можно найти по адресу [http://linuxmint.com/wiki/index.php/Linux\\_Mint\\_friendly\\_Laptops](http://linuxmint.com/wiki/index.php/Linux_Mint_friendly_Laptops).

Например, при нажатии клавиши включения питания обрабатываются события, описанные в файле `/etc/acpi/powerbtn.sh`: вначале проверяются установки, указанные

в **Параметрах управления питанием** (рис. 3.25), которые вызываются из меню **Параметры • Управление питанием**, и если событие не определено (например, используется другой менеджер рабочего стола), то вызывается команда

```
/sbin/shutdown -h now "Power button pressed"
```

которая выключает компьютер. Обладая даже начальными знаниями в программировании в командной оболочке, можно создать любую реакцию, например нажатие кнопки выключения или любой другой кнопки.



#### ПРИМЕЧАНИЕ

Научиться программированию в командной оболочке Linux можно по инструкциям в **Bash Reference Manual** (<http://www.gnu.org/software/bash/manual/bashref.html>) или **Advanced Bash-Scripting Guide** (<http://www.tldp.org/LDP/abs/abs-guide.pdf>), русский перевод — [http://www.opennet.ru/docs/RUS/bash\\_scripting\\_guide/](http://www.opennet.ru/docs/RUS/bash_scripting_guide/).

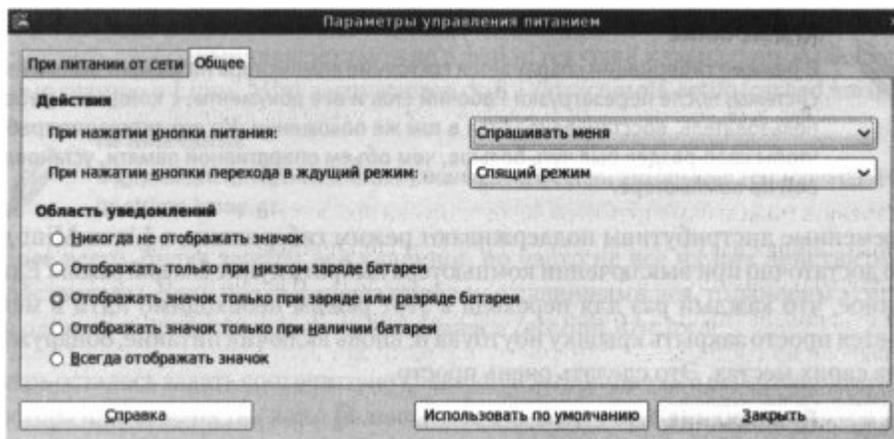


Рис. 3.25. Управление питанием в Linux Mint

Реакция на закрытие крышки ноутбука описывается в файле `cat /etc/acpi/lid.sh`. После изменений потребуется перезапуск демона `acpid`:

```
$ sudo /etc/init.d/acpid restart
```

## Автозапуск приложений

Для удобства пользователя некоторые приложения можно добавить в автозагрузку, тогда они будут стартовать вместе с системой и не нужно искать их в меню. Здесь Linux Mint, как, впрочем, и любой другой дистрибутив Linux, предлагает несколько способов.

Самый простой из них — использовать **Центр управления**, в котором выполнить команду меню **Персональные • Запускаемые приложения** и в появившемся окне **Параметры запускаемых приложений** выбрать необходимые (рис. 3.26).

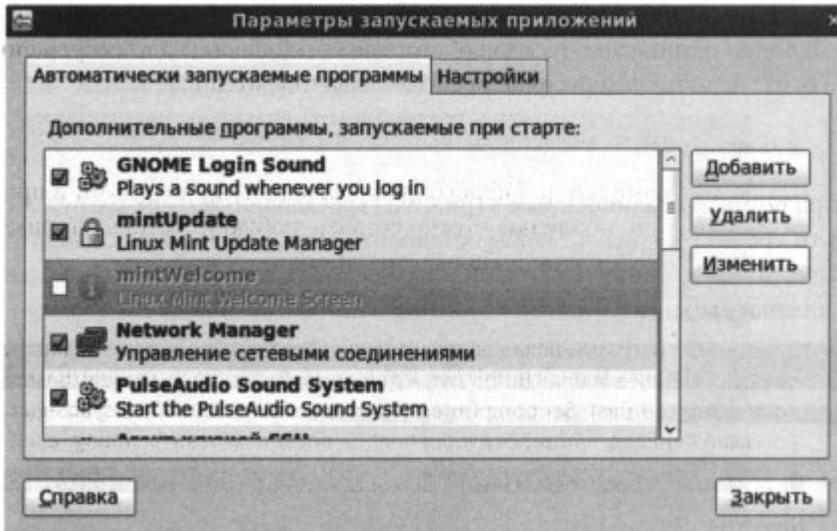


Рис. 3.26. Редактируем список приложений, загружающихся автоматически

Чтобы отключить приложение из автозагрузки, достаточно снять флажок, при помощи **Изменить** можем изменить параметры запуска приложения. Чтобы добавить новое приложение, нажимаем одноименную кнопку и заполняем предложенные программы. Все просто и понятно. Обратите внимание на вкладку **Настройки**, содержащую всего один флажок, активация которого разрешит запоминать работающие приложения при выходе из системы. В итоге после перезагрузки пользователь получит **Рабочий стол** таким же, как и при выключении компьютера. Нажатие кнопки **Запомнить работающие приложения** позволит сохранить список работающих приложений. Эти настройки относятся исключительно к **Рабочему столу** GNOME. Кроме этого при загрузке системы стартуют сервисы, которые управляются совсем в другом месте.

## Настройка сервисов

К сожалению, графического инструмента для редактирования списка сервисов разработчики не предлагают. Но, очевидно, это связано с тем, что пользователь должен знать, что действительно необходимо отключать, иначе, убрав какой-то нужный сервис, можно получить неработающую систему. В репозитории дистрибутива доступно несколько инструментов, среди них — утилиты с псевдографическим интерфейсом **rcconf** и **sys-rc-conf**.

Чтобы установить их, введите команду

```
$ sudo apt-get install rcconf
```

```
$ sudo apt-get install sys-rc-conf
```

Как вариант, можно воспользоваться графическими утилитами установки приложений. В меню ссылки на указанные утилиты не добавляются, поэтому проще запустить их, используя терминал:

```
$ sudo sys-rc-conf
```

Далее при помощи клавиши пробел (рис. 3.27) устанавливаем автозагрузку сервиса на пяти уровнях загрузки. Чтобы остановить сервис, нажимаем клавишу - (минус), запустить — клавишу + или =. Перемещение производится при помощи стрелок на клавиатуре.

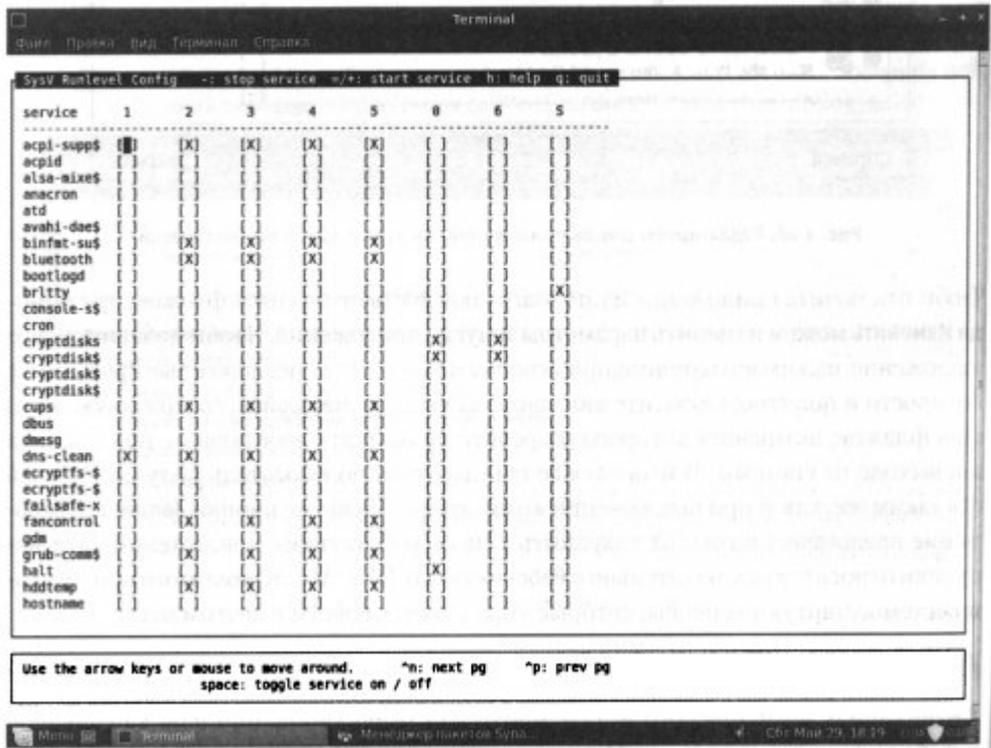


Рис. 3.27. Настройка автозагрузки сервисов при помощи утилиты sysv-rc-conf

Чтобы закончить работу с программой, нажимаем клавишу q.

Утилита rcconf также запускается в терминале.

```
$ sudo rcconf
```

позволяет редактировать автозагрузку на текущем уровне запуска операционной системы (рис. 3.28).



или, как вариант, вместо `restart` используем `force-reload` для быстрой перезагрузки сервиса.



#### ПРИМЕЧАНИЕ

Если запустить скрипт без дополнительных указаний, он выдаст допустимые параметры, обычно это `start`, `stop`, `restart` и `force-reload`.

Если посмотреть список файлов в `/etc/init.d`, то мы увидим, что их достаточно много. Удалить оттуда файл, значит, фактически лишить пользователя возможности загрузить сервис вручную. Возникает резонный вопрос: они все загружаются? Нет. На этапе развития Unix-систем пришли к простой схеме: в каталоге `/etc/init.d` размещаются скрипты, а в отдельном каталоге или файле указывается список тех, которые необходимо запускать. В Linux Mint используются именно каталоги. В `/etc` находятся каталоги с именем, начинающимся с `rc` — `/etc/rc.local`, `/etc/rc0.d`, `/etc/rc1.d`, `/etc/rc2.d`, `/etc/rc3.d`, `/etc/rc4.d`, `/etc/rc5.d`, `/etc/rc6.d` и `/etc/rcS.d`. Если просмотреть список файлов внутри, то увидим, что все они являются символическими ссылками на файл в `/etc/init.d`. Приведу для примера часть файлов, размещенных в `/etc/rc5.d/`. Для этого воспользуемся консольной командой `ls`.

```
$ ls -al /etc/rc2.d/

lrwxrwxrwx  1 root root    20 2010-05-22 13:14 K20fancontrol
-> ../init.d/fancontrol

lrwxrwxrwx  1 root root    17 2010-05-22 13:14 S20hddtemp
-> ../init.d/hddtemp

lrwxrwxrwx  1 root root    19 2010-05-22 13:14 S25bluetooth
-> ../init.d/bluetooth

lrwxrwxrwx  1 root root    18 2010-05-22 13:14 S99rc.local
-> ../init.d/rc.local
```

Как видите, файл `/etc/rc5.d/K20fancontrol` на самом деле является символической ссылкой на файл `/etc/init.d/fancontrol`.



#### ПРИМЕЧАНИЕ

В консоли можно указывать абсолютное имя файла как относительно корня дерева, так и относительно текущего положения. При этом `(.)` точка означает текущий каталог, а две точки `(..)` — каталог уровнем выше.

Теперь разберем, что мы получили. Из каталога `/etc/rcS.d` запускаются базовые системные компоненты, их лучше не трогать. Чуть выше было сказано об уровнях загрузки (runlevels) Linux, которые позволяют установить сервисы в зависимости от выбранного варианта использования системы. Всего пять уровней, рассмотрим их.

1. **Однопользовательский режим** — в системе не запускаются дополнительные службы, как правило, доступна одна консоль, отсутствуют сетевые подключения. Применяется обычно при аварийно-восстановительных работах.
2. **Многопользовательский режим с минимально необходимыми сервисами** — доступ к сети отсутствует, применяется при первичной настройке серверов в целях повышения безопасности.
3. **Многопользовательский режим** — работают все предопределенные сервисы, сеть доступна, в таком режиме загружаются все сервера.
4. **Не используется/не определен** — администратор может самостоятельно его настроить.
5. **Многопользовательский режим с использованием графической подсистемы** — именно в нем работают по умолчанию все дистрибутивы, ориентированные на работу в качестве настольной системы.

Кроме этого определены режимы: 0 — остановка системы и 6 — перезагрузка. Их также не следует трогать, поскольку в этих режимах выполняются операции по корректному завершению работы системы (размонтирование ресурсов, остановка служб и т. д.). Так говорит теория, но на практике из правил, как обычно, имеются исключения. Если просмотреть ссылки в Linux Mint, то получится, что разницы между уровнями 2-5 нет. Команда **runlevel**, введенная в Linux Mint, выводит уровень 2 загрузки:

```
$ runlevel
```



#### ПРИМЕЧАНИЕ

Как вариант, для просмотра уровня загрузки можно использовать команду **who -g**.

В дальнейшем будем считать, что Linux Mint загружается на уровне 2, поэтому нас интересует содержимое каталога **/etc/rc2.d/**. После установки системы в нем найдем более десятка ссылок, имя которых будет начинаться с буквы S или K, затем следует цифра и название программы. Цифра показывает порядок загрузки демонов. Это сделано для того, чтобы удовлетворить зависимости. Например, некоторый сервис требует, чтобы вначале была активирована сеть: если его запустить, перед тем как будут подняты сетевые интерфейсы, он завершится с ошибкой. Чем меньше цифра, тем раньше загрузится демон. Если вы добавляете свой сервис и не уверены в том, что знаете порядок, установите просто самую максимальную цифру — 99. Кстати, под 99 идет ссылка на файл **/etc/init.d/rc.local**. Он вместе с **/etc/rc.local** предназначен для запуска сервисов, определенных самим пользователем. Просто открываем файл в редакторе и вписываем команду для запуска, и нужная программа или сервис

будут загружены вместе с системой. Самостоятельно создать ссылку при помощи команды в терминале очень просто:

```
$ ln -s /etc/init.d/samba /etc/rc2.d/S20samba
```

Здесь S (start) показывает, что демон будет загружаться, а K (Kill) — не будет работать. Иными словами, чтобы временно приостановить запуск демона при загрузке систем, мы можем не удалять символическую ссылку, а просто заменить букву S на K:

```
$ sudo mv /etc/rc2.d/S20hddtemp /etc/rc5.d/K20hddtemp
```

Теперь сервис, отслеживающий температуру жестких дисков, не будет стартовать при загрузке системы. Отключая ненужные сервисы, мы можем экономить системные ресурсы, чуть разгрузив систему.

Вручную создавать и удалять ссылки необязательно. В комплекте Linux Mint доступна консольная утилита `update-rc.d`, которая позволяет создавать все необходимые ссылки и контролирует очередность загрузки сервисов. Все ее параметры доступны в справочной системе `man`, чтобы получить доступ к ней, следует ввести в консоли

```
$ man update-rc.d
```

Параметры запуска просты. Чтобы удалить программу `program` из автозагрузки, используем

```
$ sudo update-rc.d program remove
```

Если необходимо просто посмотреть, что будет делать утилита без выполнения собственно операции удаления, добавляем ключ `-n`. Ключ `-f` позволит принудительно удалить ссылки в любом случае. Для активации сервиса на всех уровнях используем параметр `defaults`.

```
$ sudo update-rc.d program defaults
```

Можно указать и действия на конкретных уровнях:

```
$ sudo update-rc.d program start 20 2 3 4 5 . stop 20 0 1 6 .
```

Теперь программа будет стартовать на уровнях 2-5 и отключена на уровнях 0,1 и 6. Цифра 20 на первой позиции указывает число, которое будет стоять после S или K, то есть S20 или K20.



#### ПРИМЕЧАНИЕ

Утилита `Insservls` используется для обновления списка символических ссылок на основе LSB-сценариев загрузки, чтобы вначале запускались зависимые сервисы. Она доступна после установки Linux Mint, просто запустите ее в терминале.

## Продвинутые настройки

Как я говорил в самом начале, основная идеология GNOME — спрятать все лишнее от пользователя, предоставив ему готовую среду. За это его любят и за это недолюбливают те пользователи, которые видят свое рабочее окружение по-другому. Все настройки рабочей среды хранятся в файлах XML-стандарта (от англ. extensible Markup Language — «расширяемый язык разметки»), системных и размещенных в рабочем каталоге пользователя. При загрузке параметров данные считываются, и если они не определены или файл отсутствует, то используются установки по умолчанию. Настроек много, все охватить не получится, в книге будет описан сам принцип и даны некоторые популярные установки.



### ПРИМЕЧАНИЕ

За более подробной инструкцией по настройке GNOME обратитесь к руководству администратора, размещенному по адресу <http://library.gnome.org/admin/system-admin-guide/stable/>.

Начнем с настроек Gtk. По умолчанию установки рабочей среды не оптимальны, но их легко переопределить, самостоятельно отредактировав (или создав) файл `~/gtkrc-2.0`. Рассмотрим некоторые из них.

```
$ gedit ~/gtkrc-2.0

# задержка, при которой появляется меню при наведении указате-
ля мышки, по умолчанию 100 мс

gtk-menu-popup-delay = 0

# задержка в миллисекундах, появляющаяся перед скрытием подменю,
когда указатель перемещается на другое подменю

gtk-menu-popdown-delay = 0

# задержка перед появлением подменю

    gtk-menu-bar-delay=0
```

Описание остальных параметров Gtk можно найти в официальной документации проекта <http://library.gnome.org/devel/gtk/2.20/GtkSettings.html>. Для ускорения запуска программ часто рекомендуют кэшировать значки в памяти. Организовать это просто:

```
$ echo "find /usr/share/pixmaps/ | xargs cat > /dev/null" >> ~/
mystart

$ echo "find /usr/share/icons/shiki-wise/ | xargs cat > /dev/
null" >> ~/mystart
```

Делаем файл исполняемым:

```
$ chmod +x ~/mystart
```

Теперь добавляем его в автозагрузку, как показано выше: **Центр управления • Персональные • Запускаемые приложения** и **Параметры запускаемых приложений**, нажимаем кнопку **Добавить** и вписываем путь к скрипту. Хотя в зависимости от используемой темы кэширование может не работать.

## Редактор GConf

Большая часть настроек доступна посредством редактора GConf, который упрощает администрирование среды GNOME. GConf состоит из наборов установок, демона gconfd-2, отслеживающего все изменения в конфигурационных файлах, и командной утилиты gconftool-2. Если сравнить с Windows, то он представляет собой реестр, в котором все данные хранятся в XML-файлах. Например, многие пользователи устанавливают утилиты, автоматически переключающие раскладку клавиатуры, в итоге на панели показываются два индикатора раскладки. Чтобы убрать индикатор GNOME, достаточно ввести команду

```
$ gconftool-2 -s /desktop/gnome/peripherals/keyboard/general/disable_indicator -t bool true
```

Чтобы прочитать предыдущее значение параметра GNOME или проверить произведенную установку, используем опцию `—get`. Например,

```
$ gconftool-2 --get /apps/metacity/general/reduced_resources
```

Теперь его можно активировать:

```
$ gconftool-2 --type bool --set /apps/metacity/general/reduced_resources true
```

В результате стандартный оконный менеджер рабочей среды GNOME — Metacity уменьшит потребление ресурсов за счет отключения лишней анимации. Например, при перемещении окон не будет показываться содержимое. Анимацию при включении и выключении панелей отключаем установкой параметра `enable_animations` в `false`.



### ПРИМЕЧАНИЕ

Все настройки GConf, произведенные пользователем, сохраняются в XML-файлах, размещенных в каталогах `~/gconf` и `~/gconfd`, общесистемные настройки — `/etc/gconf`.

```
$ gconftool-2 --type bool --set /apps/panel/global/enable_animations false
```

Если анимация панелей активна, то ее еще можно скорректировать, установив скорость анимации **Быстрая**:

```
$ gconftool-2 --type string --set /apps/panel/global/panel_animation_speed panel-speed-fast
```

В GNOME, если в свойствах среды выставлен режим перевода окна открытого приложения на передний план при наведении на него указателя мыши, можно установить свой вариант задержки. Например, 100 мс:

```
$ gconftool-2 --type integer --set /apps/metacity/general/auto_raise_delay 100
```

Параметров очень много, и все их запомнить просто невозможно, но в этом нет необходимости. В настоящее время доступно несколько редакторов, позволяющих произвести все необходимые настройки GConf напрямую в графической среде. В комплекте Linux Mint также доступна утилита **Редактор конфигураций GConf-editor** (рис. 3.29). В меню нет ярлыка для его запуска, поэтому открываем консоль и вводим команду:

```
$ gconf-editor
```

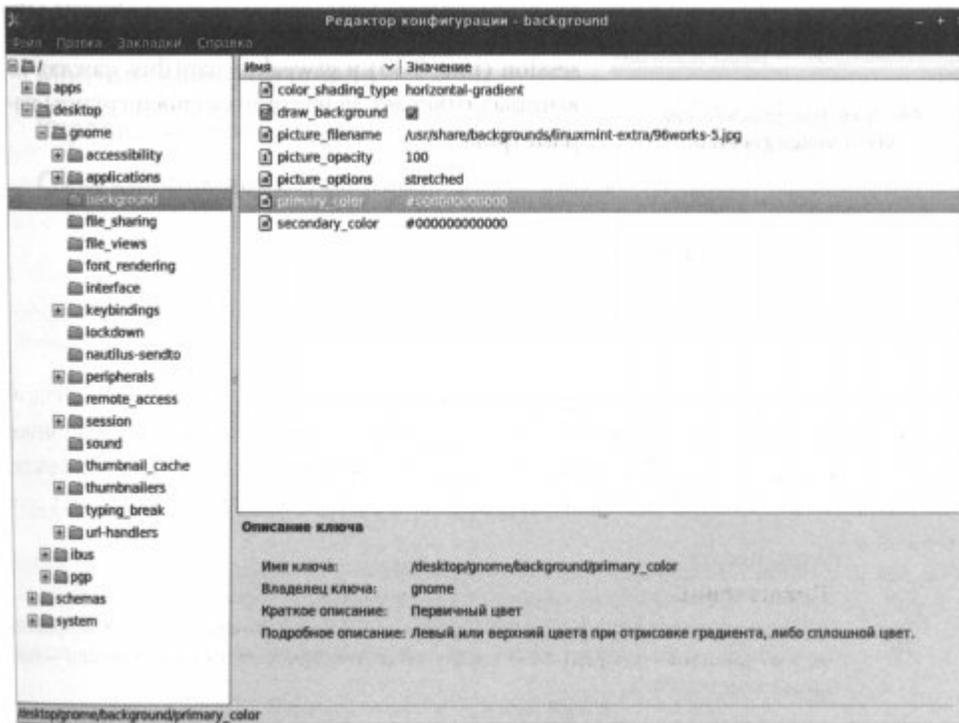
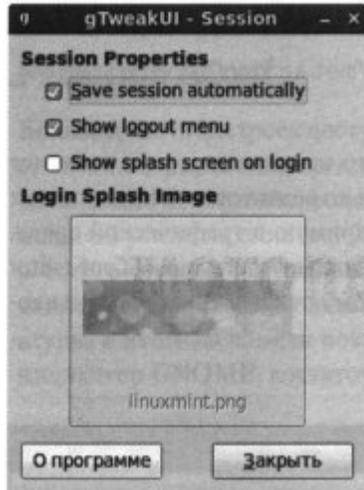


Рис. 3.29. Редактор конфигурации GNOME — GConf-editor

Окно **Редактора конфигураций** можно назвать стандартным для такого рода программ, он в чем-то похож на редактор реестра **regedit** из Windows. Слева размещена ветка параметров, при выборе любого из них его установки появляются в правой части окна. Внизу выводится краткая характеристика на русском языке. Конечно, некоторое время придется потратить на то, чтобы найти все нужные установки, ведь их очень много. Например, значки **Компьютер**, **Домашний каталог пользователя** и др., которые мы устанавливали при помощи **Desktop Settings**, активируются в ветке

**Apps • Nautilus • Desktop**. Прочие настройки **Рабочего стола** размещены в отдельной ветке **Desktop**.



**Рис. 3.30.** Настройка сессий при помощи gTweakUI

Для быстрого и простого тюнинга **Рабочего стола** можно рекомендовать набор утилит **gTweakUI** (<http://gtweakui.sourceforge.net>), которые позволяют активировать нужную функцию установкой флажка. Пакет с **gTweakUI** доступен в репозитории дистрибутива, поэтому установить его можно при помощи **Менеджера пакетов** или командой

```
$ sudo apt-get install gtweakui
```

После этого в системе появятся четыре новые утилиты: **gtweakui-galeon**, **gtweakui-menus**, **gtweakui-session** (рис. 3.30) и **gtweakui-nautilus**, каждая из которых отвечает за настройку своей группы параметров.

# Глава 4

## Linux Mint под капотом

Файлы и каталоги

Права доступа

Монтирование разделов

Управление процессами

Сбор информации о системе

Итак, первое знакомство с системой позади, самое время разобрать некоторые моменты внутреннего устройства Linux, чтобы понять, как все работает.

## Файлы и каталоги

Одна из главных концепций Unix-систем говорит, что все является файлом, в том числе и любое устройство, подключенное к компьютеру. Каталог, содержащий файлы, также является файлом. Исходя из этого однозначно сказать, чем является `/home/grinder/temp` — файлом, каталогом или ссылкой, — нельзя. Чтобы избежать двусмысленности, иногда каталог отмечают еще одной косой чертой `/home/grinder/temp/`. Но несколько подробнее об этом мы поговорим чуть дальше.

Литературы для тех, кто хочет разобраться более глубоко в этом вопросе, предостаточно, поэтому затронем лишь основные моменты. Файл — одно из самых важных понятий Linux. Файлом в этой системе является все.

### ВНИМАНИЕ



**В отличие от Windows в Linux регистр в имени файла имеет значение, поэтому `file` и `File` — это два разных файла.**

Любое устройство, подключенное к системе, информация, находящаяся на жестком диске, обязательно представлена файлом. Но, хотя для работы со всеми файлами используются обычные команды, внутри файлы совершенно разные, и текстовый файл существенно отличается от файла, представляющего устройство. Каталог также является файлом и содержит ссылки на другие файлы и каталоги. Чтобы убедиться в этом, достаточно посмотреть «внутри» при помощи любого шестнадцатеричного редактора.

### ВНИМАНИЕ!



**В имени файла можно использовать символы любого языка, в том числе и русского. Но от использования некоторых символов, применяемых командной оболочкой, лучше удержаться: `*`, `\`, `&`, `<`, `>`, `;`, `(`, `)`, `|`, знаков пробела и табуляции.**

Именно каталог определяет местонахождение файла в дереве файловой системы, так как файл не содержит информации о своем положении. Если файл не описан ни в одном из каталогов, он потерян, и получить к нему доступ нельзя.

### ПРИМЕЧАНИЕ



**Удаление файла фактически означает его «исключение» из списков каталога и освобождение связанных с ним метаданных. Сам файл остается на жестком диске компьютера, пока не будет затерт другими данными. Но некоторое время его еще можно восстановить.**

Помимо информации о расположении файла каталог содержит ссылки на метаданные, позволяющие производить манипуляции с этими файлами. Еще одной

особенностью программ в Linux, с которой придется смириться новичку, является отсутствие расширений в файлах, как это принято в Microsoft Windows. Нет, при работе некоторых программ, рассчитанных на конечного пользователя, вроде офисного пакета OpenOffice.org, создаются файлы, имеющие привычные расширения, но это можно считать скорее данью современным тенденциям, чем правилом.

**ВНИМАНИЕ!**

Если в дальнейшем при описании команд, вводимых с консоли, будет использоваться знак доллара (\$), значит, для выполнения этой команды достаточно прав обычного пользователя, а если решетка (#) — нужны права root.

Наличие расширения позволяет пользователю и программе быстро определить, какого типа данные находятся внутри файла. Но, как и в Windows, наличие расширения — это только соглашение об имени файла, но совсем необязательно, что внутри находится именно этот тип данных. Многие приложения, например файловые менеджеры Nautilus или Konqueror, при анализе файлов в открытом каталоге определяют тип содержимого файла на основании самих данных. Это легко проверить.

**ВНИМАНИЕ!**

В дистрибутиве Linux Mint, чтобы пользователь мог использовать sudo, он должен входить в группу admin. Первый созданный при установке пользователь автоматически в нее заносится.

Уберите расширение у файлов, и вы увидите, что значок, показывающий тип файла, не изменится (рис. 4.1), а при щелчке кнопкой мыши на нем будет запускаться нужное приложение.

Тип содержимого любого файла можно проверить при помощи утилиты, которая так и называется — file.

```
$ file /
```

```
/: directory
```

```
$ file /bin/bash
```

```
/bin/bash: ELF 32-bit LSB executable, Intel 80386, version 1  
(SYSV), for GNU/Linux 2.2.0, dynamically linked (uses shared  
libs), for GNU/Linux 2.2.0, stripped
```

Кроме команды file узнать тип файла можно при помощи команды `ls -l` или `stat` (рис. 4.2). Вывод `ls` покажет такую информацию.

```
$ ls -l /dev/
```

```
crw-rw      1 root audio 14,  4 2008-05-01 10:55 audio
```

```
drwxr-xr-x  3 root root  60 2008-05-01 13:54 bus
```

```
lrwxrwxrwx 1 root root 3 2008-05-01 13:55 cdrom -> hdc
brw-rw---- 1 root disk 3, 0 2008-05-01 13:54 had
```



Рис. 4.1. Вывод файлов по типу содержимого в Nautilus

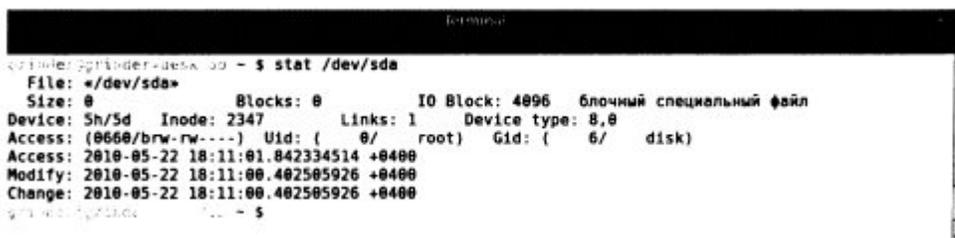


Рис. 4.2. Вывод утилиты stat

Первая буква как раз и показывает тип файла. Это может быть:

- - (тире) — обычный файл с пользовательскими данными (текстовый, графический, мультимедийный или исполняемый файл, файл базы данных и др.), для операционной системы такие файлы — просто последовательность байт;
- **d** — каталог (англ. directory);

- l — символическая ссылка (англ. link);
- b — файл блочного устройства (англ. block);
- c — файл символьного устройства (англ. character);
- f — файл FIFO или именованный канал;
- s — сокет (англ. socket).

С обычными файлами и каталогами все ясно. Разберем особенности остальных типов файлов.

В операционной системе Microsoft Windows существует понятие ярлыка, когда на любой файл можно сделать связанную ссылку. В Linux тоже есть такая возможность. Выглядит она следующим образом. Некоторые метаданные и имя файла хранятся в каталоге, то есть получается, что файл существует вне зависимости от того, как его называют, это позволяет одному файлу иметь несколько имен. Файловые системы Linux (как и в Windows, начиная с Windows 2000) поддерживают два типа ссылок. Символическая ссылка является неким аналогом ярлыка (по назначению, но не структуре). Внутри такого файла содержится ссылка на другой файл. При удалении основного файла ссылка перестает работать, так как показывает на несуществующий объект. Символические ссылки используются для более удобной организации файлов. Например, в некоторых дистрибутивах разработчики не полностью придерживаются FHS, но при помощи символических ссылок имитируют нужную структуру. Другой вариант — имитация более ранних версий библиотек.



#### ПРИМЕЧАНИЕ

Можно создавать символическую ссылку не только на файл, но и на каталог.

Некоторые программы жестко требуют определенную версию библиотек, в новых все функции поддерживаются для обратной совместимости, поэтому вместо установки двух вариантов используют один, но с разным именем. Символические ссылки в команде `ls -l` обозначаются буквой `l`. Кроме того, при помощи стрелки указывается оригинальный файл, на который делается эта ссылка (см. выше вывод команды по тексту). В отличие от жестких ссылок символические можно создавать на файл, который не существует. Расположение файлов значения не имеет. Создаются символические ссылки при помощи команды `ln` с ключом `s`. Например, создадим символическую ссылку с именем **symlinks** на файл **file**:

```
$ ln -s file /home/grinder/symlink
```

Проверяем:

```
$ ls -l /home/grinder/symlink
```

```
lrwxrwxrwx 1 grinder grinder 7 2010-05-01 20:58 /home/grinder/  
symlink -> file
```

Поскольку символическая ссылка — это не файл, а путь к файлу, размер файла символической ссылки обычно небольшой, в нашем примере всего 7 байт. Права доступа созданной ссылки не содержат ограничений (о правах доступа далее в главе), что видно из строки `gwxgwxgwx`. Возможность работы с файлом, на который указывает ссылка, будет определяться правами доступа самого файла. Поэтому нельзя создать символическую ссылку на защищенный файл и получить доступ к нему.



#### ПРИМЕЧАНИЕ

Обычно вывод команды `du` показывает нулевой размер файла символической ссылки. Чтобы увидеть истинный размер такого файла, следует использовать ключ `-b`, показывающий размер в байтовом эквиваленте.

Жесткие ссылки, в отличие от символических, полностью равноправны, так как фактически это два имени одного и того же файла. Информация о количестве ссылок содержится непосредственно в метаданных файла. При удалении одной из ссылок остальные полностью работоспособны. Пока имеется хоть одна ссылка, файл будет существовать. Однако жесткая ссылка имеет единственное ограничение — ее создание возможно только в пределах одного логического раздела диска. Создаются ссылки при помощи команды `ln`, без аргументов:

```
$ ln file /home/grinder/hardlink
```

Количество жестких ссылок на файл можно увидеть во втором столбце вывода команды `ls`:

```
$ ls -l /home/grinder/hardlink
```

```
-rw-r--r-- 2 grinder grinder 52219904 2010-05-22 14:33 /home/  
grinder/hardlink
```

В нашем примере показано количество ссылок 2. Размер жесткой ссылки совпадает с размером исходного файла, но физически файл только один, поэтому, вне зависимости от количества ссылок на него, он занимает на диске ровно столько, сколько и исходный. Права доступа созданной ссылки полностью повторяют права оригинального файла.

В Linux жесткие ссылки не редкость. Часто разработчики дают два имени своей программе — в зависимости от того, под каким именем она запущена, она будет иметь различную функциональность. Узнать такие программы-близнецы можно, только проанализировав номер `inode` при помощи ключа `i` команды `ls`. Например, команда `gzip` предназначена для сжатия и распаковки файлов наподобие ZIP, только с использованием кодирования Лемпеля — Зива (LZ77). В Linux есть отдельная утилита для распаковки, смотрим на номера `inode`:

```
$ ls -i /bin/gunzip /bin/gzip
```

```
25256 /bin/gunzip 25256 /bin/gzip.
```

Как видите, номер inode у этих двух файлов совпадает, то есть на самом деле это один файл. Вывод команды `ls -l` показывает наличие четырех жестких ссылок на файл `/bin/gzip`. Давайте найдем остальные, используя в качестве фильтра вывода команду `grep` с номером нужного inode.

```
$ ls -li /bin/* | grep 25256

25256 /bin/gunzip

25256 /bin/gzip

25256 /bin/uncompress

25256 /bin/zcat
```

Как видите, четыре команды являются на самом деле одним файлом, но в зависимости от имени, с которым этот файл запущен, он выполняет разные задачи.

Надеюсь, со ссылками все ясно. Теперь переходим к файлам устройств. Хотя во многих статьях файлы устройств называют драйверами, это несколько неправильно. На самом деле это промежуточный узел, при помощи которого программы обмениваются данными с драйверами. Сам драйвер либо входит в состав ядра, либо реализован как модуль ядра.

#### ВНИМАНИЕ



**Права доступа файла устройства определяют возможность его использования пользователем или группой. Этот момент очень важен. Например, если вы можете соединиться с Интернетом или записывать информацию с помощью CD/DVD-привода только под суперпользователем, это означает, что вы просто не имеете нужных прав.**

Как уже говорилось выше, внешне эти файлы мало отличаются от обычных файлов и для работы с ними пользователь может использовать те же инструменты. Файлы устройств (как и драйверы) могут быть двух типов: символьные и блочные. Проще это понять, обратившись к способу передачи данных. Так, жесткий диск передает информацию блоками как единое целое. Это блочное устройство, и соответствующий ему файл устройства также будет блочным:

```
$ ls -l /dev/sda
brw-rw 1 root disk 8, 0 2010-05-22 16:20 /dev/sda
```

Буква `b` в выводе команды `ls` как раз и показывает на блочное устройство. Но некоторые устройства, например клавиатура, передают данные байт за байтом. Такие устройства называют символьными. К символьным кроме клавиатуры принадлежат терминал и разделы жесткого диска без файловой системы. Файлу символьного устройства соответствует буква `c` (рис. 4.3).

```

grinder@grinder-desktop ~ $ ls -l /dev/sda
brw-rw---- 1 root disk 8, 0 2010-05-22 18:11
grinder@grinder-desktop ~ $ ls -l /dev/tty0
crw--w---- 1 root root 4, 0 2010-05-22 18:10
grinder@grinder-desktop ~ $

```

Рис. 4.3. Блочное и символьное устройство в Linux

Чтение и запись данных в блочных устройствах буферизируются.

Файлы FIFO (от. англ. First Input First Output — «первый вошел, первый вышел»), или именованные каналы, обозначаемые в выводе буквой *f*, используются для передачи данных от одной программы к другой. Особенность FIFO состоит в том, что поток информации может идти только в одном направлении. Одна программа открывает файл FIFO только на запись, а другая — только на чтение. И в том случае, когда программам необходим двунаправленный канал обмена информацией, придется открывать два FIFO. Хотя альтернатива есть: использование сети или специальных файлов — сокетов (от англ. socket — «розетка»). При настройке разного типа серверов или программ вы можете столкнуться с обоими вариантами, и придется выбирать, какой из способов использовать. Например, демон *clamd* антивируса ClamAV или система управления базами данных MySQL могут использовать для обмена данными как сеть, так и сокет. Сеть обычно применяют для обмена данными тогда, когда общающиеся программы находятся на разных компьютерах, хотя и не всегда. Сокет в чем-то похож на FIFO, только данные передаются в обоих направлениях.

```

$ ls -al /dev/* | grep ^s

srw-rw-rw- 1 root root 0 2008-06-25 13:21 /dev/log

```

В этом примере вывод команды `ls -al` был перенаправлен на вход утилиты `grep`, которая предназначена для поиска строк с заданными параметрами. В качестве задания `grep` получила вывести все строки, начинающиеся на *s* (знак `^` означает привязку к началу строки).

Есть еще один момент. Поскольку специальный файл обозначает физическое устройство, у него отсутствует понятие размера. Вместо этого такие файлы обозначаются двумя числами *major* (старший) и *minor* (младший), которые для каждого

устройства будут уникальными. При этом `major` показывает на тип устройства, а `minor` — на конкретное устройство. Например, в выводе команды

```
$ ls -l /dev/sda
```

```
brw-rw---- 1 root disk 8, 0 2010-06-25 16:20 /dev/sda
```

цифра 8 соответствует `major`, а 0 — `minor`. Если посмотреть номера для файла `/dev/sda1`, который является первым разделом на этом диске, то обнаружим, что цифра `major` осталась прежней, а номер `minor` изменился и стал равен 1.

Самое интересное, что, если нужные файлы устройств система не создала автоматически, администратор это может сделать вручную. Для этого применяется специальная команда `mknod` (от англ. *make node* — «создать узел») или скрипт `MAKEDEV`.



#### ПРИМЕЧАНИЕ

Полностью формат команды и дополнительные ключи можно просмотреть в справочном руководстве, вызываемом командой `man mknod`.

Чтобы использовать `mknod`, нужно знать `major` и `minor` числа устройства. Формат команды такой:

```
mknod /dev/имя_файла { c | b } MAJOR MINOR
```

Буквы `c` и `b` позволяют указать, какое устройство нужно создать: символьное или блочное. Цифру `major` можно узнать, просмотрев список уже имеющихся устройств или проанализировав вывод команды `lsdev`. Но основным источником информации по номерам `major` и `minor` является файл `devices.txt`, который находится в каталоге **Documentation** исходных текстов ядра.

## Права доступа

Теперь коснемся одного из самых важных вопросов, с которыми сталкивается пользователь при знакомстве с Linux. Речь пойдет о правах доступа. Файлы в Linux, как, впрочем, и процессы, созданные пользователем, имеют двух владельцев: пользователя (`user owner`) и группу-пользователя (`group owner`). Кроме этого определяются права для остальных (`other`), то есть тех, которые не вошли в первые два списка. Каждый пользователь может быть членом сразу нескольких групп, одна из которых называется первичной (`primary`), а все остальные — дополнительными (`supplementary`). Наличие групп дает определенную гибкость в организации доступа к конкретному файлу. Например, чтобы разрешить запись дисков в Linux Mint, достаточно добавить пользователя в группу `cdrom`. Аналогично можно организовать совместное использование некоторого ресурса: достаточно создать новую группу и включить в нее всех, кому это действительно необходимо. Обычно владельцем

файла является пользователь, который его создал, а владелец-группа устанавливается равной первичной группе пользователя, создавшего файл.



#### ПРИМЕЧАНИЕ

В некоторых версиях Unix владелец-группа наследуется от владельца-группы каталога, в котором создается файл.

Просмотреть текущие права можно при помощи команды `ls -l`, как это мы делали ранее:

```
$ ls -l sound.mp3
-rw-r--r-- 1 grinder grinder 6706501 2010-05-11 00:29 sound.mp3
```

В первой позиции показываются права доступа по категориям; что означает `-rw-r--r--`, разберем чуть ниже. В третьей и четвертой (то есть **grinder grinder**) выводятся владелец-пользователь и группа.

Чтобы изменить владельцев файла, используется команда `chown`, принимающая в качестве параметров имя нового владельца или группу и список файлов. Формат такой:

```
$ sudo chown new_owner file1 file2 ...
```

На месте названия файла может быть и имя каталога, но при этом владелец файлов внутри каталога не изменится, а чтобы это произошло, добавляем флаг `-R`. При использовании любой команды оболочки можно пользоваться регулярными выражениями, если есть необходимость отобразить файлы, удовлетворяющие определённому критерию:

```
$ sudo chown -R user *
```

В примере владельцем всех файлов в текущем каталоге и подкаталогах будет пользователь **user**. Команда `chown` позволяет установить и группу-владельца. Для этого нужно сразу за именем владельца без пробелов и других знаков поставить двоеточие и написать название необходимой группы:

```
$ sudo chown -R sergej:video *
```

Допускается и такой вариант записи:

```
$ sudo chown -R :video *.
```

Кроме `chown` для изменения владельца группы можно использовать команду `chgrp`, синтаксис использования данной команды аналогичен предыдущей:

```
$ chgrp audio /home/user/*
```

Владение файлом определяет те операции, которые тот или иной пользователь может совершить над файлом. Самые очевидные из них — это изменение владельца и группы для некоторого файла. Данные операции может проделать суперпользователь и владелец файла (для некоторых — только суперпользователь). Причем группу, если вы являетесь владельцем файла, можно изменить только на свою первичную (по умолчанию имеет то же название, как и имя соответствующего пользователя). Все эти ограничения введены по нескольким причинам. Во-первых, для того чтобы никто не мог подсунуть какой-нибудь зловредный файл или скрипт, а во-вторых, чтобы, если на компьютере установлен лимит дискового пространства для конкретного пользователя, нельзя было превысить его, просто переопределив владельца.

Кроме владения файлом права доступа определяют базовые операции, которые можно совершить над файлом. Их три:

- Read — доступ на чтение;
- Write — доступ на запись;
- eXecute — доступ на выполнение.

Буквы, выделенные верхним регистром, как раз и показываются в выводе команды `ls -l` в первой позиции. Все указанные операции устанавливаются для каждой из трех категорий отдельно. Причем изменить их может, опять же, только владелец-пользователь и суперпользователь.

Для установки соответствующих прав используется команда `chmod`. Применяется она в двух формах: абсолютной, когда игнорируются старые права и устанавливаются новые, и относительной, когда к имеющимся правам добавляются новые (или убираются старые). Абсолютная форма предполагает задание прав доступа в восьмеричной форме. Чтобы получить полный код необходимого режима файла, нужно просто сложить значения кодов, приведенных в табл. 4.1.

**Таблица 4.1. Код команды `chmod`**

Восьмеричный код	Режим файла
0001	Право на выполнение для всех
0002	Право на запись для всех
0004	Право на чтение для всех
0010	Право на выполнение для группы
0020	Право на запись для группы
0040	Право на чтение для группы
0100	Право на выполнение для владельца
0200	Право на запись для владельца

Продолжение 

Таблица 4.1 (продолжение)

Восьмеричный код	Режим файла
0400	Право на чтение для владельца
1000	Включения бита сохранения задачи
2000	Если файлом, выполняемым включения бита, является SGID
4000	Если файлом, выполняемым включения бита, является SUID

Таким образом, команда

```
$ chmod 755 file
```

устанавливает следующие права доступа: если это исполняемый файл, то запустить его на выполнение и прочитать содержимое имеют право все (то есть владелец, группа и остальные), а владелец дополнительно имеет право на изменение содержимого — запись.

Относительная форма команды требует конкретного указания классов доступа (и — владелец, g — группа, o — остальные, a — все вместе), соответствующие права доступа (r — чтение, w — запись, x — выполнение) и операцию, которую необходимо произвести для списка файлов (+ — добавить, — удалить, = — присвоить) для соответствующего списка файлов. Например, команда

```
$ sudo chmod u+w, ug+r, a+x file
```

добавляет к имеющимся право запустить файл на выполнение, группа и владелец смогут прочесть содержимое, а владелец, кроме того, и изменить содержание. Применение = относится, скорее, к абсолютному заданию прав доступа, так как устанавливает соответствующие права вместо имеющихся.

К слову, вызвав свойства файла или каталога и перейдя на вкладку Права, вы можете изменить группу и параметры доступа к файлу для тех объектов, владельцами которых вы являетесь (рис. 4.4).

Для каталогов права доступа имеют немного другой смысл. Каталог, как говорилось выше, это файл, содержащий имена всех файлов, которые имеются в данном каталоге. Так вот, право на чтение каталога позволяет всего лишь получить имена файлов, находящихся в данном каталоге. А вот для того, чтобы получить другую дополнительную информацию, нужно право на исполнение. Чтобы перейти внутрь каталога, необходимо иметь право на выполнение. Кстати, из-за наличия этих особенностей можно добиться так называемого эффекта dark directory, когда создаем в каталоге файлы, которые доступны, только если пользователь знает точно имя соответствующего файла. Право на запись для каталога позволяет изменять его содержимое, то есть удалять и записывать файлы, при этом права доступа к конкретному файлу игнорируются.

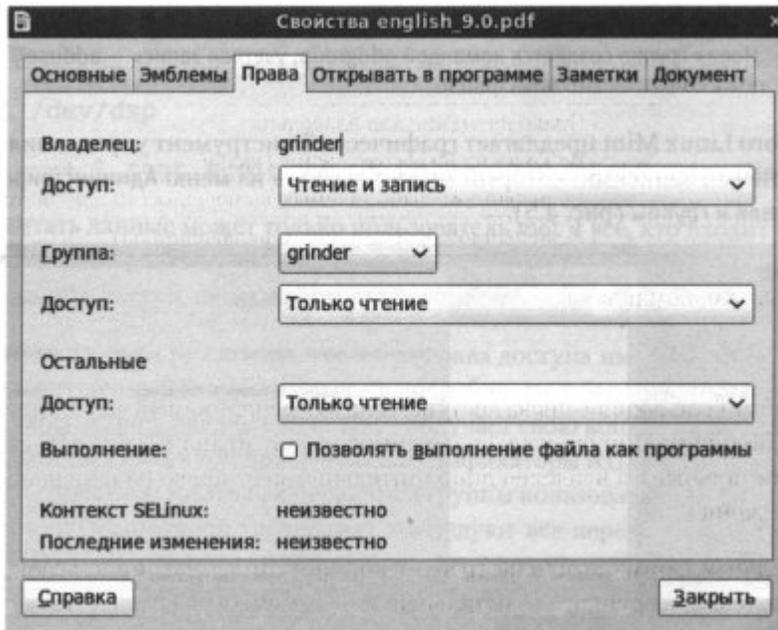


Рис. 4.4. Изменение параметров доступа к файлу

Обратите внимание еще на один момент. Права на доступ проверяются в следующей последовательности: суперпользователь, владелец, группа-владелец и остальные. Следовательно, если вы, являясь владельцем, забыли установить для себя право на запись, но установили его всем остальным, то не надейтесь, что сможете записать в него что-нибудь, даже если вы являетесь членом группы, просто потому, что все остальные могут, а вы являетесь владельцем. Система при запросе нужного ресурса проверит, кем он запрашивается в приведенной выше последовательности, и допустит только к разрешенным операциям, дальнейшая проверка прав просто не будет проводиться.

Группу, в которую входит пользователь, можно изменить, отредактировав файл `/etc/group`. Внутри файл состоит из ряда строк вида

```
cdrom:x:24:grinder
```

то есть в данной строке описана группа `cdrom`, в которую входит пользователь `grinder`. Чтобы добавить в эту группу пользователя `user`, просто дописываем его логин через запятую:

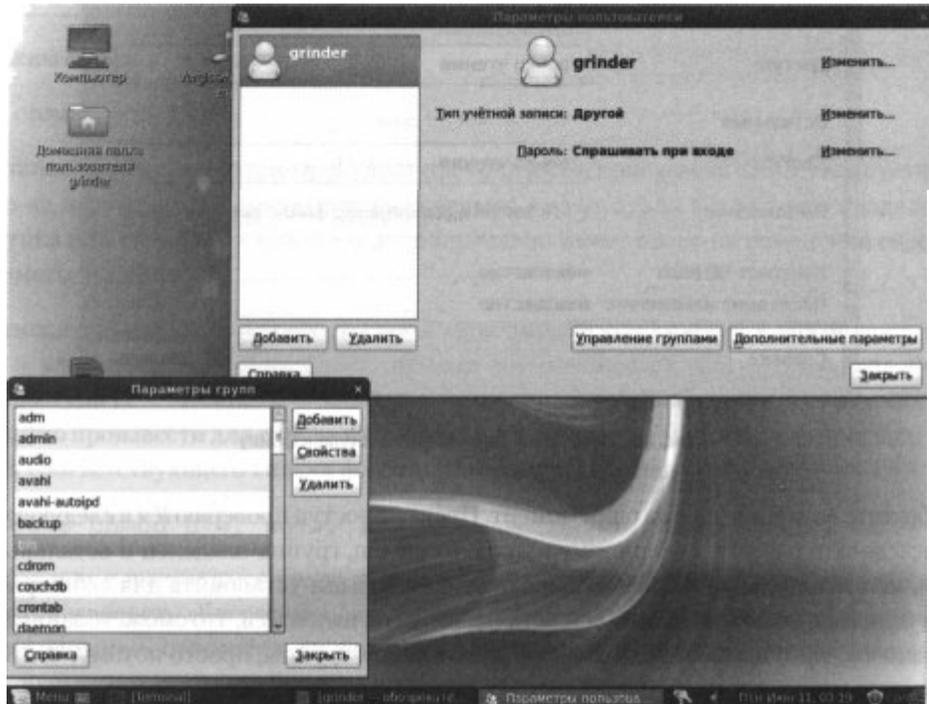
```
cdrom:x:24:grinder,user
```

Цифра 24 означает GID (group id) — цифровое значение группы.

**ПРИМЕЧАНИЕ**

Новая группа создается командой `addgroup`, учетная запись — `adduser`.

Кроме этого Linux Mint предлагает графический инструмент управления группами и учетными записями, который можно вызвать из меню **Администрирование • Пользователи и группы** (рис. 4.5).



**Рис. 4.5. Управление группами и учетными записями**

С его помощью можно очень просто добавить новую учетную запись или группу и установить членство в группах.

Данные об учетной записи хранятся в файле `/etc/passwd`. Формат такой:

```
username:пароль:uid:gid:uid комментарий:домашний каталог:shell
```

Например:

```
grinder:x:1000:1000:grinder,,,:/home/grinder:/bin/bash
```

**ПРИМЕЧАНИЕ**

Все обычные учетные записи имеют UID начиная с 1000.

Теперь узнаем, зачем это нужно. Например, смотрим данные устройства, выводящего звук:

```
$ ls -l /dev/dsp
```

```
crw-rw----+ 1 root audio 14, 3 2010-06-10 11:59 /dev/dsp
```

то есть считать данные может только пользователь `root` и все, кто входит в группу `audio`. Следовательно, если у вас не проигрывается звук и система выдает, что не может получить доступ, просто проверьте свои права.

Справедливости ради нужно отметить, что права доступа имеет не пользователь, а процесс, запущенный им. Не вдаваясь в подробности, поясню, что каждый пользователь, зарегистрировавшись в системе, получает свою копию текущего процесса `shell`, который имеет установленные идентификаторы `RID` и `RGID` — реальные идентификаторы пользователя и первичной группы пользователя. А все процессы, запущенные пользователем (дочерние), наследуют все переменные, в том числе `RID` и `RGID`.

У нас остались нерассмотренными три режима файла: бит сохранения задачи (`sticky bit`), а также флаги `SUID` и `SGID`. Со `sticky bit` все просто, этот бит указывает на необходимость сохранения копии выполняющейся программы в памяти после завершения выполнения. Данный режим позволяет сэкономить время на запуске программы при частом использовании, но в современных системах применение этого режима встречается редко.

Флаги `SUID` и `SGID` позволяют изменить (расширить) права пользователя (группы), запустившего программу, на время ее выполнения. Как уже говорилось, запущенное приложение имеет такие же права доступа к системным ресурсам, как и пользователь, запустивший программу. А установки этих флагов позволяют назначить права доступа исходя из прав доступа владельца файла. Следовательно, если владельцем запущенного приложения является `root`, то любой, независимо от того, кто запустил данное приложение, будет иметь права суперпользователя. Вместе с тем при установке флага `SUID` наследуются права владельца файла, а `SGID` — группы-владельца.

Буква `s` в выводе команды `ls -l` означает, что установлен флаг `SUID`, а владельцем файла является — `root`, и теперь, кто бы ни запустил утилиту на выполнение, на время работы программы он временно получает права суперпользователя, то есть право произвести запись в защищенный системный файл. Естественно, утилита должна (и делает это) производить изменение учетной записи только запустившего ее пользователя. Как вы понимаете, требования по безопасности к программам, использующим данный метод, должны быть повышенными.

Для установки битов `SUID/SGID` в символьной форме используется буква `s`, `sticky bit` устанавливается буквой `-t`, а с помощью буквы `l` можно установить

блокировку файла для устранения возможных конфликтов, когда несколько процессов попытаются работать с одним и тем же файлом.



#### ПРИМЕЧАНИЕ

Обычно sticky bit установлен в каталоге /tmp. Это сделано для того, чтобы удалить файл мог только пользователь, создавший его. При установке бита SGID для каталога все вновь созданные файлы будут теперь наследовать группу не по пользователю, создавшему его, а по группе — владельцу каталога.

## Монтирование разделов

Как уже говорилось в гл. 2, все разделы жесткого диска, сменные носители и сетевые ресурсы монтируются в дерево файловой системы. Разберем этот вопрос чуть подробнее.

Вообще, проблем с подключением разделов и сетевых ресурсов Windows сети в Linux Mint нет. Во время установки можно сразу прописать информацию о точках монтирования, которая затем будет использована при загрузке системы. Все такие разделы автоматически будут монтироваться. Иначе, достаточно лишь выбрать устройство в окне файлового менеджера **Nautilus**, дважды щелкнув на нем кнопкой мыши, чтобы получить к нему доступ. Например, чтобы перейти к сетевому ресурсу, размещаемому на Windows-системе (протокол SMB, Server Message Block), открываем Сеть выбираем группу **Windows** (рис. 4.6), затем переходим к конкретному компьютеру и, выбрав сетевую папку, дважды щелкаем на ней кнопкой мыши.

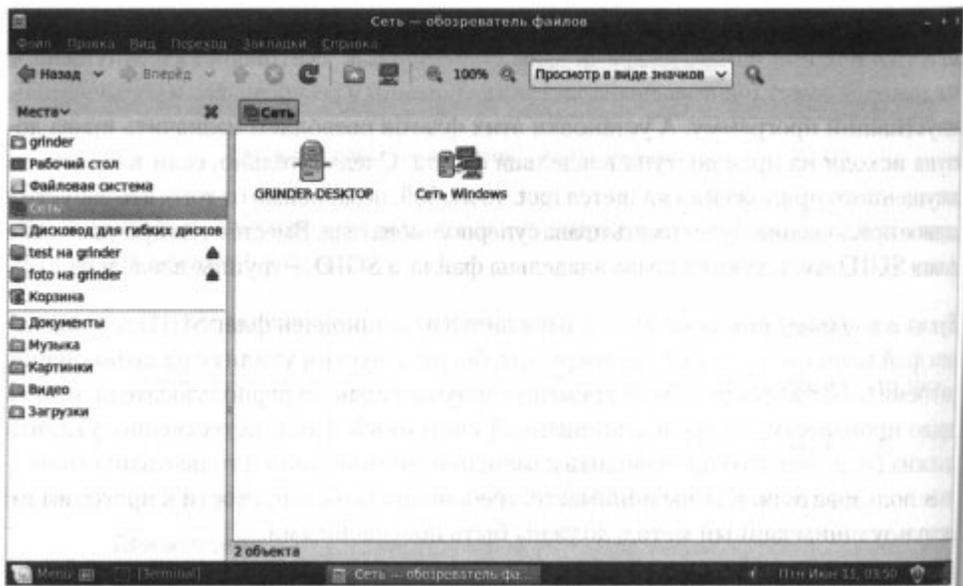


Рис. 4.6. Монтируем сетевой ресурс

После этого папка будет автоматически смонтирована. Ярлык для доступа появится на **Рабочем столе** и в боковом окне **Nautilus** (см. рис. 4.6). Причем пользователь получает доступ не только на чтение, но и на запись, если позволяют права доступа, установленные владельцем ресурса. Чтобы размонтировать, достаточно нажать на небольшой треугольник рядом с названием ресурса или выбрать в контекстном меню пункт **Отключить**.

Но в некоторых случаях необходимо постоянное монтирование ресурса, с возможностью задания альтернативной кодировки имен файлов и прав доступа. Тогда об этом позаботиться нужно самому.

## Команда mount

Команда `mount` является тем универсальным средством, которое позволяет проинформировать все необходимое. Чтобы просмотреть список смонтированных устройств, вводим ее без аргументов (для чтения права `root` не требуются):

```
$ mount

/dev/sdal on / type ext4 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nosuid,nodev)
none on /sys type sysfs (rw,noexec,nosuid,nodev)
none on /sys/fs/fuse/connections type fusectl (rw)
none on /sys/kernel/debug type debugfs (rw)
none on /sys/kernel/security type securityfs (rw)
none on /dev type devtmpfs (rw,mode=0755)
none on /dev/pts type devpts (rw,noexec,nosuid,gid=5,mode=0620)
none on /dev/shm type tmpfs (rw,nosuid,nodev)
none on /var/run type tmpfs (rw,nosuid,mode=0755)
none on /var/lock type tmpfs (rw,noexec,nosuid,nodev)
none on /lib/init/rw type tmpfs (rw,nosuid,mode=0755)
binfmt_misc on /proc/sys/fs/binfmt_misc type binfmt_misc
(rw,noexec,nosuid,nodev)

gvfs-fuse-daemon on /home/grinder/.gvfs type fuse.gvfs-fuse-
daemon (rw,nosuid,nodev,user=grinder)
```

В результате получим список, на первой позиции которого указывается исходный ресурс, на его месте может быть `none` (системный ресурс) или, как в нашем случае, программа `gvfs-fuse-daemon`. На второй — точка, куда он смонтирован.



#### ПРИМЕЧАНИЕ

Аналогичный список можно получить, прочитав содержимое файла `/etc/mtab`.

Чтобы смонтировать раздел или другой *ресурс* при помощи **mount**, в самом простом случае достаточно указать его имя и точку монтирования. Например, раздел `/dev/sdb2` следует подключить к `/home/grinder/audio`:

```
$ sudo mount /dev/sdb2 /home/grinder/audio
```

Теперь, перейдя в `/home/grinder/audio`, мы можем работать с файлами внутри. Но часто параметры монтирования, установленные по умолчанию, не совсем удобны.



#### СОВЕТ

Посмотреть полную информацию о разделе диска — доступный объем, свободное место — можно с помощью команды `df-h`.

Файловая система обычно определяется автоматически, но если на разделах подключаемого жесткого диска имеются логические ошибки, то необходимо дополнительно указать и тип файловой системы при помощи ключа `-t`:

```
$ sudo mount -t ext3 /dev/sdb2 /home/grinder/audio
```

В примере выбран тип файловой системы `ext3`. Другие файловые системы монтируют со следующими параметрами:

- `-t ext2`, `-t ext3`, `-t ext4`, `-t reiser`, `-t xfs` — монтирование разделов с соответствующими файловыми системами;
- `-t ntfs-3g` — монтирование раздела NTFS;
- `-t vfat` — монтирование раздела FAT32;
- `-t iso9660` — монтирование привода компакт-дисков или ISO-образа;
- `-t smbfs` или `-t cifs` — монтирование ресурса системы Windows.

Поддерживаемые дистрибутивом файловые системы можно узнать, просмотрев вывод команды `dmesg` или список исполняемых файлов `mount.*`. Последнее сделать очень просто: набираем в терминале `mount.` и нажимаем клавишу табуляции.



#### СОВЕТ

Полный список всех возможных значений `-t` можно посмотреть в `man mount`.

Параметры доступа к примонтированному разделу указываются при помощи флага `-o`. Для доступа на чтение/запись указывается флаг `-o rw`, для доступа только на чтение `-o ro`. Разрешим чтение и запись в нашем примере:

```
$ sudo mount -t ext3 -o rw /dev/sdb2 /home/grinder/audio
```



#### ПРИМЕЧАНИЕ

Параметры доступа к разделу указываются в последней позиции вывода команды `mount`.

Но в зависимости от сборки ядра кодировка имен файлов, набранных латиницей, в примонтированном разделе может быть нечитаема. Укажем ее явно:

```
$ sudo mount -t ext3 -o rw,icharset=utf8,codepage=866 /dev/sdb2 /home/grinder/audio
```

Параметр `-o icharset=utf8` явно задает кодировку системной локали, в Linux Mint это `utf8`, а `codepage=866` добавляет поддержку кодировки русского языка.

Параметр `-o force` позволяет принудительно монтировать разделы жесткого диска в том случае, если по неопределенным причинам утилита `mount` отказывается его монтировать с обычными параметрами.

Компакт-диск, вставленный в привод, также монтируется автоматически. Вручную его смонтировать можно так:

```
$ sudo mount -t iso9660 -o ro /dev/cdrom /mnt
```

Но `mount` интересна еще и тем, что позволяет монтировать ISO-образы диска. Для этого кроме типа файловой системы указываем и параметр `-o loop`.

```
$ sudo mount -t iso9660 -o loop ~/file.iso /home/iso
```

Чтобы размонтировать раздел, достаточно указать либо устройство, либо точку монтирования:

```
$ sudo umount /dev/sdb2
```

```
$ sudo umount /home/grinder/audio
```

Сетевой диск монтируется аналогичным образом — указываем его сетевое имя и точку монтирования. Дополнительно можно задать логин и пароль для доступа. Если этого не сделать, то пароль будет запрошен в момент подключения, а логин будет взят текущий, под которым работает пользователь. Но для начала нам нужно установить пакет `smbfs`:

```
$ sudo aptitude install smbfs
```

Теперь вводим

```
$ sudo mount -t smbfs -o изгпаше=логин,password=пароль //server/Video /home/grinder/video
```



#### ПРИМЕЧАНИЕ

Вместо имени можно указать IP-адрес системы.

## Автоматическое монтирование устройств — /etc/fstab

Естественно, для постоянных ресурсов очень неудобно монтирование вручную, ведь придется каждый раз вводить одни и те же команды. Но в этом нет необходимости. Чтобы система сама смонтировала раздел или сетевую папку во время загрузки, их достаточно внести в файл /etc/fstab. Формат простой, каждая строка соответствует устройству, знак решетки # означает комментарий, и такая строка игнорируется. По сути параметры внутри файла напоминают и повторяют опции команды mount. После установки системы внутри найдем несколько записей:

```
$ cat /etc/fstab

# <file system> <mount point> <type> <options> <dump> <pass>

# корневой раздел /dev/sda1 смонтирован по UUID, файловая система ext4

UUID=5bc61bd9-38b5-43a4-b3e8-11f9fd8ab389 /ext4 errors=remount-ro 0 1

# раздел /dev/sda2 файловая система ReiserFS

/dev/sda2 / reiserfs notail,relatime 0 1

# раздел swap на /dev/sda5

UUID=34ef4987-4f77-42f5-9794-fc57d822b05d none swap sw
```

Кроме того, в этом файле прописываются ресурсы, которые не нужно монтировать автоматически, например привод компакт-дисков или любой другой ресурс:

```
/dev/cdrom /media/cdrom iso9660,udf ro,noauto,user,exec 0 0
```

Чтобы его не монтировать, в качестве параметра указывают **noauto**. Это **удобно** еще и потому, что мы можем смонтировать прописанный в /etc/fstab ресурс, просто указав его имя или точку монтирования, а все остальные параметры будут **взяты из** файла. Раздел с файловой системой NTFS монтируется через драйвер NTFS-3g:

```
/dev/sda6 /mnt/windows ntfs-3g defaults 0 0
```

## Перенос раздела /home

Многие новички изначально выбирают самую простую схему размещения, когда все дерево находится на одном (корневом) разделе жесткого диска. Несмотря на первоначальную простоту, такая схема имеет и свои недостатки. Например, по мере установки новых программ и заполнения данными каталогов пользователей может случиться так, что жесткий диск заполнится. Поэтому в один прекрасный момент вы просто не сможете зарегистрироваться в системе, так как негде будет создать временный файл.



### ПРИМЕЧАНИЕ

Некоторые программы, среди которых и видеоподсистема X Window, используют для обмена данными специальные файлы FFO или сокеты (о них чуть дальше). Если их негде будет создавать, то программа просто не запустится.

По мере наполнения файла или с появлением опыта возникает вопрос, как перенести на другой раздел некоторые данные и изменить точку монтирования. На пользовательских системах чаще всего выносят на отдельный раздел каталог /home. Ничего сложного в этом нет. Но простое копирование командой `cp -R` каталогов даст непредсказуемый результат, и все из-за того, что кроме обычных файлов внутри могут быть специальные файлы или ссылки. Однако команда `cp` имеет специальные ключи, которые позволяют перенести данные со всеми атрибутами, включая права доступа к файлу.

Для примера перенесем раздел /home в раздел /dev/sda5. Создание и форматирование файловой системы и подробности работы с командой `mount` мы рассмотрим далее в этой главе, пока ограничимся командами.

Создаем каталог, в который смонтируем /dev/sda5:

```
$ sudo mkdir /mnt/home
```

Монтируем /dev/sda5 в /mnt/home:

```
$ sudo mount /dev/sda5 /mnt/home
```

Теперь копируем файлы из каталога /home в /mnt/home. Это можно сделать либо используя команду `cp`, либо предварительно создав архив.

```
$ sudo cp -a /home /mnt/home
```

Ключ `-a` (`--archive`) при копировании по возможности сохраняет структуру и атрибуты исходных файлов. Правда, в описании указано, что он не обязан сохранять структуру каталогов, но обычно с этим проблем нет. Аналогично можно перенести и всю файловую систему, для этого следует в качестве исходного каталога указать корневой, то есть «/». Но если на компьютере находится несколько

дисков, может быть полезен параметр `-x` или `--one-file-system`, который позволяет ограничить работу `cp` в пределах одной файловой системы, то есть чтобы она «не перескочила» на другой диск. Но существует один момент, о котором нужно знать. Сегодня есть несколько вариантов и версий команды `cp`, которые поддерживают свои спецификации. Так, параметр `-a` относится к версии `cp`, предлагаемой проектом GNU (<http://www.gnu.org/>).



#### ПРИМЕЧАНИЕ

Как я уже упоминал, название проекта GNU происходит от рекурсивного акронима GNU's Not Unix («GNU не Unix»), его задачей является разработка свободной UNIX-подобной операционной системы. В настоящее время разработано большое количество утилит, но ядро GNU/Hurd еще не готово (хотя оно разрабатывается с 1990 г. и уже нормально функционирует). Сейчас GNU с успехом заменяет GNU/Linux, в котором используются утилиты проекта GNU и ядро Linux.

Хотя в Linux используется как раз вариант `cp` и потому проблем не возникает, но при работе в других UNIX-подобных системах к ним нужно быть готовым. В принципе, вместо параметра `-a` можно использовать — `dpR`.

```
$ sudo cp -dpR /home /mnt/home
```

Еще одним вариантом является использование архиватора `tar`, который может быть полезен при создании резервной копии раздела или переноса его в другую систему, так как он умеет сохранять оригинальные атрибуты.

В нашем случае команда выглядит следующим образом:

```
$ sudo (cd /home && tar cf - .) | (cd /mnt/home && tar xpvf -)
```

Иными словами, мы создали архив `tar` каталога `/home` (параметр `c`, от англ. create — «создать»). Но вместо того, чтобы создать файл, передав вывод команды другой копии `tar`, мы фактически распаковали архив, разместив файлы на своих местах (параметр `x` — предназначен для распаковки архива (от англ. eXtract — «извлечь»). Параметр `-p` позволяет установить нужные права на файлы, извлекаемые из архива.

При необходимости команды можно разделить и сжать архив при помощи программы `gunzip` (или `bzip2`). Создадим архив:

```
$ cd /home; sudo tar czf home.tar.bz .
```

и разархивируем:

```
$ cd /mnt/home; sudo tar xzf home.tar.bz
```

Теперь, когда файлы находятся на своем месте, смонтируем раздел в `/home`:

```
$ sudo mount /dev/sda5 /home
```

Некоторое время тестируем. Если все работает нормально, добавляем новую точку монтирования в файл `/etc/fstab`, чтобы раздел монтировался автоматически при загрузке системы (подробнее о команде `mount` и файле `fstab` поговорим дальше). И не забываем удалить старые файлы с корневого раздела, ведь они фактически остались на диске. Размонтируем, если нужно, раздел, чтобы случайно не удалить скопированные файлы:

```
$ sudo umount /dev/sda5
```

Теперь удаляем ненужные файлы:

```
$ sudo rm -Rf /home/*
```

Диск почищен, монтируем `/dev/sda5` и работаем.

В итоге мы получили некоторые сведения о назначении основных каталогов. Такой подход имеет большую гибкость, поскольку можно все дерево каталогов разместить на одном разделе жесткого диска либо разнести их по разным разделам одного или нескольких жестких дисков или даже сетевых ресурсов. Замена или добавление носителя никак не скажется на алгоритме работы системы и останется полностью прозрачной для пользователя.

## Управление процессами

Файлы файлами, а жизнь системе дают процессы. Грубо говоря, процесс — это программа в стадии выполнения. После своего запуска процесс выполняет свою работу с правами учетной записи запустившего его пользователя. Одновременно в системе могут выполняться сотни процессов, которыми нужно уметь управлять. Просмотреть список процессов можно при помощи программы `ps` (`process status`). Введенная без аргументов, она выведет процессы пользователя, который ее запустил. Чтобы получить список всех процессов, следует добавить ключ `-a` (рис. 4.7).

В этом случае команда покажет:

- `USER` — учетная запись, от имени которой выполняется процесс;
- `PID` (`Process Identifier`) — идентификатор процесса, число от 1 до 65 535;
- `%CPU` и `%MEM` — процент использования времени процессора и занимаемой памяти;
- `STAT` — статус процесса;
- `COMMAND` — команда, которая запустила процесс, обычно ее терминал обрезает, поэтому, чтобы увидеть команду полностью, следует добавить ключ `w`.

```

Terminal
grinder@grinder-desktop ~ $ ps
  PID TTY          TIME CMD
 4148 pts/0    00:00:01 bash
 5288 pts/0    00:00:00 ps
grinder@grinder-desktop ~ $ ps -aux
Warning: bad ps syntax, perhaps a bogus '-?' See http://procps.sf.net/faq.html
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  0.0  0.2 23688 1192 ?        Ss   Jun10   0:03 /sbin/init
root            2  0.0  0.0      0     0 ?        S    Jun10   0:00 [kthreadd]
root            3  0.0  0.0      0     0 ?        S    Jun10   0:00 [migration/0]
root            4  0.0  0.0      0     0 ?        S    Jun10   0:00 [ksoftirqd/0]
root            5  0.0  0.0      0     0 ?        S    Jun10   0:00 [watchdog/0]
root            6  0.0  0.0      0     0 ?        S    Jun10   0:00 [migration/1]
root            7  0.0  0.0      0     0 ?        S    Jun10   0:00 [ksoftirqd/1]
root            8  0.0  0.0      0     0 ?        S    Jun10   0:00 [watchdog/1]
root            9  0.0  0.0      0     0 ?        S    Jun10   0:00 [events/0]
root           10  0.0  0.0      0     0 ?        S    Jun10   0:01 [events/1]
root           11  0.0  0.0      0     0 ?        S    Jun10   0:00 [cpuset]
root           12  0.0  0.0      0     0 ?        S    Jun10   0:00 [khelper]
root           13  0.0  0.0      0     0 ?        S    Jun10   0:00 [netns]
root           14  0.0  0.0      0     0 ?        S    Jun10   0:00 [async/mgr]
root           15  0.0  0.0      0     0 ?        S    Jun10   0:00 [pm]
root           17  0.0  0.0      0     0 ?        S    Jun10   0:00 [sync_supers]
root           18  0.0  0.0      0     0 ?        S    Jun10   0:00 [bdi-default]
root           19  0.0  0.0      0     0 ?        S    Jun10   0:00 [kintegrityd/0]
root           20  0.0  0.0      0     0 ?        S    Jun10   0:00 [kintegrityd/1]
root           21  0.0  0.0      0     0 ?        S    Jun10   0:02 [kblockd/0]
root           22  0.0  0.0      0     0 ?        S    Jun10   0:01 [kblockd/1]
root           23  0.0  0.0      0     0 ?        S    Jun10   0:00 [kacpid]
root           24  0.0  0.0      0     0 ?        S    Jun10   0:00 [kacpi_notify]
root           25  0.0  0.0      0     0 ?        S    Jun10   0:00 [kacpi_hotplug]
root           26  0.0  0.0      0     0 ?        S    Jun10   1:01 [ata/0]
root           27  0.0  0.0      0     0 ?        S    Jun10   0:07 [ata/1]
root           28  0.0  0.0      0     0 ?        S    Jun10   0:00 [ata_aux]
root           29  0.0  0.0      0     0 ?        S    Jun10   0:00 [ksuspend_usbd]
root           30  0.0  0.0      0     0 ?        S    Jun10   0:00 [khubd]

```

Рис. 4.7. Просмотр информации о процессах

Есть и другие характеристики, более подробно вывод можно найти в документации (**man ps**). Статус процесса может быть разным: **S** — процесс в состоянии ожидания/сна (**sleeping**), **R** — процесс выполняется или готов к выполнению (**run**), **D** — процесс в глубоком сне (например, ожидает дискового ввода/вывода), **T** — процесс остановлен, **Z** — процесс-зомби, **<** и **N** — соответственно процесс с отрицательным и положительным значением `nice`.

В нормально работающей системе можно встретить все процессы, кроме **Z**. Это процесс, который уже завершился, но запись не удалена с таблицы процессов. Вреда он не причиняет, и ресурсы на него не тратятся, только занимает место в списке, и все. Чтобы его удалить, достаточно перезагрузить систему.

Предположим, зависла некоторая программа или собственноручно написанный скрипт — как остановить их выполнение? Для этого используется команда `kill`, в качестве параметра которой передается PID процесса.

Например, уничтожим процесс под номером 1234:

```
$ sudo kill 1234
```

Чтобы уничтожить процесс по имени, что очень удобно, когда имеются дочерние подпроцессы, следует использовать команду `killall`, которая принимает шаблон имени процесса.

```
$ sudo killall mpg123
```



#### ПРИМЕЧАНИЕ

Список процессов в динамике показывает команда `top` (рис. 4.8).

```

top - 06:44:38 up 18:46, 3 users, load average: 0.85, 0.29, 0.15
Tasks: 166 total, 1 running, 164 sleeping, 0 stopped, 1 zombie
Cpu(s): 1.6%us, 1.0%sy, 0.0%ni, 96.9%id, 0.3%wa, 0.2%hi, 0.0%si, 0.0%st
Mem: 505460k total, 465848k used, 39612k free, 11332k buffers
Swap: 1477624k total, 104252k used, 1373372k free, 185644k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  855 root        20   0 246m  23m 6388  S   3.4  4.7   7:36.59 Xorg
4146 grinder    20   0 215m  14m  9.8m S   1  2.9   0:10.90 gnome-terminal
   47 root        20   0   0     0   0   S   0  0.0   0:50.17 scsi_eh_1
 1233 root        20   0 11272 372 296  S   0  0.1   0:47.50 irqbalance
 1555 root        20   0 46684 404 328  S   0  0.1   0:19.13 udisks-daemon
5375 grinder    20   0 19216 1420 1028 R   0  0.3   0:00.07 top
   1 root        20   0 23688 1192 636  S   0  0.2   0:03.64 init
   2 root        20   0   0     0   0   S   0  0.0   0:00.11 kthreadd
   3 root        RT   0   0     0   0   S   0  0.0   0:00.15 migration/0
   4 root        20   0   0     0   0   S   0  0.0   0:00.68 ksoftirqd/0
   5 root        RT   0   0     0   0   S   0  0.0   0:00.00 watchdog/0
   6 root        RT   0   0     0   0   S   0  0.0   0:00.11 migration/1
   7 root        20   0   0     0   0   S   0  0.0   0:00.62 ksoftirqd/1
   8 root        RT   0   0     0   0   S   0  0.0   0:00.00 watchdog/1
   9 root        20   0   0     0   0   S   0  0.0   0:00.60 events/0
  10 root        20   0   0     0   0   S   0  0.0   0:01.36 events/1
  11 root        20   0   0     0   0   S   0  0.0   0:00.00 cpuset
  12 root        20   0   0     0   0   S   0  0.0   0:00.01 khelper
  13 root        20   0   0     0   0   S   0  0.0   0:00.00 netns
  14 root        20   0   0     0   0   S   0  0.0   0:00.00 async/mgr
  15 root        20   0   0     0   0   S   0  0.0   0:00.00 pm
  17 root        20   0   0     0   0   S   0  0.0   0:00.13 sync_supers
  18 root        20   0   0     0   0   S   0  0.0   0:00.09 bdi-default
  19 root        20   0   0     0   0   S   0  0.0   0:00.00 kintegrityd/0
  20 root        20   0   0     0   0   S   0  0.0   0:00.00 kintegrityd/1
  21 root        20   0   0     0   0   S   0  0.0   0:02.62 kblockd/0
  22 root        20   0   0     0   0   S   0  0.0   0:01.82 kblockd/1
  23 root        20   0   0     0   0   S   0  0.0   0:00.00 kacpid
  24 root        20   0   0     0   0   S   0  0.0   0:00.00 kacpi_notify

```

Рис. 4.8. Вывод процессов при помощи команды `top`

При помощи команды `nice` можно указать приоритет процесса, увеличив или уменьшив его. Повысить приоритет команды может только пользователь `root`. Команда может смещать приоритет в диапазоне от `-20` (наивысший приоритет) до `19` (низший приоритет) от текущего, то есть для повышения следует указать отрицательное число:

```
$ sudo nice -5 rhythmbox
```

В Linux Mint имеется и графический интерфейс, в котором можно получить список процессов, процент использования ресурсов (процессор, память, сеть) и заполнение

файловой системы. Открыть его можно в **Центр управления • Система • Системный монитор** (рис. 4.9).

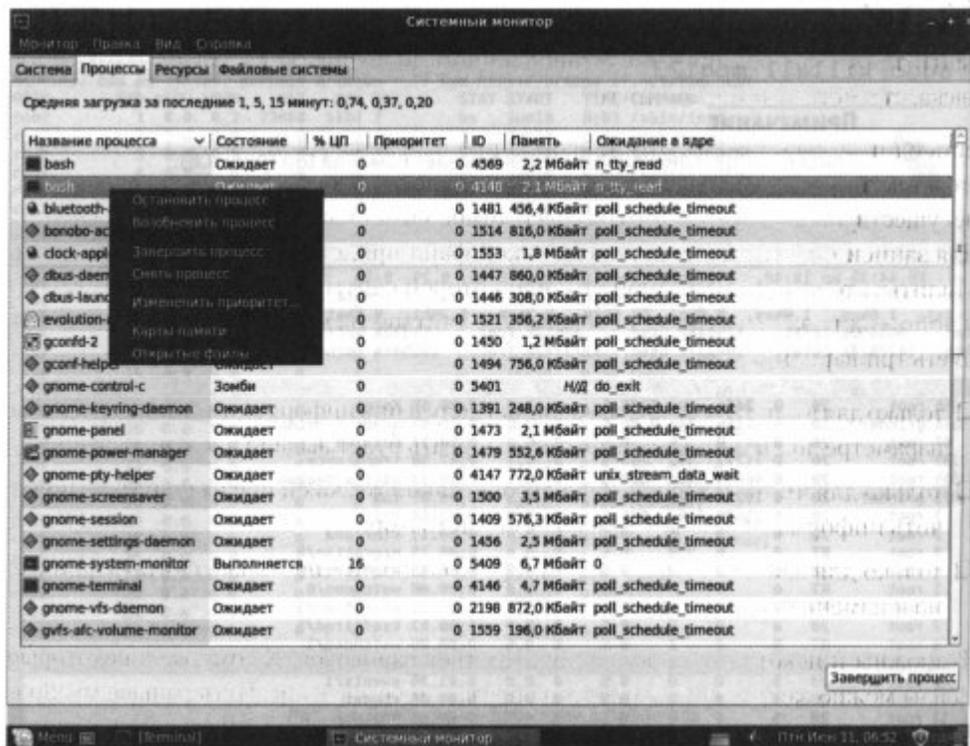


Рис. 4.9. Системный монитор

## Файловая система procfs

Некоторые параметры работы ядра и его модулей можно изменять, не перегружая систему. Для этого используется виртуальная файловая система procfs, размещаемая в каталоге /proc, которая представляет собой дополнительный механизм, позволяющий ядру обмениваться данными с процессами.



### ПРИМЕЧАНИЕ

Подробную информацию о работе procfs можно найти на сайте проекта <http://procfs.sf.net/>.

Первоначально procfs была разработана для свободного получения информации о состоянии процессов, теперь ее функции выше. Виртуальной она называется потому, что имеющиеся здесь файлы и каталоги на самом деле не располагаются на жестком диске. После загрузки ядра они находятся в оперативной памяти. Для пользователя этот механизм полностью прозрачен. Многие программы, в том числе

и имеющие графический интерфейс, собирают информацию из файлов в `/proc`, форматируют их и выводят результат на экран. Среди консольных утилит — это `top`, показывающая загрузку системы, и `ps`, выводящая список процессов. В `/proc` можно найти также информацию об установленном оборудовании, разметке жесткого диска, статистику и многое другое.

Есть один только нюанс: хотя информацию из файла можно прочитать, открыть его в текстовом редакторе не получится, не только потому, что файлов физически не существует, но и потому, что их содержимое может измениться в любую секунду. Для записи и считывания данных в такой файл придется использовать обычные утилиты: это команда `cat`, при помощи которой считывается содержимое файла, и `echo` — для записи. Кроме того, следует учитывать, что в файлы в `procfs` могут иметь три варианта доступа:

- только для чтения — предназначены для получения информации об определенном параметре, при попытке в них что-то записать будет выдано предупреждение;
- только для чтения пользователем `root` — такой же, как и предыдущий, но получить информацию может только администратор;
- только для записи пользователем `root` — позволяет не только считать данные, но и изменить параметр.

Возможны и некоторые комбинации этих трех вариантов. К тому же в некоторые файлы можно записать только строго определенное значение. Чуть раньше мы просматривали и изменяли алгоритм работы планировщика CFS. При попытке записи иного значения будет выдана ошибка. И это далеко не все, что можно сделать в `/proc`.

Если вывести список содержимого каталога `/proc`, то увидим большое количество файлов и подкаталогов. С `procfs` мы столкнемся еще не один раз на протяжении книги, пока рассмотрим только некоторые файлы, которые позволяют управлять работой ядра и его модулей.

Для примера проверим, включен ли кулер на процессоре:

```
$ sudo cat /proc/acpi/fan/FAN/state
status: on
```

Можно ли его выключить командой

```
echo 0 > /proc/acpi/fan/FAN/state
```

зависит от материнской платы. Кстати, текущую температуру процессора можно посмотреть так:

```
$ sudo cat /proc/acpi/thermal_zone/THRM/temperature
temperature: 23 C
```

Причем в Linux температура выводится в градусах Цельсия, в других системах вроде FreeBSD — в кельвинах. В файле `trip_points` можно считать или задать политику управления охлаждением системы:

```
$ sudo cat /proc/acpi/thermal_zone/THRM/trip_points
critical (S5): 65 C
passive: 63 C: tcl=4 tc2=3 tsp=60 devices=0xdf852338
active[0]: 63 C: devices=0xdf85ff90
```

Возможны три варианта политики: `critical` (критическая температура, после которой возможен автоматический переход в спящий режим), `passive` (уменьшение частоты процессора) и `active` (активный режим работы кулера). Причем последних может быть несколько ступеней — от 0 до 9, количество, опять же, зависит от оборудования. Шаблон команды на изменение этих параметров выглядит так:

```
echo -n "critical:hot:passive:active0:...:activeX" > trip_points
```

Например:

```
$ echo "105:100:100:78:70:60:50" > /proc/acpi/thermal_zone/TZ0/trip_points
```

Режим охлаждения можно посмотреть с помощью команды

```
$ sudo cat /proc/acpi/thermal_zone/THRM/cooling_mode
cooling mode: active
```

Пассивный режим можно устанавливать не во всех устройствах, хотя современные ноутбуки его обычно поддерживают.

Очень много параметров, которые можно изменить на лету, находятся в подкаталоге `/proc/sys`. Например, интересный параметр доступен в файле `/proc/sys/fs/file-max`, установленное здесь число определяет максимальное количество файлов, которые можно открыть. В разных дистрибутивах используется свое значение. При необходимости сюда можно записать свою цифру:

```
$ sudo echo "4096" /proc/sys/fs/file-max
```

Файл `/proc/sys/kernel/ctrl-alt-del` содержит цифру, которая указывает системе, как действовать в случае нажатия комбинации `Ctrl + Alt + Delete`. В этом файле возможно использование одного из двух значений:

- 0 — разрешает принимать эту комбинацию и производит нормальное выключение с предварительным корректным завершением программ;
- 1 — комбинация клавиш приводит к эффекту, подобному простому отключению питания без предварительного завершения программ.

По умолчанию используется 0. В файлах `/proc/sys/kernel/domainname` и `/proc/sys/kernel/hostname` можно установить имя домена и сетевое имя узла.

В `/proc/sys/net` собраны файлы, в которых можно изменить некоторые сетевые настройки. Например, в `/proc/sys/net/core/netdev_max_backlog` указывается максимальное количество пакетов в очередь на обработку. Если ядро не успевает обрабатывать пакеты, его лучше увеличить. Значение по умолчанию — 300, но в некоторых дистрибутивах установлено вплоть до 1000.

Файлы в каталоге `/proc/sys/vm` отвечают за работу с виртуальной памятью. Например, число в файле `/proc/sys/vm/swappiness` содержит коэффициент, который показывает, как часто необходимо переносить данные из оперативной памяти на жесткий диск. Чем выше это число, тем реже выполняется данная операция. В Ubuntu по умолчанию установлено следующее значение:

```
$ cat /proc/sys/vm/swappiness
```

Если оперативной памяти мало, а вы вынуждены загружать приложения, требующие большого объема памяти, можно попробовать увеличить этот параметр вплоть до максимальных 100. В этом случае загруженный OpenOffice.org или GIMP будет работать быстрее за счет того, что нужная информация задерживается в быстрой оперативной памяти. Но переключение между задачами будет выполняться медленнее.

## Постоянные установки ядра

Все параметры в каталоге `/proc/sys` будут действительны до первой перезагрузки системы, после этого придется всю процедуру выполнять повторно. Есть два пути решения данной задачи. Первый — прямолинейный. В одном из стартовых скриптов следует прописать все нужные команды так, как будто они вводятся в командную строку терминала. В каждом дистрибутиве Linux обычно имеется файл, предназначенный специально для пользовательских установок. В дистрибутиве Ubuntu и некоторых других — это файл `/etc/init.d/rc.local`. Если такого файла в вашем дистрибутиве нет, то наверняка есть файл с похожим именем. Например, чтобы добавить некоторые параметры, о которых говорилось ранее, открываем его в текстовом редакторе и добавляем в конце файла такие строки:

```
echo "70" > /proc/sys/vm/swappiness
```

```
echo "4096" /proc/sys/fs/file-max
```

И так далее.

Второй метод — использование утилиты `sysctl`, которая как раз и предназначена для управления параметрами ядра на лету. Для управления используется конфигурационный файл `/etc/sysctl.conf`, имеющий понятную структуру. При загрузке системы все описанные в нем параметры выполняются. После загрузки при помощи `sysctl` также можно управлять параметрами ядра, вводя команды непосредственно в окне терминала.

Указанные два способа немного различаются. Фактически названия всех переменных, которыми оперирует `sysctl`, являются именами файлов в `/proc/sys`. Причем, поскольку `sysctl` оперирует файлами `/proc/sys`, эта приставка в названии параметра отбрасывается. Так, файл `/proc/sys/fs/file-max` соответствует параметру `fs/file-max`. Иными словами, чтобы при загрузке изменить значение в `/proc/sys/fs/file-max`, в конфигурационном файле `/etc/sysctl.conf` записываем строку:

```
fs/file-max = 4096
```

Но поскольку слэш используется в именах файлов и к тому же в командной строке, его необходимо экранировать. Чтобы упростить набор при вводе параметра в терминале, слэш, обозначающий каталог, заменяется на точку. Таким образом, в нашем случае команда

```
echo "4096" > /proc/sys/fs/file-max
```

которую мы использовали ранее, при применении `sysctl` выглядит так:

```
$ sudo sysctl -w fs.file-max="4096"
```

```
fs.file-max = 4096
```



#### ПРИМЕЧАНИЕ

В файле `/etc/sysctl.conf` можно использовать оба варианта написания параметра.

Чтобы получить список всех переменных, доступных для изменения, достаточно ввести команду `sysctl -a` (рис. 4.10).

Вывод займет несколько экранов. Да, и не забывайте, что использование `sysctl` в терминале не добавит введенный параметр в свой конфигурационный файл. Если вы хотите, чтобы после перезагрузки изменения остались в силе, не забудьте его записать в `/etc/sysctl.conf`.



```
net.ipv6.conf.lo.accept_dad = -1
net.ipv6.conf.eth8.forwarding = 0
net.ipv6.conf.eth8.hop_limit = 64
net.ipv6.conf.eth8.mtu = 1500
net.ipv6.conf.eth8.accept_ra = 1
net.ipv6.conf.eth8.accept_redirects = 1
net.ipv6.conf.eth8.autoconf = 1
net.ipv6.conf.eth8.dad_transmits = 1
net.ipv6.conf.eth8.router_solicitations = 3
net.ipv6.conf.eth8.router_solicitation_interval = 4
net.ipv6.conf.eth8.router_solicitation_delay = 1
net.ipv6.conf.eth8.force_wd_version = 0
net.ipv6.conf.eth8.use_tempaddr = 0
net.ipv6.conf.eth8.temp_valid_lft = 604800
net.ipv6.conf.eth8.temp_preferred_lft = 86400
net.ipv6.conf.eth8.regen_max_retry = 5
net.ipv6.conf.eth8.max_desync_factor = 600
net.ipv6.conf.eth8.max_addresses = 16
net.ipv6.conf.eth8.accept_ra_defrtr = 1
net.ipv6.conf.eth8.accept_ra_pinfo = 1
net.ipv6.conf.eth8.proxy_ndp = 0
net.ipv6.conf.eth8.accept_source_route = 0
net.ipv6.conf.eth8.disable_ipv6 = 0
net.ipv6.conf.eth8.accept_dad = 1
net.ipv6.conf.eth8.ip6frag_high_thresh = 262144
net.ipv6.conf.eth8.ip6frag_low_thresh = 196608
net.ipv6.conf.eth8.ip6frag_time = 60
error: permission denied on key 'net.ipv6.route.flush'
net.ipv6.route.gc_thresh = 1024
net.ipv6.route.max_size = 4096
net.ipv6.route.gc_min_interval = 0
net.ipv6.route.gc_timeout = 60
net.ipv6.route.gc_interval = 30
net.ipv6.route.gc_elasticity = 0
net.ipv6.route.mtu_expires = 600
net.ipv6.route.min_adv_mss = 12
```

Рис. 4.10. Список доступных для изменения параметров

## Сбор информации о системе

Собрать точную информацию об установленном оборудовании можно при помощи как графических программ, так и консольных утилит. Графическая утилита HardInfo запускается из меню **Системные утилиты** и выводит все данные о компьютере, операционной системе, сетевых подключениях, включая используемые переменные, установки и прочую полезную информацию (рис. 4.11). Этот вариант очень простой: перебираем вкладки и получаем нужные данные. При необходимости программа позволяет сгенерировать отчет, сохранив результат в текстовый или HTML-файл.

Разберем, как получить новые данные в консоли. Так, чтобы получить информацию обо всех устройствах, подключенных к PCI-разъемам, используем команду `lspci` (рис. 4.12).

К слову, в Debian или Ubuntu она по умолчанию не устанавливается. Поэтому в ответ мы можем получить сообщение `lspci: command not found`, которое означает, что данная утилита не установлена. Но это легко исправить. Достаточно установить пакет `pciutils`:

```
$ sudo apt-get install pciutils
```

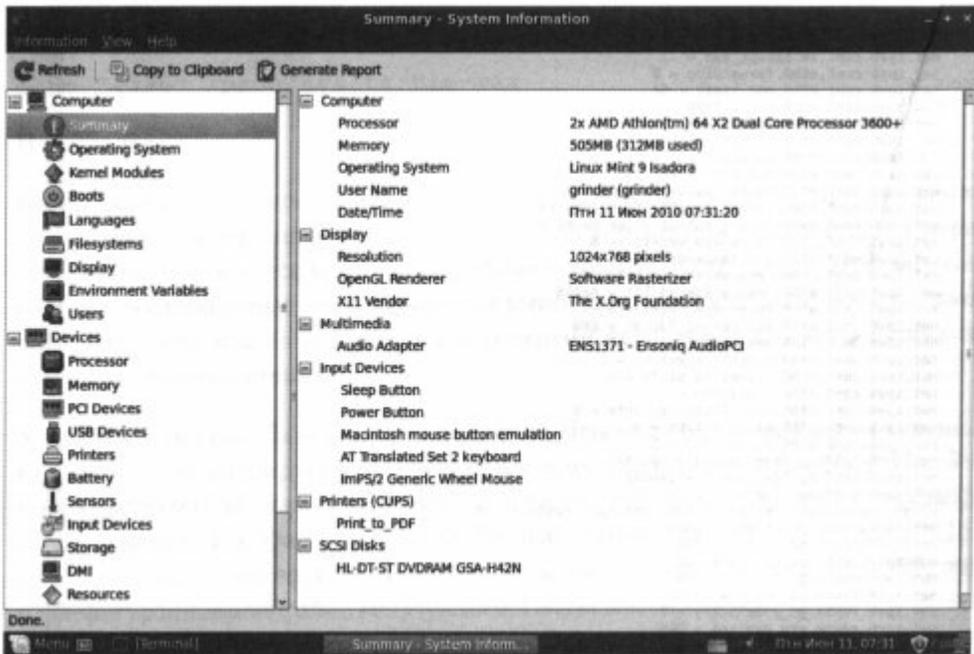


Рис. 4.11. Программа HardInfo

```

Terminal
root@minty:~# lspci
00:00.0 Host bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX Host bridge (rev 01)
00:01.0 PCI bridge: Intel Corporation 440BX/ZX/DX - 82443BX/ZX/DX AGP bridge (rev 01)
00:07.0 ISA bridge: Intel Corporation 82371AB/EB/MB PIIX4 ISA (rev 08)
00:07.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:07.3 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:07.7 System peripheral: VMware Virtual Machine Communication Interface (rev 10)
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
00:10.0 SCSI storage controller: LSI Logic / Symbios Logic 53c1030 PCI-X Fusion-MPT Dual
SI (rev 01)

```

Рис. 4.12. Вывод lspci

Повторяем запрос. Итак, мы уже знаем многое об оборудовании, теперь выясним марку процессора. Вводим в консоли (рис. 4.13):

```
$ cat /proc/cpuinfo
```

И последний шаг — получение информации об используемых модулях ядра. Подключаем все устройства, которые будут использоваться впоследствии, проверяем их работоспособность и выводим их список при помощи команды `lsmod`. Вам могут и не понадобиться все модули, которые будут показаны в списке. Узнать некоторую информацию о конкретном модуле можно, используя команду `modinfo` с указанием названия интересующего вас модуля.

```

grinder@grinder:~$ cat /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 15
model        : 75
model name    : AMD Athlon(tm) 64 X2 Dual Core Processor 3600+
stepping     : 2
cpu MHz      : 2015.986
cache size   : 256 KB
fpu          : yes
fpu_exception : yes
cpuid level  : 1
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
              clflush mmx fxsr sse sse2 syscall nx mmxext fxsr_opt rdtscp lm 3dnowext 3dnow constant_t
              sc rep_good tsc_reliable pni cx16 hypervisor lahf_lm extapic
bogomips     : 4020.64
TLB size     : 1024 4K pages
clflush size : 64
cache_alignm : 64
address sizes : 40 bits physical, 48 bits virtual
power management: ts fid vid ttp tm stc

processor       : 1
vendor_id     : AuthenticAMD
cpu family    : 15
model        : 75
model name    : AMD Athlon(tm) 64 X2 Dual Core Processor 3600+
stepping     : 2
cpu MHz      : 2015.986
cache size   : 256 KB
fpu          : yes
fpu_exception : yes
cpuid level  : 1
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36

```

Рис. 4.13. Информация о процессоре

```

$ modinfo reiserfs

modinfo reiserfs

filename:      /lib/modules/2.6.32-21-generic/kernel/fs/rei-
serfs/reiserfs.ko

license:      GPL

author:       Hans Reiser <reiser@namesys.com>

description:  ReiserFS journaled filesystem

srcversion:   186997D00EA2E7DCB528AE4A

depends:

vermagic:     2.6.32-21-generic SMP mod_unload modversions

```

Назначение модуля можно прочитать в строке `description`, правда, информация приведена на английском, а в выводе некоторых модулей она еще и отсутствует.

## Глава 5

# Установка приложений в Linux Mint

Особенности программ Linux

Некоторые приложения, входящие в состав Linux Mint

Программы для просмотра видео

Музыкальные проигрыватели

Программы для записи дисков

Редактирование ID3-тегов

Видеоредактор OpenShot

Одна из первых проблем, с которой столкнется любой пользователь, установивший Linux Mint, — нехватка определенных программ. Причем проблема двоякого плана: как установить и что, собственно, устанавливать. И здесь выяснится, что просто скачать файл и дважды щелкнуть на нем кнопкой мыши в большинстве случаев недостаточно. Какой файл качать? Что нужно сделать, чтобы программа установилась и работала? Как выбрать программу? Эти вопросы мы сейчас и разберем.

## Особенности программ Linux

В современных дистрибутивах Linux программы устанавливаются буквально одним-двумя щелчками кнопкой мыши и даже проще, чем в Windows. Пользователю в стандартных условиях даже не нужно задумываться, что и откуда берется. Но если появится необходимость в установке нестандартного решения, уже нужно представлять сам процесс. Поэтому вначале разберем основы.

Главный девиз Unix-систем звучит как KISS (акроним Keep It Simple Stupid — «Не усложняй»). Здесь не любят что-то делать повторно, все, что требуется, открыто и доступно в виде исходного кода, библиотек, реализующих некоторые функции, и т. д. Например, разработчик программы для Windows в большинстве случаев вынужден писать с нуля все необходимое для своей программы. Это происходит потому, что любая разработка защищена патентами и нужно либо платить, либо все делать самостоятельно. Естественно, на это уходит драгоценное время; разработчик вынужден разбираться в самых разнообразных вопросах либо привлекать еще людей, которые помогут добавить нужную функциональность, что приводит к удорожанию продукта. Кроме того, разработчику приходится повторять путь, пройденный другими, устранять ошибки, которые неизменно появляются в любом случае. Но зато программа устанавливается двойным щелчком кнопкой мыши, так как все необходимые для работы компоненты уже включены в дистрибутив программы. В Linux процесс разработки выглядит совсем по-другому.

Программы, библиотеки и др. распространяются с исходными кодами под лицензией GNU GPL, которая ставит только одно условие: если при разработке вы используете программу с этой лицензией, то и полученный продукт также должен распространяться по лицензии GNU GPL. Такой подход программисту дает возможность быстро создать новый продукт, используя чужие наработки. Он использует уже готовые компоненты, не вникая в суть их работы, и создает новую программу. Каждый улучшает только свою часть программы. Например, в Linux очень много консольных утилит, начинающему пользователю тяжело освоить все параметры командной строки, ему удобнее, чтобы программа имела понятный и легкий в освоении графический интерфейс. Что делает программист в Linux? Он не создает новую программу, он просто пишет графическую надстройку ко всем нужным консольным утилитам, которая скрывает от пользователя особенности работы в консоли, часто

добавляя новую функциональность. Такие программы называются фронт-эндом. Например, популярный консольный видеопроигрыватель MPlayer имеет несколько графических надстроек — Kplayer, KMplayer, smplayer, Freevo и др.

Но в итоге, чтобы программа заработала, необходимо установить и все остальные компоненты и библиотеки, которые она использует. Такие пакеты называются зависимостями. Например, смотрим зависимости программы для записи дисков Brasero (рис. 5.1).



```

$ sudo apt-cache depends brasero
brasero
Зависит: libappindicator0
Зависит: libatk1.0-0
Зависит: libbeagle1
Зависит: libbrasero-media0
Зависит: libc6
Зависит: libcairo2
Зависит: libcanberra-gtk0
Зависит: libcanberra0
Зависит: libdbus-1-3
Зависит: libdbus-glib-1-2
Зависит: libdbusmenu-glib1
Зависит: libfontconfig1
Зависит: libfreetype0
Зависит: libgconf2-4
Зависит: libglib2.0-0
Зависит: libgststreamer-plugins-base0.10-0
Зависит: libgststreamer0.10-0
Зависит: libgtk2.0-0
Зависит: libice0
Зависит: liblaunchpad-integration1
Зависит: libnautilus-extension1
Зависит: libpango1.0-0
Зависит: libsmb
Зависит: libtotem-plparser17
Зависит: libunique-1.0-0
Зависит: libxml2
Зависит: zlib1g
Зависит: gstreamer0.10-plugins-base
Зависит: gnome-icon-theme
Зависит: gvfs
Зависит: wofia
Зависит: genisoimage
Зависит: dvd+rw-tools
Зависит: brasero-common
Зависит: brasero-common
Предлагает: gstreamer0.10-plugins-bad
Предлагает: gstreamer0.10-plugins-ugly
Предлагает: gstreamer0.10-fluendo-mp3
  
```

Рис. 5.1. Список зависимостей программы Brasero

Некоторые из них, вероятно, уже установлены в системе, но, если не хватит хотя бы одного, установить программу будет проблематично либо после установки она будет иметь неполную функциональность. Нужно отметить, что зависимости могут быть двух видов:

- жесткая (hard) — обязательная, без нее программа, скорее всего, работать не будет;
- мягкая (soft) — рекомендуемая, без нее программа работать будет, но, установив предлагаемый пакет, вы получите дополнительную функциональность в программе (например, работу с некоторым форматом файлов для проигрывателей).

Кроме того, в soft-зависимости может быть указана документация или дополнительные конфигурационные файлы.

Именно поэтому часто, скачав всего один установочный файл (в терминологии Linux — пакет), установить программу нельзя, пользователь получает целый ряд сообщений об ошибке.

Но есть и другая проблема, о которой необходимо знать. В отличие от Windows, релизы которой можно пересчитать на пальцах одной руки, дистрибутивов Linux не просто много, а очень много. Отличаются они не только названием, ведь по сути Linux — всего лишь ядро, разрабатываемое под руководством Линуса Торвальдса, все остальное — это что-то вроде набора «Сделай сам». Данные наборы часто несовместимы не только между собой, но для каждой новой версии одного и того же дистрибутива потребуется специально собранный пакет. Но не все так плохо, поскольку практически во всех современных дистрибутивах используются пакетные системы, скрывающие особенности внутреннего устройства от пользователя, и установка производится одним щелчком кнопкой мыши, просто об этом нужно знать.

## Пакетные системы

Рассказ о системах управления пакетами может занять не один десяток страниц, сегодня существует около 20 различных систем с вариантами. Практически каждый дистрибутив имеет свои особенности, здесь исключений даже больше, чем правил. Это, конечно, вносит дополнительную путаницу и никак не способствует увеличению популярности Linux, но обычно достаточно просто выбрать «свой» дистрибутив и освоить принятую в нем систему установки. Кроме этого дистрибутив, отпочковавшийся от родительского, как правило, поддерживает пакеты родителя, хотя бы по той причине, что база популярных дистрибутивов — Linux Debian, Ubuntu, RedHat/Fedora, Slackware и др. — содержит не одну тысячу пакетов, не использовать которые было бы большой ошибкой.

Собственно средства управления пакетами можно разделить на две категории:

- менеджеры пакетов, непосредственно предназначенные для управления прекомпилированными программами;
- системы управления репозиториями пакетов, являющиеся надстройкой над пакетными менеджерами и позволяющие автоматизировать установку, обновление и удаление программ, сборку пакетов.

Менеджеры пакетов в первую очередь отличаются форматами пакетов. Издавна в Linux существовало три основных формата пакетов:

- deb — формат пакетов, появившийся в дистрибутиве Debian и производных, самым известным из которых является Ubuntu, именно он используется в Linux Mint и будет нас интересовать в дальнейшем;

- `rpm` (Red Hat Packages Manager) — формат, изначально принятый в дистрибутиве RedHat, сегодня применяется в его многочисленных потомках и иных дистрибутивах (Fedora, Mandriva, ALTLinux и др.);
- `tar.gz` — формат пакетов, появившийся в дистрибутиве Slackware.

Сегодня этот список больше. Первые два типа содержат информацию, в которой указаны зависимости этого пакета. Пакеты `tar.gz` (по сути, это обычные архивы) таких данных не содержат, поэтому пользователи Slackware вынуждены самостоятельно заботиться о зависимостях. В принципе, отсутствие информации внутри пакета о зависимостях абсолютно не препятствует контролю над ними. Данную функцию можно без проблем возложить на внешние программы, что с успехом и делается, а управление такими пакетами часто оказывается даже более гибким, чем пакетами, в которых такая информация имеется.

Для управления каждым типом пакетов используется своя программа — менеджер пакетов. При установке с их помощью программа самостоятельно создаст необходимые каталоги, распределит по ним файлы, создаст ссылки, настроит конфигурационные файлы, а в некоторых случаях и скачает все зависимости. Кроме этого менеджеры пакетов используются не только для установки нового пакета, но и для обновления, получения списка установленных пакетов, информации о пакетах и для деинсталляции пакетов, если в них отпала необходимость.

Теперь разберем все эти моменты подробнее.

## Установка пакетов

Как говорилось выше, в Linux Mint используются пакеты с расширением `deb`, для работы с которыми используется консольная утилита `dpkg`, входящая в состав дистрибутива. Это самый нижний уровень, и большинству пользователей, устанавливающих программы с репозитариев пакетов (о них ниже), скорее всего, не придется пользоваться услугами `dpkg`. Но такая необходимость все же может возникнуть, например, если в репозитарии пакетов доступна устаревшая версия программы или нужной программы нет (новая, не проверена, исключена по разным соображениям и т. д.).

Сам пакет имеет такой формат:

**`gnome-mplayer-0.9.4-i_amd64.deb`**,

где **`gnome-mplayer`** — название программы; **`0.9.4-1`** — версия программы; **`amd64`** — указывает, под какой тип процессора собран этот пакет, в данном случае — для 64-битных систем, для 32-битных здесь может стоять `i386`, `i586` или `i686` в зависимости от уровня оптимизации под конкретный тип процессора.

Если в пакете находятся исходные тексты, которые затем нужно будет компилировать, то такой пакет содержит значение **src**. Пакет, который не содержит бинарных данных, то есть подходит для всех систем, может быть отмечен как **noarch**.

Учитывая, что deb-пакеты могут быть собраны для самых разных дистрибутивов, сборщик пакета часто указывает дистрибутив, для которого создан пакет. Поэтому еще одной составляющей может быть имя дистрибутива и даже его версия — ubuntu, debian и т. д.



#### ПРИМЕЧАНИЕ

Человека, который отвечает за сборку пакета, называют **maintainer**, в Интернете часто можно встретить это слово в русской транскрипции «майнтейнер» или «сборщику». Он полностью отвечает за то, что пакет протестирован и работоспособен и его установка не вызовет неприятностей. В процессе сборки он может использовать патчи, изменяющие функциональность или устраняющие найденные ошибки. Но таких людей мало, и все пакеты отслеживать трудно, поэтому в проекты постоянно привлекают добровольцев.

Если такого поля нет, то, возможно, пакет универсален и подходит для всех дистрибутивов, использующих **rpm**.

Чтобы установить deb-пакет, следует использовать ключ **-i**:

```
$ sudo dpkg -i package_name.deb
```

Часто пользователь получает нужные пакеты со всеми зависимостями на CD- или DVD-диске, прилагаемом к какому-то журналу. Установить их все можно, указав ключ **-R** (**--recursive**), в качестве параметра используется каталог, в котором находятся все файлы:

```
$ sudo dpkg -R /usr/local/src
```

Хотя возможен и другой вариант — использование шаблонов регулярных выражений:

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

Теперь будут установлены все deb-пакеты, находящиеся в текущем каталоге.



#### ПРИМЕЧАНИЕ

Чтобы получить информацию обо всех параметрах любой консольной команды Linux, используйте ключ **--help**.

Если все не нужны, то можно использовать другой шаблон. Например, для установки всех пакетов, содержащих в имени слово **mplayer**, используем следующий шаблон:

```
$ sudo dpkg -i *mplayer*.deb
```

Для удаления пакета `dpkg` вызываем с ключом `-r` (или `--remove`), просто указав имя пакета в качестве последнего аргумента (рис. 5.2), но в этом случае все конфигурационные файлы будут сохранены.



```

Terminal
ubuntu@ubuntu-desktop:~$ sudo dpkg -r xchat
(Чтение базы данных ... на данный момент установлено 118308 файлов и каталогов.)
Удаляется пакет xchat ...
Обрабатываются триггеры для menu ...
Обрабатываются триггеры для map-db ...
Обрабатываются триггеры для desktop-file-utils ...
Обрабатываются триггеры для python-gmenu ...
Rebuilding /usr/share/applications/desktop.ru_RU.utf8.cache...
Обрабатываются триггеры для python-support ...
  
```

Рис. 5.2. Удаление пакета

В соответствии с Debian Policy Manual (в котором говорится, что «в списке рекомендованных пакетов следует указывать пакеты, устанавливаемые вместе с основным пакетом всегда, кроме случаев нетипичной установки») система управления пакетами теперь по умолчанию устанавливает не только пакеты, указанные в списке зависимостей основного пакета, но и рекомендованные. Если вы хотите избежать этого для конкретных пакетов, используйте `apt-get --no-install-recommends`; если вы хотите никогда не устанавливать рекомендованные пакеты, добавьте `APT::Install-Recommends "false";` в `/etc/apt/apt.conf`. Учтите, что в результате этого в некоторых программах могут быть доступны не все возможности.

#### ВНИМАНИЕ



При удалении пакета обычно подтверждение не запрашивается, то есть считается, что если пользователь ввел такую команду, то он знает, что делает. Будьте внимательны!

Чтобы удалить пакет полностью, вместе с конфигурационными файлами, используем ключ `-P`:

```
$ sudo dpkg -P xchat
```

Дополнительные ключи `--force` и `--ignore-depends` помогут проигнорировать проблемы, возникшие при установке или удалении пакета, например, когда пакет не может быть установлен из-за неудовлетворенных зависимостей. Но при применении этих ключей вся ответственность о стабильной работе программы или системы лежит полностью на вас.

Чтобы получить список установленных пакетов, используем ключ `-l` (`--list`) (рис. 5.3).

Кроме названия пакета, версии программы и описания в первой позиции указывается статус. В большинстве случаев мы получаем состояние `installed`, означающее,

что пакет распакован и правильно настроен. Иногда попадают пакеты config-files, когда сама программа удалена и в системе остались лишь конфигурационные файлы. Остальные параметры смотрите в man-документации, которую можно получить, введя в консоли `man dpkg`.

Поскольку вывод обычно большой, то результат сохраняют в файл для более комфортного просмотра:

```
$ sudo dpkg -l > install.txt
```

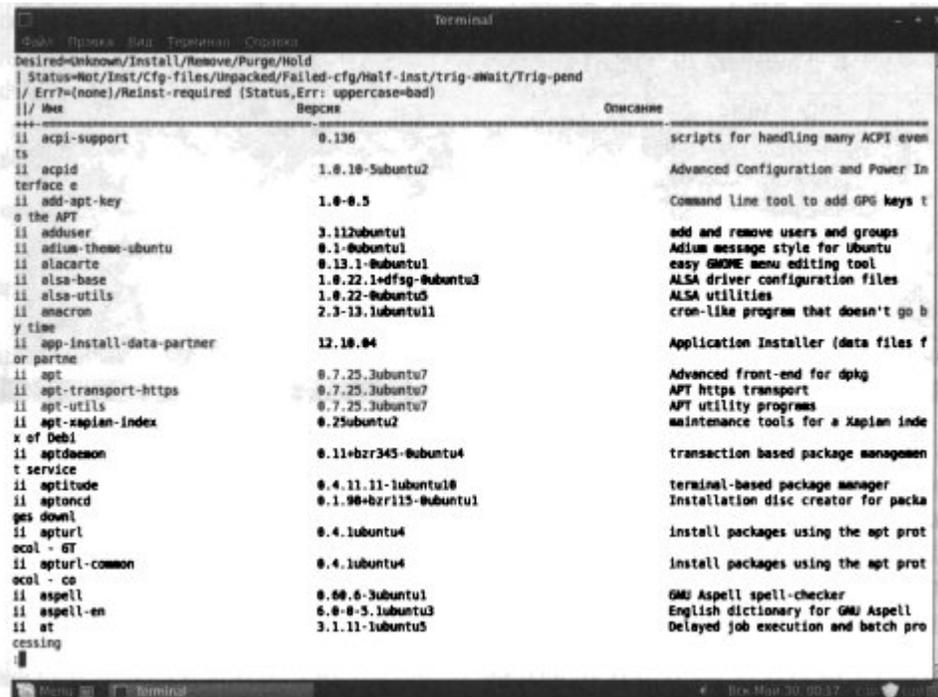
Теперь в файле содержится список всех программ. Чтобы вывод происходил по-экранно, нужно использовать программу `less`:

```
$ sudo dpkg -l | less
```

В результате на экран будет выведен первый список пакетов. Используя стрелки на клавиатуре, можно перемещаться вверх-вниз в поисках нужного.

Кроме того, можно использовать фильтрацию вывода при помощи консольной утилиты `grep`. Например, отберем все установленные программы, в названии и описании которых имеется слово `video`:

```
$ sudo dpkg -l | grep -i video
```



```
Terminal
Файл Правка Вид Терминал Справка
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-Inst/Trig-await/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
++-----+
||/ Name                               Версия                               Описание
+-----+
ii acpi-support                         0.136                               scripts for handling many ACPI even
ts
ii acpid                                 1.0.10-Subuntu2                     Advanced Configuration and Power In
terface e
ii add-apt-key                           1.0-0.5                              Command line tool to add GPG keys t
o the APT
ii adduser                               3.112ubuntu1                        add and remove users and groups
ii adobe-fonts-ubuntu                    0.1-0ubuntu1                        Adobe message style for Ubuntu
ii ataccart                              0.13.1-0ubuntu1                     easy GNOME menu editing tool
ii alsa-base                             1.0.22.1+dfsg-0ubuntu3             ALSA driver configuration files
ii alsa-utils                            1.0.22-0ubuntu5                     ALSA utilities
ii anacron                              2.3-13.1ubuntu11                    cron-like program that doesn't go b
y time
ii app-install-data-partner              12.10.04                             Application Installer (data files f
or partne
or partne
ii apt                                   0.7.25.3ubuntu7                     Advanced front-end for dpkg
ii apt-transport-https                  0.7.25.3ubuntu7                     APT https transport
ii apt-utils                            0.7.25.3ubuntu7                     APT utility programs
ii apt-xapian-index                     0.75ubuntu2                          maintenance tools for a Xapian inde
x of Debi
ii aptdaemon                            0.11+bzr345-0ubuntu4                transaction based package managemen
t service
ii aptitude                             0.4.11.11-1ubuntu10                 terminal-based package manager
ii aptoncd                               0.1.90+bzr115-0ubuntu1              Installation disc creator for packa
ges downl
ii apturl                                0.4.1ubuntu4                         install packages using the apt prot
ocol - GT
ii apturl-common                        0.4.1ubuntu4                         install packages using the apt prot
ocol - co
ii aspell                               0.60.6-3ubuntu1                     GNU Aspell spell-checker
ii aspell-en                             6.0-0-5.1ubuntu3                    English dictionary for GNU Aspell
ii at                                    3.1.11-1ubuntu5                      Delayed job execution and batch pro
cessing
ii
```

Рис. 5.3. Просмотр списка пакетов

Во время установки пакет не только распаковывается, но и настраивается, иногда возникает необходимость перенастроить пакет. Это можно сделать при помощи команды `dpkg --configure название_пакета` или утилиты `dpkg-reconfigure`.



#### ПРИМЕЧАНИЕ

Многие операции можно проделать с помощью файлового менеджера Midnight Commander (рис. 5.4), который устанавливается командой `sudo apt-get install mc`. Так, установив курсор на файле и нажав F3 (просмотр), смотрим информацию о пакете, а зайдя в пакет, нажав клавишу Enter и выбрав файл INSTALL, можно установить выбранный пакет.

```

mc [grinder@grinder-desktop]:~/Рабочий стол
Файл  Правка  Вид  Терминал  Справка
Файл: bus 2.5.2-1.all.deb  Строка 1  Позиция 8  6705 байт
новый пакет Debian, версия 2.0
размер 85206 байт(а); управление архива длиной 3558 байт(а).
 613 байт(а),   39 строк   control
 2208 байт(а),   34 строк   md5sums
Package: bus
Version: 2.5.2-1
Architecture: all
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Fabio Marzocca <thesaltydog@gmail.com>
Installed-Size: 526
Depends: menu, sysv-rc, perl, libgtk2-perl (>= 1:1.100-1), libglib-perl (>= 1:1.100-1), liblocale-gettext-perl
Conflicts: file-rc
Section: admin
Priority: optional
Homepage: http://www.marzocca.net/linux/bus.html
Description: graphical runlevel editor
  Boot-up Manager is a graphical tool to allow easy configuration
  of all services in user and system runlevels, as far as changing
  Start/Stop services priority.

#ls -l /usr/share/pixmaps/bus.png
-rw-r--r-- root/root          9 2009-08-26 23:51 /usr/share/pixmaps/bus.png
#ls -l /usr/share/pixmaps/bus_serv_bsh.png
-rw-r--r-- root/root       784 2009-08-26 23:51 /usr/share/pixmaps/bus_serv_bsh.png
#ls -l /usr/share/pixmaps/bus_serv_none.png
-rw-r--r-- root/root       393 2009-08-26 23:51 /usr/share/pixmaps/bus_serv_none.png
#ls -l /usr/share/pixmaps/bus_serv_ofl.png
-rw-r--r-- root/root      1333 2009-08-26 23:51 /usr/share/pixmaps/bus_serv_ofl.png
#ls -l /usr/share/pixmaps/bus_serv_on.png
-rw-r--r-- root/root      1333 2009-08-26 23:51 /usr/share/pixmaps/bus_serv_on.png
#ls -l /usr/share/pixmaps/bus_serv.png
-rw-r--r-- root/root     10611 2009-08-26 23:51 /usr/share/pixmaps/bus_serv.png
#ls -l /usr/share/locale/
-rw-r--r-- root/root          0 2009-08-26 23:51 /usr/share/locale/
#ls -l /usr/share/locale/it/
-rw-r--r-- root/root          0 2009-08-26 23:51 /usr/share/locale/it/
#ls -l /usr/share/locale/it/LC_MESSAGES/
-rw-r--r-- root/root      12165 2009-08-26 23:51 /usr/share/locale/it/LC_MESSAGES/bus.mo
#ls -l /usr/share/locale/de/
-rw-r--r-- root/root          0 2009-08-26 23:51 /usr/share/locale/de/
#ls -l /usr/share/locale/de/LC_MESSAGES/
-rw-r--r-- root/root          0 2009-08-26 23:51 /usr/share/locale/de/LC_MESSAGES/bus.mo
#ls -l /usr/share/locale/es/
-rw-r--r-- root/root          0 2009-08-26 23:51 /usr/share/locale/es/
1  Папки  2  Папки  3  Папки  4  Тек  5  Строки  6  Поиск  7  Поиск  8  Поиск  9  Поиск  10  Поиск

```

Рис. 5.4. Просмотр информации о пакете в Midnight Commander

К `dpkg` написан и графический фронт-энд GDebi, который входит в состав Linux Mint. Его можно вызвать, выбрав пункт **Открыть с помощью менеджера пакетов GDebi**. Откроется окно программы установки (рис. 5.5). Нажав кнопку, размещенную справа сверху, можно установить или переустановить пакет.

В разных вкладках доступны: описания, подробности о пакете и список файлов, которые он содержит. Удалить программу в Linux Mint еще проще, достаточно выбрать ее в меню, вызвать контекстное меню и выбрать в нем пункт **Удалить**.

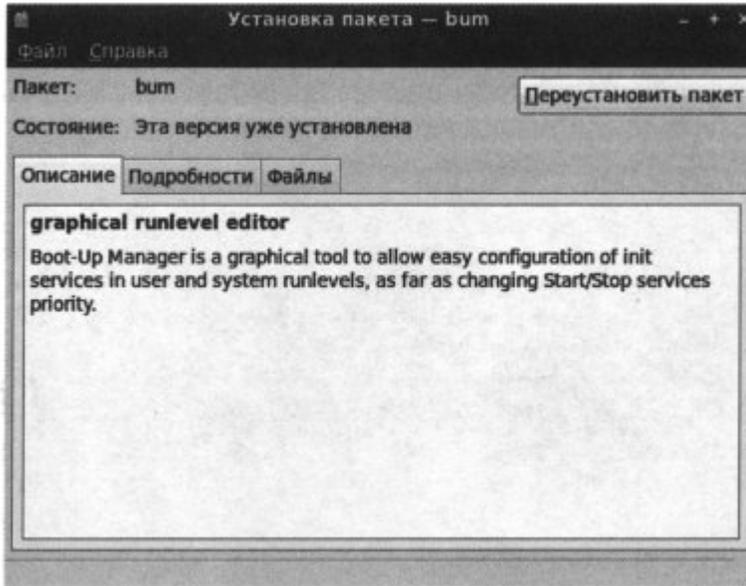


Рис. 5.5. Установка пакета в Linux Mint при помощи GDebi

Как видите, все очень просто, хотя все равно неудобно. Ведь если пакет требует для своей работы другие пакеты, их придется также предварительно скачать и устанавливать одним списком. Но что делать, если таких пакетов десятков, которые в свою очередь потребуют еще несколько пакетов? В таком случае установка может превратиться в ад. Кроме того, очень неудобно искать пакеты для совместимости по всем сайтам Интернета.

Чтобы выйти из такой ситуации, разработчики пошли дальше, собрав все пакеты в одном месте, при этом специальные программы отслеживают все зависимости между ними, которые также устанавливаются автоматически. В итоге установка приложений в Linux даже проще, чем в Windows. Но установка индивидуальных пакетов сегодня применяется в тех случаях, когда нужного пакета нет в репозитории и для экономии трафика, ведь когда в наличии уже есть нужный пакет, зачем скачивать его повторно?

К слову, к `dpkg` доступна интерактивная оболочка, обладающая функциями контроля зависимостей, — `dselect` (рис. 5.6), по умолчанию она в системе не устанавливается, но это легко решить.

```
$ sudo apt-get install dselect
```

Далее запускаем утилиту с правами администратора:

```
$ sudo dselect
```



Функционально АРТ состоит из нескольких утилит, наиболее часто приходится использовать две — `apt-get` и `apt-cache`. Утилиты из состава АРТ очень просты в использовании. Перед началом работы следует обязательно обновить список пакетов в базе данных утилиты, делается это командой

```
$ sudo apt-get update
```

Некоторое время ждем, пока будут загружаться списки пакетов, после чего можно устанавливать и обновлять программы. Обновление системы выполняется также одной командой

```
$ sudo apt-get upgrade
```

либо более «интеллектуальной»

```
$ sudo apt-get dist-upgrade
```

Чтобы установить любой пакет, достаточно ввести

```
$ sudo apt-get install имя_пакета
```

Если пакет необходимо обновить, вместо `install` используем `update`. При наличии соединения с Интернетом все остальное утилита сделает сама. Естественно, чтобы устанавливать, нужно знать, что устанавливать, АРТ предлагает свой путь. Так, чтобы найти программу, проигрывающую MP3-файлы, достаточно использовать команду `apt-cache`, которая применяется для работы с данными, хранящимися в локальном кэше пакетов:

```
$ sudo apt-cache search mp3
```

В результате будет выведен список всех соответствующих пакетов. Кроме этого в консоли при попытке запустить программу, которая еще не установлена в системе, выводится подсказка, говорящая о том, какую команду нужно выполнить, чтобы установить программу (рис. 5.7).



Рис. 5.7. Подсказка, выдаваемая при запуске неустановленной программы



## Утилита aptitude

Разработчики дистрибутивов Debian и Ubuntu рекомендуют для установки приложений использовать утилиту aptitude, которая также идет в комплекте Linux Mint. Утилита aptitude является надстройкой над dpkg и представляет собой мощное средство контроля над пакетами, работающее в командном и интерактивном режиме. В некоторых обзорах ее считают надстройкой над APT, но на самом деле это не так, просто они используют одну и ту же библиотеку libapt.

Нельзя также считать aptitude полноценной заменой APT, хотя бы потому, что последний обеспечивает полный набор операций — от установки единичного пакета до полной пересборки системы из исходных текстов. Возможности aptitude ограничены установкой, обновлением и удалением программ и получением информации о пакетах. Но эти операции она делает настолько хорошо, что многие пользователи предпочитают именно aptitude.

Чтобы запустить aptitude в интерактивном режиме, достаточно просто набрать в строке терминала команду без аргументов:

```
$ sudo aptitude
```

Появится окно с псевдографическим интерфейсом (рис. 5.9), в котором, передвигаясь по пунктам, можно установить или обновить пакеты и просмотреть список устанавливаемых программ.

Рабочее пространство программы разбито на три части: вверху находится строка меню со справкой по популярным горячим клавишам, в середине — список категорий, внизу — поле с описанием пакетов и категорий. Управление производится при помощи клавиш на клавиатуре, полный список можно получить, нажав клавишу со знаком вопроса (рис. 5.10). Например, чтобы установить пакет, следует его выбрать и нажать клавишу +, чтобы удалить пакет — клавишу -, чтобы фиксировать в текущей версии (запретить обновление) — клавишу =, перемещение производится при помощи клавиш с изображением стрелок. После выбора пакетов в полосе вверху будет показана информация по количеству загружаемых данных и справка по комбинации клавиш для дальнейших операций. Принцип прост: вначале отмечаем при помощи + и - все пакеты и группы пакетов, а затем запускаем выполнение задания, нажав клавишу **g**.

В самом верхнем окне aptitude предлагает на выбор семь категорий пакетов. Поскольку они не совпадают с классификацией пакетов в репозитории, скажу несколько слов о них:

- **Security Updates** — доступные обновления безопасности;
- **Обновляемые пакеты** — пакеты, для которых доступны новые версии;

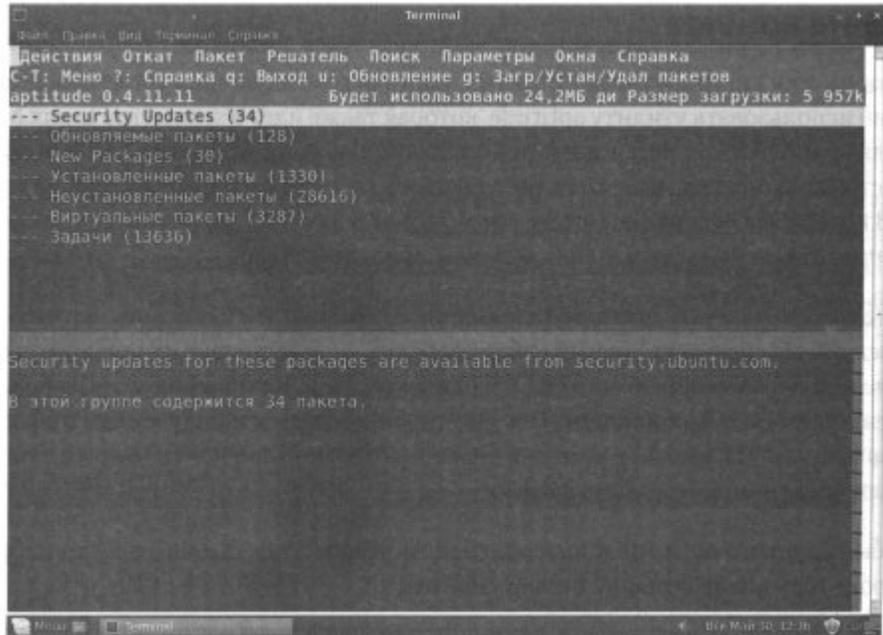


Рис. 5.9. Окно интерактивного режима aptitude

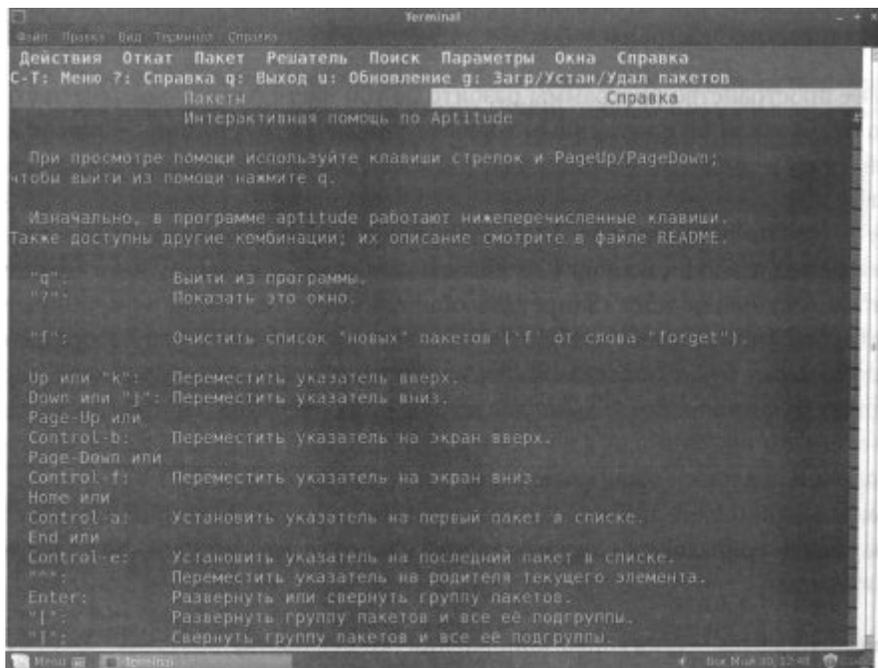


Рис. 5.10. Справка по комбинациям клавиш aptitude

- **New packages** — новые пакеты программ, которые появились в репозитории Ubuntu, чтобы очистить список, следует нажать клавишу f;
- **Установленные пакеты** — пакеты, установленные в системе;
- **Неустановленные пакеты** — пакеты, доступные в репозитории, но не установленные в дистрибутиве;
- **Виртуальные пакеты** — на самом деле таких пакетов не существует, они указывают на другие пакеты;
- **Задачи** — группы пакетов, позволяющие быстро настроить систему под определенные нужды.

Возле каждой категории в скобках указано количество пакетов, в каждой категории находятся секции по назначению (**Рабочий стол**, программы администрирования, мультимедиа и т. д.), затем — по репозиториям (о них ниже).

Все в принципе понятно, объяснение потребуют, наверное, два последних пункта. В репозитории Linux очень много программ, которые к тому же могут иметь и разные версии. Чтобы упростить пользователю поиск и установку программ, некоторые из них используют короткое имя, которое легко запоминается. Вот такое имя и является виртуальным пакетом, при вводе которого будет установлен связанный с ним пакет. Задачи представляют собой виртуальные пакеты, только более высокого уровня. Например, чтобы настроить сервер или **Рабочий стол** с другим оконным менеджером, необходимо установить большое количество пакетов. Их имена запомнить нереально, наиболее часто встречающиеся задачи собраны в такие виртуальные пакеты. Например, чтобы установить **Рабочий стол** Ubuntu достаточно установить всего один пакет — `ubuntu-desktop`, который в свою очередь установит более 900 связанных пакетов (рис. 5.11).

При выборе отдельного пакета в первой колонке показывается его статус. В первой позиции может стоять: `i` (`install`) — установленный пакет; `p` (`purge`) — неустановленный или не полностью удаленный пакет; `c` (`clean`) — удаленный пакет с сохранением конфигурационных файлов и `v` (`virtual`) — виртуальный пакет. Во второй позиции параметры обычно уточняют состояние пакета: `A` (`Auto`) — установленный как зависимость другого пакета (автоматически); `h` (`hold`) — нельзя обновить; `u` (`unpacked`) — пакет только распакован, но не установлен; `H` — установка пакета не завершена; `C` — пакет установлен, но не настроен; `B` (`bad`) — поврежденный пакет.

Интерактивный режим популярен у новичков, удобен при первоначальной настройке системы, поиске «битых» пакетов и других операциях, требующих наглядности. В повседневной эксплуатации обычно используют командный режим, который более удобен, так как позволяет быстро произвести все необходимые операции, не тратя времени на поиск пакета в меню. Принцип работы с `aptitude` в этом режим похож на работу с `APT`. Вначале обновляем список пакетов, ключ здесь аналогичен `apt-get`:

```
$ sudo aptitude update
```



```

grinder@grinder-desktop ~ $ aptitude search video
v  girl1.0-gstvideo-0.10 -
v  gstreamer0.10-videosink -
v  gstreamer0.10-videosource -
p  libvideo-capture-v4l-perl - Perl interface to the Video4linux framgrab
p  libvideo-fourcc-info-perl - Perl module to find information about codec
p  libvideo-frequencies-perl - Many, many frequency constants and lists
p  libvideo-ivtv-perl - Perl extension for using V4l2 in the ivtv p
p  mythvideo - A generic video player frontend module for
p  nicovideo-dl - Download videos from www.nicovideo.jp
p  oggvideotools - A toolbox for manipulating and creating Ogg
p  oggvideotools-dbg - A toolbox for manipulating and creating Ogg
p  ogmrip-video-copy - ogmrip plugin to extract DVD track
p  python-gnuradio-video-sdl - GNU Radio SDL Interface Library
v  python2.6-gnuradio-video-sdl -
p  ubuntustudio-video - Ubuntu Studio video Package
p  video-dvdrip-doc - Documentation for dvd::rip - dummy package
p  videocut - application for creating compositions of sc
p  videogen - Создает сроки modeline, используя параметры
p  videolan-doc - documentation for the VideoLAN streaming so
p  videoporama - Make and export image slideshows
p  videotrans - DVD authoring utilities
p  x11proto-video-dev - X11 Video extension wire protocol
v  xf86-video-driver-rival28 -
v  xserver-xorg-video-6 -
i  xserver-xorg-video-all - the X.Org X server -- output driver metapac
i  xserver-xorg-video-apm - X.Org X server -- APM display driver
i  xserver-xorg-video-ark - X.Org X server -- ark display driver
i  xserver-xorg-video-ati - X.Org X server -- AMD/ATI display driver wr
p  xserver-xorg-video-ati-dbg - X.Org X server -- AMD/ATI display driver wr
    
```

Рис. 5.12. Поиск программ в aptitude

```

grinde inder de ~ $ aptitude show videotrans
Пакет: videotrans
Состояние: не установлен
Версия: 1.6.0-0ubuntu5
Приоритет: необязательный
Раздел: multiverse/graphics
Сопровождающий: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Размер в распакованном виде: 344k
Зависимости: libc6 (>= 2.4), mplayer, mjpegtools, dvdauthor, ffmpeg, imagemagick
Описание: DVD authoring utilities
A collection of utilities and scripts designed to help with DVD authoring:
* movie-to-dvd - convert video to MPEG2 + MP2 or AC3
* movie-title - combine video and title sequence into a menu
* movie-make-title - create background video for DVD menu
* movie-make-title-simple - create background image+audio for DVD menu
* movie-rip-tv.com - gather information about TV series from tv.com
* movie-rip-epg.data - convert EPG to .info for movie-title
* movie-compare-dvd - verify if DVD was burned correctly
Сайт: http://videotrans.sourceforge.net

grinde ~ $ █
    
```

Рис. 5.13. Вывод информации о пакете

**ПРИМЕЧАНИЕ**

Чтобы aptitude не устанавливал пакетов с мягкими зависимостями, добавьте в команду параметр `-R`. Чтобы такая установка действовала постоянно, добавьте в файл `~/aptitude/config` строку `aptitude::Recommends-Important "false"`.

Переустановить пакет так же просто:

```
$ sudo aptitude reinstall название_пакета
```

Чтобы удалить ненужную программу, используем ключ **remove**:

```
$ sudo aptitude remove название_пакета
```

Конфигурационные файлы при этом будут оставлены, чтобы удалить пакет полностью, вместо `remove` следует ввести `purge`. Причем здесь опять же есть отличие aptitude от АРТ, вместе с пакетом будут удалены и все зависимости (которые помечаются буквой **a**), если, конечно, у зависимостей не будет других приложений, требующих их наличия. При этом две команды `markauto` и `unmarkauto` позволяют пользователю самостоятельно пометить или снять метку `auto` с пакетов, установленных автоматически.

**ПРИМЕЧАНИЕ**

Все команды aptitude можно посмотреть на справочной странице `man aptitude`.

## Графические утилиты для установки программ

В большинстве современных дистрибутивов Linux, ориентированных на применение на **Рабочих столах**, доступны утилиты с графическим интерфейсом, которые понятны пользователю и не требуют знания команд. В такой программе достаточно выбрать нужное приложение, и оно будет автоматически установлено. Linux Mint не исключение. Причем разработчики предложили две такие программы — Менеджер программ (`mintinstall`) и Менеджер пакетов (`synaptic`), ссылку для запуска которых можно найти в меню.

Программа собственной разработки `mintinstall` призвана упростить сам процесс установки приложений. Окно, появившееся после запуска, содержит список категорий программ (рис. 5.14).

Выбираем категорию, появляется список программ. Установленные программы помечены зеленым флажком. Разработчики ввели систему рейтингов, и все пользователи, которые установили ту или иную программу, могут выставить ей оценку, чтобы упростить выбор остальным. Программы, для которых уже выставлены оценки, идут первыми: чем больше баллов, тем выше в списке располагается программа. Выбрав программу, можем получить более подробную информацию о ней и почитать отзывы (рис. 5.15). К сожалению, описание большинства программ не переведено, поэтому желательно знать английский хотя бы в минимальном объеме.

Кроме того, показывается снимок окна работающей программы, что позволяет оценить ее внешний вид.

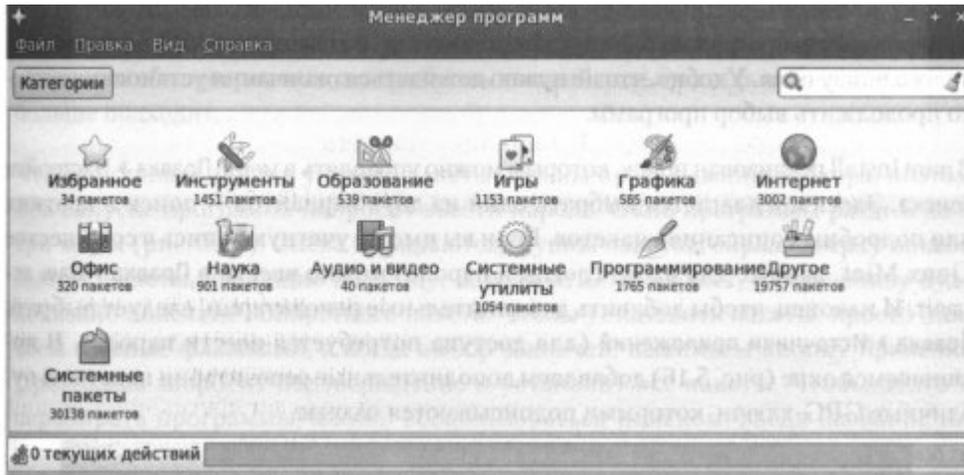


Рис. 5.14. Программа mintInstall

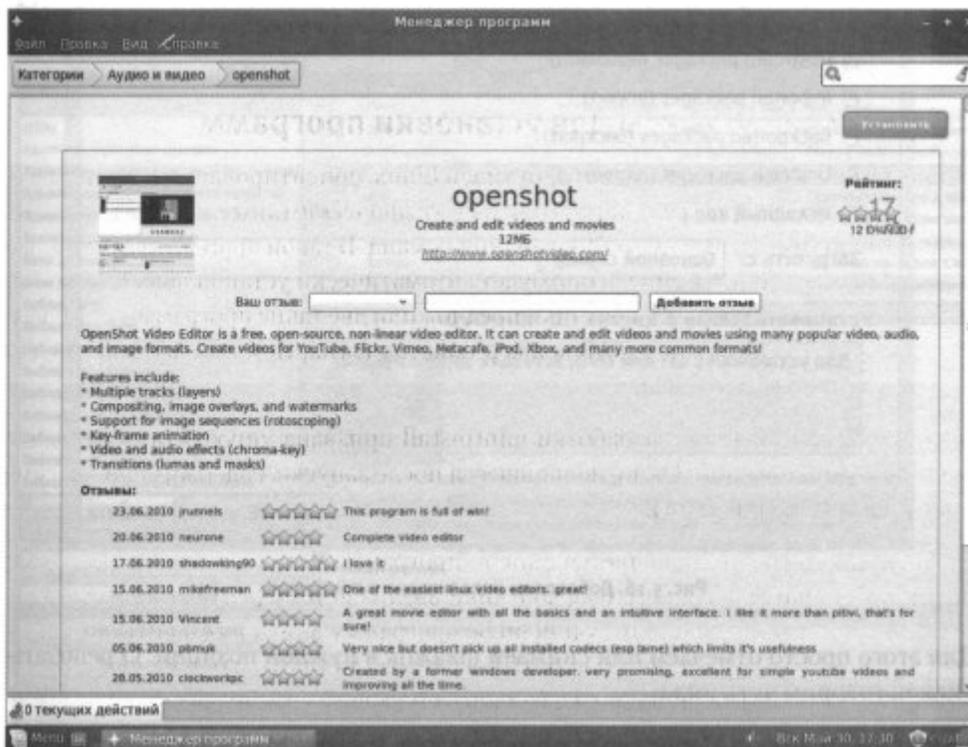


Рис. 5.15. Описание программы в окне Менеджера программ

Вы сами можете отправить свой отзыв и указать рейтинг, воспользовавшись полем вверху страницы. Чтобы установить выбранную программу, достаточно нажать кнопку **Установить**, расположенную вверху страницы, после чего потребуется ввести пароль. Все, программа будет устанавливаться, о ходе процесса будет говорить полоса внизу окна. Удобно, что не нужно дожидаться окончания установки и можно продолжить выбор программ.

В `mintinstall` реализован поиск, которым можно управлять в меню **Правка • Настройки поиска**. Здесь предлагается выбрать один из двух вариантов — поиск в кратких или подробных описаниях пакетов. Если вы имеете учетную запись в сообществе Linux Mint, данные для входа (логин и пароль) можно ввести в **Правка • Ваш аккаунт**. И наконец, чтобы добавить дополнительные репозитории, следует выбрать **Правка • Источники приложений** (для доступа потребуется ввести пароль). В появившемся окне (рис. 5.16) добавляем дополнительные репозитории пакетов и публичные GPG-ключи, которыми подписываются пакеты.

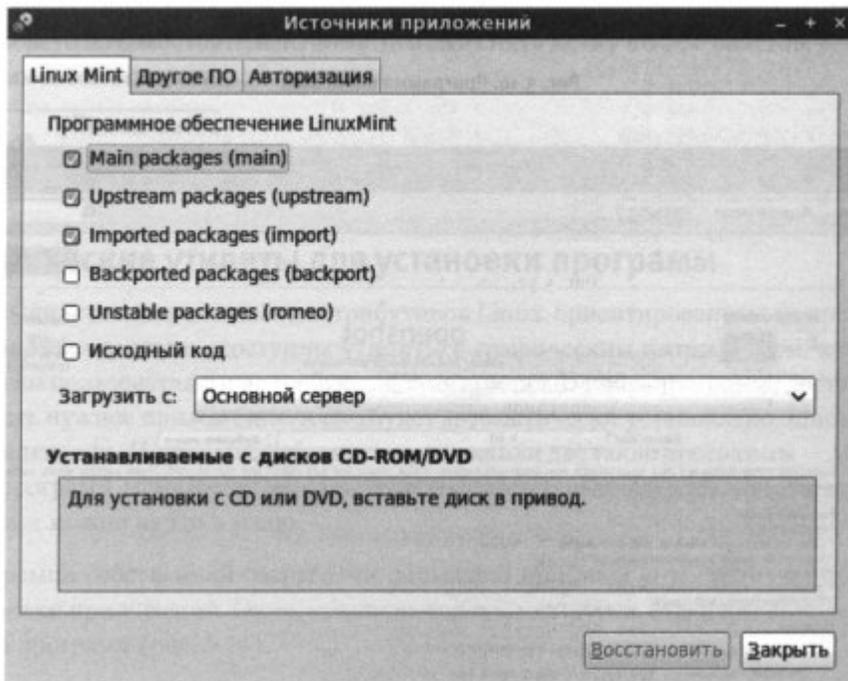


Рис. 5.16. Добавляем репозитории в `mintinstall`

Для этого просто отмечаем или снимаем флажок в нужной позиции. О репозиториях поговорим чуть ниже.

Вторая программа — Менеджер пакетов (`Synaptic Package Manager`) — это популярное средство управления пакетами в дистрибутивах, основанных на Debian, явля-

ется надстройкой над APT и dpkg. Synaptic использует библиотеки Gtk+ и после установки доступен в других дистрибутивах, в которых по умолчанию используется среда GNOME. Возможностей в Synaptic на порядок больше, чем в mintinstall, но последнему она немного уступает в удобстве. Именно поэтому разработчики включили обе программы в дистрибутив, каждый пользователь выберет то, что ему больше подходит.

Чтобы устанавливать и удалять пакеты, нужны права администратора, поэтому при запуске программа попросит ввести пароль. Окно программы разделено на три части (рис. 5.17). Слева выбираются группы пакетов, справа вверху показываются пакеты, входящие в группу, и, если отметить пакет, в окне внизу будет показано описание конкретного пакета. Чтобы установить пакеты, просто отмечаем нужные флажками, а когда выбор закончен, нажимаем кнопку **Применить**. Программа запросит подтверждение и установит все пакеты. Чтобы долго не перебирать программы, можно воспользоваться поиском, введя название или ключевое слово. Удобно, что результаты поиска текущего сеанса сохраняются и чтобы найти их, следует нажать кнопку **Результаты поиска**, расположенную слева внизу.

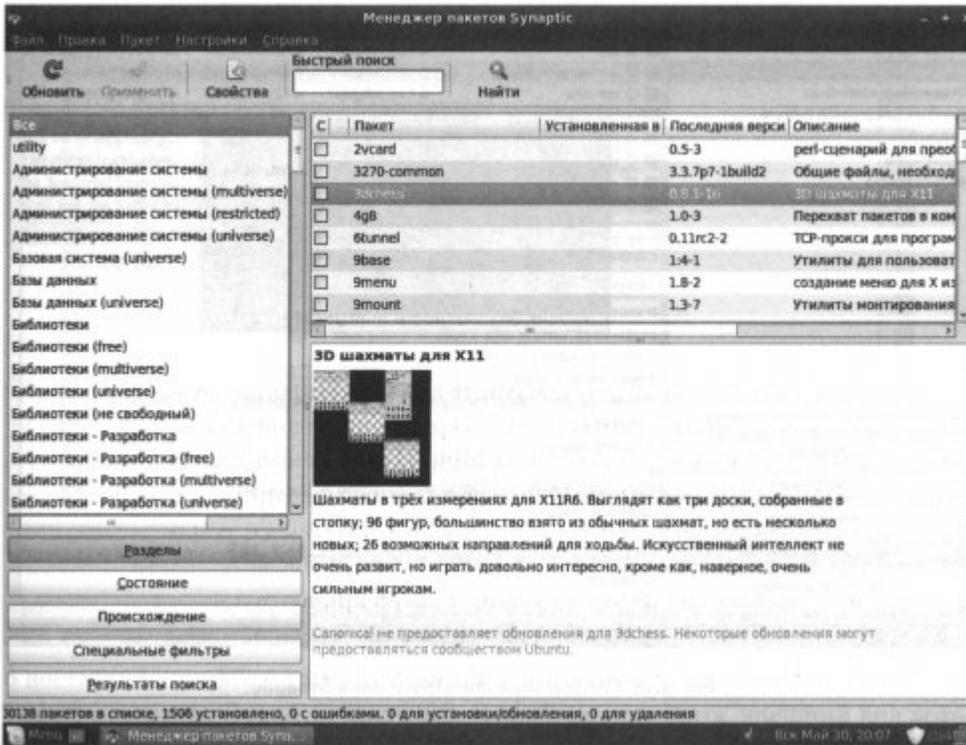


Рис. 5.17. Менеджер пакетов — Synaptic

Внизу слева находится еще несколько кнопок, которые отвечают за вывод групп пакетов. После запуска обычно нажата кнопка **Разделы**, и все пакеты в окне выше рассортированы по назначению (см. рис. 5.17) — администрирование, библиотеки, графика, мультимедиа, игры и т. д. Нажатие кнопки **Состояние** покажет пакеты в зависимости от того, установлены ли они, пакеты, которые нужно обновить. Выберите пункт **Все**, чтобы увидеть все пакеты. Если вы хотите увидеть, какие пакеты предоставляют репозитории, нажмите кнопку **Происхождение**.

Перед началом работы следует обновить список пакетов. Для этого нужно нажать кнопку **Обновить**. Все отмеченные пакеты можно найти в одноименном меню в **Специальные фильтры** (рис. 5.18). Здесь же расположены фильтры, позволяющие быстро отобразить обновляемые пакеты, с ошибками, без рекомендуемых и др.



#### ПРИМЕЧАНИЕ

Чтобы увидеть снимок окна выбранной программы, нажмите кнопку **Получить снимок экрана**.

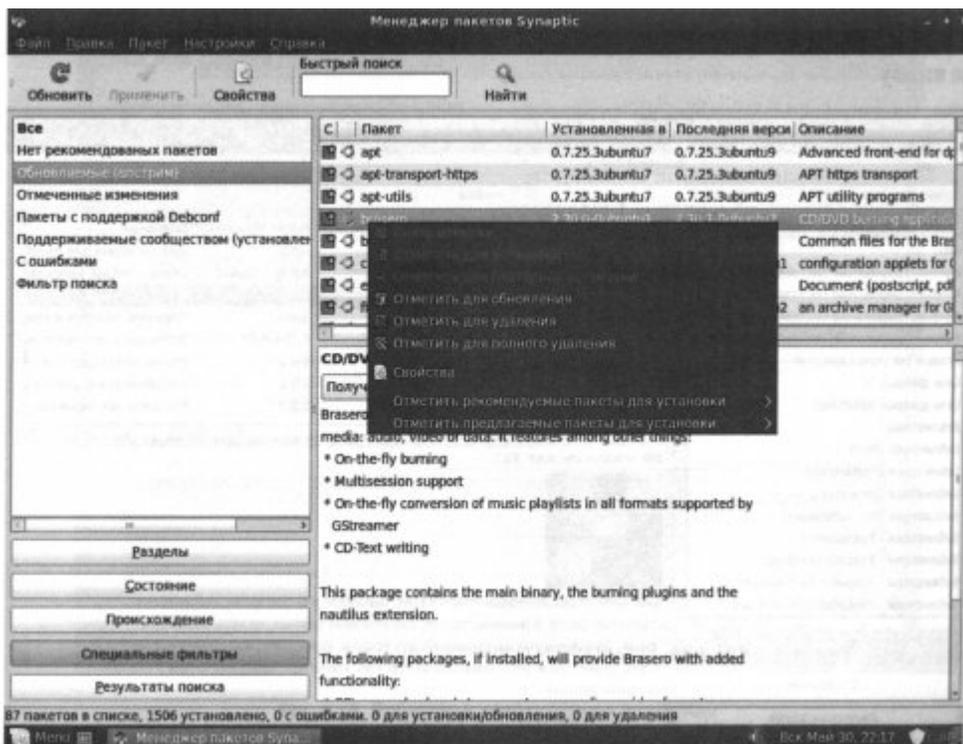


Рис. 5.18. Специальные фильтры и меню Synaptic

Чтобы удалить, обновить пакет и просмотреть его свойства, необходимо вызвать контекстное меню (см. рис. 5.18). Причем, чтобы удалить пакет вместе с конфигурационными файлами, следует выбрать вариант **Отметить для полного удаления** (ана-

лог **purge**). В контекстном меню **Пакет** доступны пункты, позволяющие заблокировать версию пакета (он не будет обновляться), назначить версию пакета (если они есть), которая будет установлена. Чтобы добавить CD-ROM с пакетами или исправить пакеты с ошибками, выбираем пункт **Правка**. В тех случаях, когда интернет-канал не позволяет скачивать пакеты, а только обновлять базу, можно сгенерировать скрипт закачки, который будет содержать все необходимые ссылки. Для этого выбираем **Файл • Создать сценарий закачки пакетов** и указываем, куда сохранить файл. Скачанные таким образом файлы можно установить при помощи **dpkg** или выбрав **Файл • Закачанные пакеты**.

Synaptic, так же как и **mintinstall**, позволяет отредактировать список репозитариев. Переходим в **Настройки • Репозитарии** и добавляем или удаляем нужные, просто установив или сняв флажок (рис. 5.19).

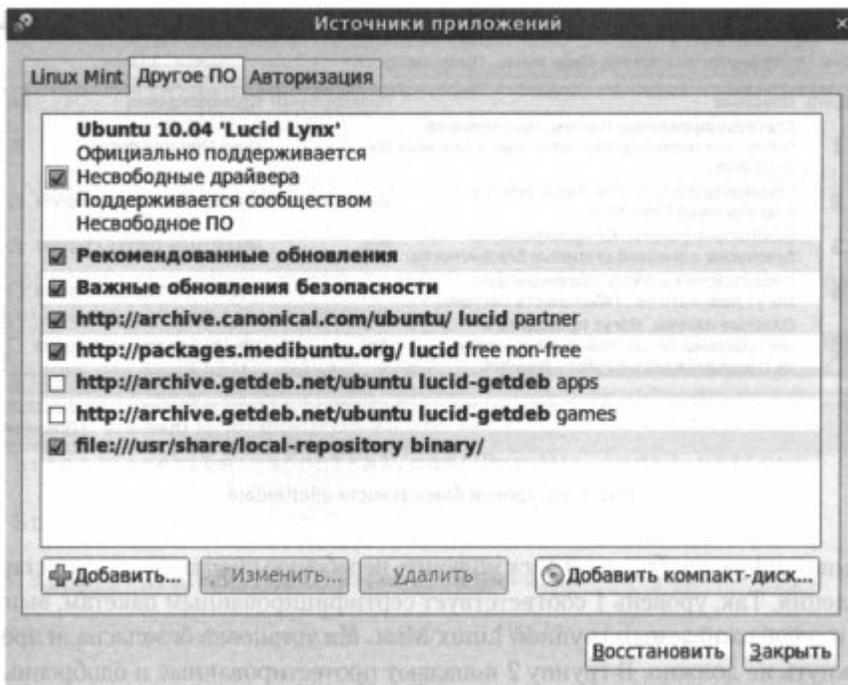


Рис. 5.19. Редактируем список репозитариев в Synaptic

Если репозитария нет в списке, нажимаем кнопку **Добавить** и вписываем адрес.

В поставке Linux Mint доступно еще одно приложение, которое связано с установкой программ. В панели задач видно изображение щита, внешний вид которого меняется в зависимости от состояния системы (просмотреть значения, можно, открыв **Правка • Параметры • Значки**). Если на щите выводится буква **I** на синем фоне, значит, доступны обновления, их количество обычно выводится

в виде всплывающего сообщения. Двойным щелчком мыши на значке можно запустить программу **mintUpdate**, предназначенную для обновления программ. После запуска программа соединяется с репозиториями и проверяет наличие обновлений, после чего показывает пользователю весь список. Достаточно нажать кнопку **Установить обновления**, и программа обновит отмеченные пакеты. Время автообновления списка пакетов по умолчанию составляет 15 минут, в большинстве ситуаций смысла в такой частой проверке нет. Установить другой промежуток времени можно в **Правка • Параметры • Автообновление**. Время регулируется в минутах, часах и днях, выберите оптимальное для себя значение.

При наличии программ, требующих обновления, **mintUpdate** сам предупредит вас об этом. Особенно ценно выглядит информация, предоставляемая **mintUpdate** при обновлении продукта. Существует пять уровней оповещения (рис. 5.20).

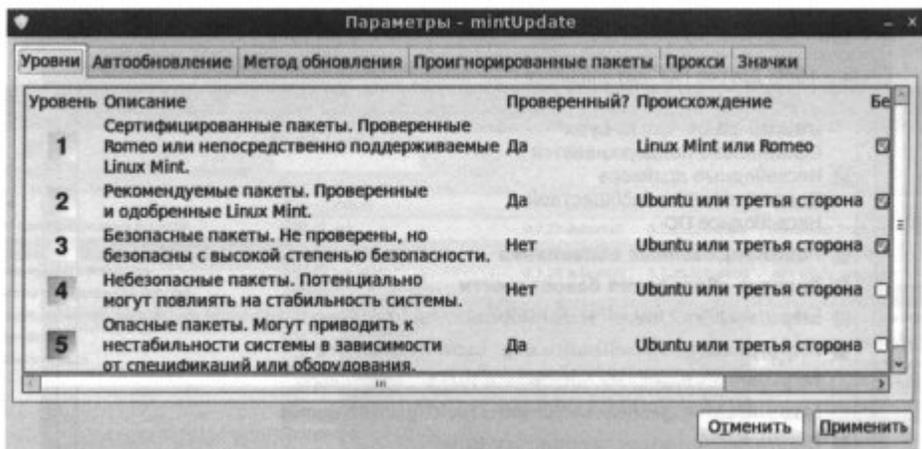


Рис. 5.20. Уровни безопасности **mintUpdate**

Чем выше цифра, тем больший риск получить нерабочую программу после установки обновления. Так, уровень 1 соответствует сертифицированным пакетам, выпускаемым и поддерживаемым группой Linux Mint. Их установка безопасна, и проблем возникнуть не должно. В группу 2 попадают протестированные и одобренные пакеты, выпускаемые разработчиками Ubuntu и третьими сторонами. В группу 3 входят безопасные пакеты Ubuntu и третьих сторон, которые не проверены разработчиками Linux Mint, но их установка не должна принести проблем. Рекомендуется устанавливать пакеты с уровнем риска, равным 3-му или меньше его, поэтому пакеты уровней 4 и 5 по умолчанию не выводятся в списке предложений. Пакеты уровней 4 и 5 могут повлиять на стабильную работу системы, поэтому устанавливать их рекомендуется пользователям, имеющим некоторый опыт работы в Linux. Пользователь еще на этапе обновления может принять правильное решение, следует ли вообще обновлять программу, и оценить связанный с этим риск.

Как видите, установить программу в Linux Mint на самом деле даже проще, чем в Windows, ведь не нужно ее даже искать в Интернете, все доступно в репозиториях, предлагаемых разработчиками и сторонними мантейнерами. По умолчанию список репозитория в Linux Mint можно назвать минимальным, в Интернете доступны дополнительные репозитории. Кроме этого отдельные проекты предлагают свои репозитории для Ubuntu и Debian, которые можно использовать и в Linux Mint. Поэтому давайте разберем, как можно управлять ими.

## Описание репозитория в Linux Mint

Пакеты в репозитории структурированы по назначению, платформе и прочим характеристикам. Список используемых дистрибутивов репозитория приведен в файле `/etc/apt/sources.list`. Заглянем внутрь.

```
$ cat /etc/apt/sources.list

deb http://packages.linuxmint.com/ isadora main upstream im-
port

deb http://archive.ubuntu.com/ubuntu/ lucid main restricted
universe multiverse

deb http://archive.ubuntu.com/ubuntu/ lucid-updates main re-
stricted universe multiverse

deb http://security.ubuntu.com/ubuntu/ lucid-security main re-
stricted universe multiverse

deb http://archive.canonical.com/ubuntu/ lucid partner

deb http://packages.medibuntu.org/ lucid free non-free

#deb http://archive.getdeb.net/ubuntu lucid-getdeb apps

#deb http://archive.getdeb.net/ubuntu lucid-getdeb games
```

Разберем теперь, что получилось. В файле описываются репозитории, находящиеся на различных узлах в Интернете. В первой строке идет тип пакета, если стоит `deb` — это указывает на то, что будут загружаться пакеты с уже скомпилированными программами. Если планируется самостоятельно компилировать программы, следует использовать `deb-src`, который отвечает за загрузку пакетов с исходными текстами. В нашем примере их нет, но обычно каждому `deb`-репозиторию соответствует `deb-src`-репозиторий, то есть мы можем написать:

```
deb http://archive.ubuntu.com/ubuntu/ lucid main restricted
universe multiverse
```

```
deb-src http://archive.ubuntu.com/ubuntu/ lucid main restrict-  
ed universe multiverse
```

Но это правило действует не всегда, обычно дается подсказка, какую строку вписывать в `/etc/apt/sources.list`.



#### ПРИМЕЧАНИЕ

Перед установкой пакета необходимо обновлять список пакетов, имеющихся в репозиториях. Это делается командой `sudo apt-get update`.

При большом количестве записей суммарный объем скачанных файлов обновлений может превышать 10 Мбайт, поэтому, если не планируется самостоятельная сборка/пересборка пакетов, строки, содержащие `deb-src`, можно не включать. Если такие строки уже есть, то лучше их не удалять, а закомментировать, поставив в начале знак `#`. Таким образом можно уменьшить нагрузку на канал и ускорить время обновления списка.

Следующее поле означает зеркало, с которого будет загружаться обновление. Официальный репозиторий Linux Mint располагается по следующему адресу: <http://packages.linuxmint.com/>.



#### ВНИМАНИЕ

Поиск имени пакета производится до первого совпадения, то есть обычно установка производится из репозитория, идущего первым по списку. Поэтому, если используется локальный репозиторий (в сети, на компакт-диске, локальном жестком диске), его нужно записать в файл `/etc/apt/sources.list` первым по списку.

Чуть ниже идут репозитории, представляемые разработчиками и партнерами Ubuntu. В частности, проект Medibuntu (Multimedia, Entertainment & Distractions In Ubuntu) предлагает пакеты, которые по разным причинам не включены в официальный репозиторий Ubuntu. Например, здесь находим различные аудио- и видеокодеки, Adobe Acrobat, Google Earth и другие программы. Адрес сайта — <http://medibuntu.org>. Неофициальный репозиторий GetDeb также предлагает программы, которых пока нет в репозитории Ubuntu, или более новые версии программ.

Нужно отметить, что в самом Ubuntu, как правило, используется ближайшее зеркало репозитория, которое прописывается во время установки дистрибутива автоматически в зависимости от региона. Так, российское зеркало имеет адрес <http://ru.archive.ubuntu.com/ubuntu/>, а украинское — <http://ua.archive.ubuntu.com/ubuntu/>. В определенных случаях использование ближайшего репозитория ускоряет загрузку пакетов. Кроме этого, читая некоторые зарубежные материалы, следует их анализировать и переносить на местные реалии. Например, если в статье говорится, что для установки добавьте в список сервер <http://gb.archive.ubuntu.com/>.

[com/ubuntu/](https://ubuntu.com/), это не значит, что нужно сразу выполнять такой совет — возможно, у вас уже использовано локальное зеркало.

Полный список репозитариев Ubuntu можно найти по адресу <https://wiki.ubuntu.com/Mirrors?action=show&redirect=Archive>.

Следующей строкой идет название версии дистрибутива. В нашем примере это `isadora` для репозитория Linux Mint и `lucid` для совместимого репозитория Ubuntu. Такая строка показывает, что будут использованы основные и тщательно оттестированные пакеты, собираемые самими разработчиками. К названию дистрибутива для репозитория Ubuntu может быть добавлено одно из следующих окончаний:

- `backports` — не оттестированные, но часто очень полезные пакеты;
- `proposed` — предлагаемые, но пока не оттестированные обновления;
- `security` — обновления безопасности;
- `updates` — обновления, прошедшие стадию `proposed`.

В большинстве домашних систем использование всех категорий пакетов не является рациональным, часто кроме обязательного `main` достаточно оставить только `backports`.



#### ПРИМЕЧАНИЕ

Посмотреть список пакетов в официальной репозитории можно по адресу <http://packages.linuxmint.com/>.

И наконец, последняя позиция представлена еще четырьмя характеристиками, причем их можно поместить в одной строке:

- `main` — пакеты со свободной лицензией, поддерживаемые разработчиками Ubuntu;
- `universe` — пакеты со свободной лицензией, но поддерживаемые сторонними разработчиками;
- `multiverse` — пакеты с несвободной лицензией, поддерживаемые сторонними разработчиками;
- `restricted` — пакеты с несвободной лицензией, которые не могут быть изменены, например двоичные драйверы.

В Linux Mint таких вариантов пять: `main`, `upstream`, `import`, `backport` и `romeo` (рис. 5.21). В документации нет четко определенной роли, но, учитывая, что пакетов они предоставляют немного, обычно их все включают.

Если набрать в строке браузера указанный адрес, то можно обнаружить, что последние два компонента в описании репозитория являются подкаталогами каталога `pool` (Linux Mint) или `dist`s (Ubuntu).



Рис. 5.21. Репозиторий Linux Mint

Если продвигаться далее, то найдем еще каталоги, соответствующие архитектуре и локализации. Но добавлять тип архитектуры и локализацию в описание репозитория не нужно, все эти данные будут получены автоматически по текущим настройкам системы. Сразу возникает вопрос: что делать, если была установлена версия системы под i386, а процессор на компьютере AMD64? Как загружать пакеты под свой тип процессора? Все просто. Открываем в любом текстовом редакторе файл `/etc/apt/apt.conf.d/01ubuntu` и в секции APT добавляем следующую строку:

```
Architecture "amd64";
```

либо в любом месте:

```
APT::Architecture "amd64";
```

Сохраняем изменения, обновляем список пакетов и пробуем получить небольшой файл. Если пакет имеет в названии amd64, все прошло как нужно.

К слову, в файле `/etc/apt/apt.conf` и нескольких файлах, размещенных в каталоге `/etc/apt/apt.conf.d/`, настраивается поведение APT в различных ситуациях. Например, в `/etc/apt/apt.conf.d/01autoremove` указываются шаблоны пакетов, которые будут использованы при очистке системы. Все параметры подробно расписаны в документации, но в нормальной ситуации их меняют редко. Единственный вопрос, который часто возникает, — это установка программ в том случае, когда пользователь подключен к Интернету через прокси-сервер. Для этого создаем файл `/etc/apt/apt.conf.d/99проху` следующего содержания:

```
$ sudo gedit /etc/apt/apt.conf.d/99proxy
```

```
Acquire::http::proxy "http://192.168.0.1:3128/"Acquire::  
Proxy "true";
```

В данном случае прокси-сервер использует порт 3128 и принимает подключение по IP-адресу 192.168.0.1. Если прокси-сервер для доступа требует логин и пароль, то указываем их в строке параметров так:

```
Acquire::http::proxy "http://user:password@192.168.0.1:3128/";
```

Кроме этого в **/etc/profile** добавляем следующие строки:

```
export HTTP_PROXY="http://192.168.0.1:3128/"export FTP_PROXY=  
"ftp://192.168.0.1:3128/"
```

Если требуются логин и пароль, их указываем аналогично:

```
export HTTP_PROXY="http://user:passwd@192.168.0.1:3128/"export  
FTP_PROXY="ftp://user:passwd@192.168.0.1:3128/"
```

Если в наличии имеется установочный компакт-диск дистрибутива Ubuntu или содержащий репозиторий, его можно также добавить в список репозитариев (как создать такой диск, расскажу чуть ниже). Для этого достаточно вставить его в привод и набрать в консоли команду

```
$ sudo apt-cdrom add
```

Программа попросит вставить диск, после чего нажимаем клавишу **Enter**.

Через некоторое время в файле **/etc/apt/sources.list** появится запись, начинающаяся со строки `deb cdrom`.

## Добавление репозитариев

Добавить новый репозитарий в список известных программе АРТ можно несколькими способами. Самый простой — открыть в любом текстовом редакторе (с правами суперпользователя) файл **/etc/apt/sources.list** и вписать нужную строку. Причем следует помнить, что чем ближе к началу файла запись, тем большим преимуществом будет обладать этот репозитарий. Другой вариант — использовать графические инструменты **Менеджер программ** и **Менеджер пакетов** (Synaptic), о которых мы говорили выше.

Отдельно хочется сказать о вкладке **Авторизация**, которая доступна в этих инструментах. Она предназначена для загрузки ключей GPG, необходимых для проверки

подписи пакетов. Последнее делать вовсе необязательно, пакеты будут устанавливаться и без этого, но так вы будете уверены, что их собрал именно тот человек, с ресурса которого он получен, да и менеджер пакетов не будет задавать лишних вопросов. Месторасположение файлов ключей обычно указывается на главной странице ресурса. Его можно добавить и в командной строке при помощи утилиты `apt-key`. Например, скачаем `gpg`-ключ при помощи утилиты `wget` и добавим его в список известных АРТ:

```
$ wget http://seveas.imbrandon.com/1135D466.gpg -O- | sudo apt-key add -
```

После того как репозитории добавлены, следует обновить список пакетов. В консоли это делается командой

```
$ sudo apt-get update
```

Информация о репозиториях и пакетах собирается в каталоге `/var/lib/apt/lists` в виде файлов, имена которых соответствуют названию архива.

```
$ ls /var/lib/apt/lists
```

```
archive.canonical.com_ubuntu_dists_lucid_partner_binary-amd64_Packages
```

Заглянув в этот каталог, можно получить информацию о любом пакете. Впрочем, заглядывать и необязательно. За вас это сделают утилиты, например `sudo apt-cache showpkg gcc` позволяет узнать все о пакете `gcc`.

## Создание локального репозитория

В некоторых случаях возникает необходимость в создании локального репозитория. Например, у вас дома или в локальной сети имеется несколько компьютеров, работающих под управлением Linux Mint. Тогда, чтобы не загружать внешний канал, проще одну систему обновлять через Интернет, остальные — из локального репозитория. Локальная копия создается при помощи команды `debmirror` (требуется установить пакет `debmirror`):

```
$ /usr/bin/debmirror --nosource -m --passive --host=archive.ubuntulinux.org --root=ubuntu --method=ftp --progress --dist=lucid,lucid-security,lucid-updates,lucid-backports --ignore-release-gpg --section=main,restricted,multiverse,universe --arch=amd64
```

Большая часть параметров понятна из объяснений выше. Трафик составит не 1 Гбайт и потребует много свободного места на диске. Чаще возникает необходи-

мость создать свой репозиторий из уже скачанных файлов и пользоваться им, как обычно, с помощью apt-get. Чтобы создать список пакетов, нам понадобится утилита dpkg-scanpackages, входящая в состав dpkg-dev. Устанавливаем:

```
$ sudo apt-get install dpkg-dev
```

Теперь создаем каталог, в который копируем все пакеты:

```
$ mkdir archives
```

Переходим в образованный каталог:

```
$ cd archives
```

и даем команду

```
$ dpkg-scanpackages . /dev/null | gzip -9c > Packages.gz
```

По окончании работы утилиты будет выдан список пакетов и их число. Чтобы добавить новый репозиторий в файл **sources.list**, прописываем путь к созданному каталогу, например:

```
deb file:/home/grinder/archives ./
```

Сохраняем изменения и вводим `sudo apt-get update`. Если в **/var/lib/apt/lists** добавился новый каталог, можно поздравлять себя с победой. Теперь можно этот каталог копировать на другие системы. Естественно, при добавлении в него пакетов процедуру придется повторить.

В Интернете можно найти достаточно описания того, как перенести свой репозиторий на CD/DVD, занимают они несколько страниц. Разработчики Linux Mint включили в состав дистрибутива замечательную утилиту APTonCD (<http://aptoncd.sourceforge.net>), которая может создать загрузочный компакт-диск с пакетов, скачанных при помощи APT (рис. 5.22) или загруженных вручную, и восстановить кэш программы APT с такого диска. Фактически мы можем создать переносной репозиторий пакетов.

Пользоваться программой очень просто. После выбора **Создание диска** будет просканирован каталог **/var/cache/apt/archives**, все найденные там файлы будут добавлены в проект. Их список показывается пользователю. Сняв флажки, можно исключить пакеты (рис. 5.23).

Нажав кнопку **Добавить**, мы можем указать на отдельные файлы или каталог с .deb-файлами, которые необходимо добавить в проект. Нажимаем **Записать**, появится окно выбора, в котором следует отметить тип носителя **CD** или **DVD**. В **Папка для образа** указываем каталог, в который будет помещен готовый ISO-образ. Программа имеет еще одну замечательную возможность — создание метапакета, который

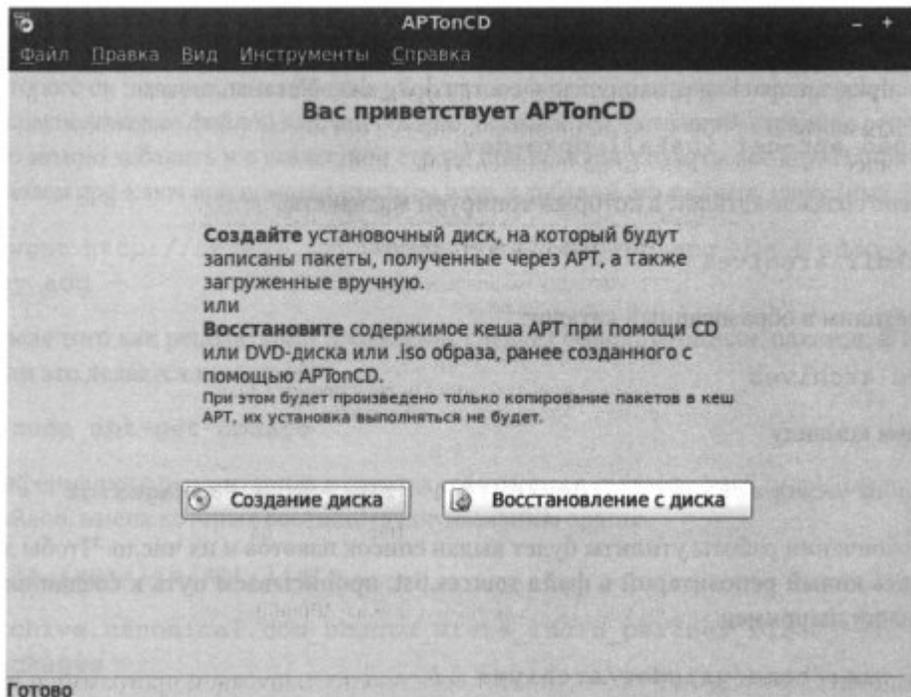


Рис. 5.22 Окно программы APTonCD

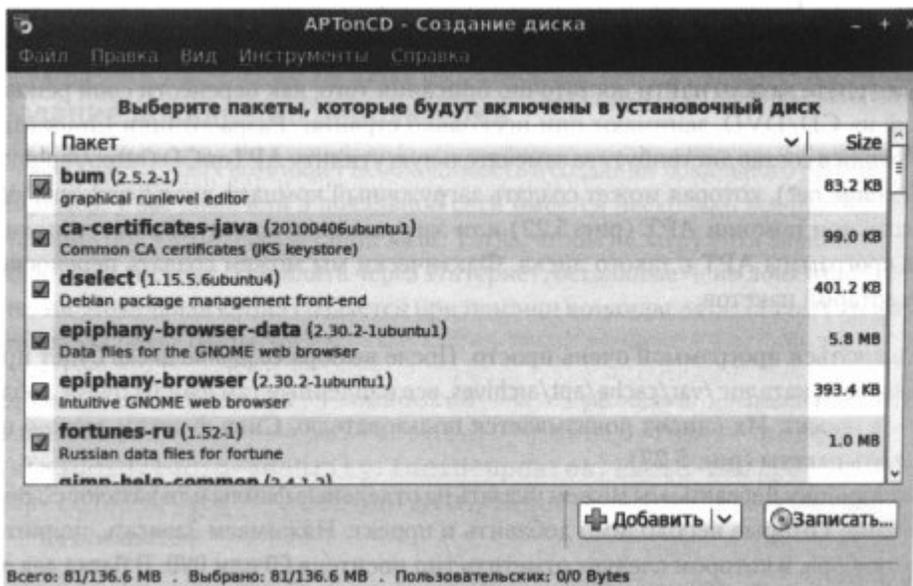


Рис. 5.23. Окно выбора пакетов APTonCD

позволит установить все программы в репозитории. Это может быть удобно, если необходимо распространить приложения на несколько систем. Чтобы такой пакет был сгенерирован, установите флажок **Создать метапакет**. Нажимаем **Применить**, программа создаст образ, по окончании процесса появится диалоговое окно, предлагающее сразу записать его на носитель (рис. 5.24).

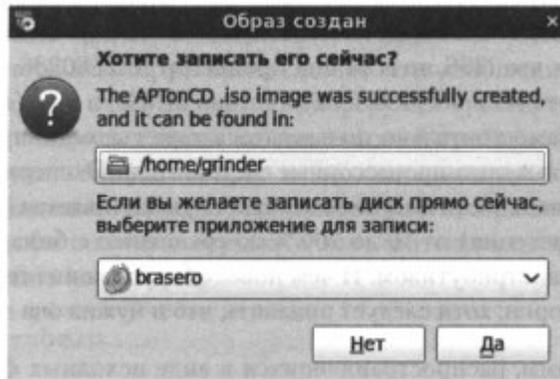


Рис. 5.24. Запись диска при помощи APTonCD

Установить программы с такого диска также просто. Запускаем программу и выбираем кнопку **Восстановление с диска** (см. рис. 5.22). Если в приводе находится диск, в следующем окне откроется список файлов. Но программа позволяет использовать и созданный ранее ISO-образ, который указываем, нажав кнопку **Загрузка**. Отмечаем пакеты и нажимаем кнопку **Восстановление с диска**.

## Компиляция программ

Даже после появления пакетов, которые представляли собой уже скомпилированные программы, компиляция долгое время оставалась и для некоторых остается основным средством установки.



### ПРИМЕЧАНИЕ

Первые прекомпилированные наборы появились в дистрибутиве SLS Linux, который в начале 1990-х гг. был самым популярным. Его создатели сочли неудобным самостоятельную сборку приложений, поэтому пользователям было предложено не только само ядро Linux и основные утилиты, но и большой набор разнообразного программного обеспечения с простой программой установки. Все это поставлялось уже в скомпилированном виде, исходный текст прилагался только для самых основных компонентов вроде ядра. Кстати, основой для популярных сегодня дистрибутивов Slackware и Debian послужил именно SLS.

Несмотря на обилие пакетных систем, о которых речь пойдет дальше, многие пользователи предпочитают самостоятельно собирать программы. Почему? Все просто. Нередко в репозиториях для не очень популярных программ находится старая

версия программ, поэтому приходится ждать, когда у сборщика дойдут до него руки.

Кроме этого иногда требуется изменить функциональность программы, установить патчи, не используемые или не поддерживаемые майнтейнером. И последняя причина — производительность. Чтобы пакет работал на максимально большом количестве оборудования, его собирают с оптимизацией под определенный тип процессора, как правило, это i386, то есть под процессор Intel 80386, выпущенный аж в 1985 г. Встречаются сборки i486, i586 (Pentium MMX) и i686 (Pentium Pro). Собирая программу самостоятельно, пользователь может скомпилировать ее под свой тип процессора, под мультипроцессорные системы и др. Эксперименты показывают, что производительность получаемого пакета увеличивается (в зависимости от аппаратного обеспечения) от 10 до 200 % по сравнению с бинарными сборками, поставляемыми с дистрибутивом. И чем новее оборудование, тем больше эффективность такой сборки; хотя следует признать, что и нужна она не всегда.

Установка программ, распространяющихся в виде исходных файлов, вызывает особые трудности у большинства начинающих пользователей Linux. Причем до сих пор считается, что это один из основных навыков, необходимых для работы в Linux. Замечу, что это очень и очень устаревшее мнение, вы сами в этом убедились, прочитав данную главу. Но попробуем разобраться.

#### **ВНИМАНИЕ!**



**Чтобы самостоятельно компилировать программы, потребуется компилятор GCC, GNU Make и прочие утилиты. В некоторых дистрибутивах они по умолчанию не устанавливаются. Чтобы установить их в Linux Mint, а также в Ubuntu и других производных дистрибутивах, необходимо ввести команду `sudo apt-get install build-essential`.**

Как правило, исходные файлы заархивированы в так называемый тарболл. Архив, в зависимости от программы, которая использовалась при его создании, имеет суффикс `.tar.gz` или `.tar.bz2` и имя вроде `program-x.xx`, где `program` — название программы, а `x.xx` — версия. В общем виде процесс установки выглядит так. Сначала распаковывается архив с помощью графической программы или команд:

```
tar xzvf program-x.xx.tar.gz
```

```
tar xjvf program-x.xx.tar.bz2
```

Затем переходим в образовавшийся каталог. Как правило, в нем можно найти файлы с именами `INSTALL` или `README`, в которых кратко описан процесс установки, сказано о зависимостях программы, даны описание программы и инструкции по сборке. В общих случаях достаточно выполнить команду `./configure` без дополнительных параметров. Скрипт самостоятельно найдет все программы и библио-

теки. И, если все будет в порядке, создаст `makefile`, необходимый для компиляции программы. Но в том случае, когда скрипт не найдет, последует сообщение об ошибке. Прочитайте его внимательно и постарайтесь понять, в чем дело.

Скорее всего, в сообщении будет сказано о том, что отсутствует какой-то файл или библиотека либо имеющаяся версия устарела. В этом случае вам придется искать и устанавливать недостающее. К сожалению, однозначного совета дать невозможно, здесь потребуются некоторый опыт. Однако если сообщение об ошибке непонятно, лучше поискать ответ на различных форумах — велика вероятность того, что кто-то уже столкнулся с такой же или похожей проблемой и нашел удачное решение. Когда будете задавать вопрос, не поленитесь вставить в него текст сообщения и указать название и версию устанавливаемой программы и используемого дистрибутива. Чтобы получить справку о дополнительных параметрах скрипта, следует использовать ключ `-help`:

```
$ ./configure --help
```

Наиболее часто используемым параметром является `--prefix`, с помощью которого указывается каталог для установки программ, отличный от используемого по умолчанию (обычно `/usr/local/`). В некоторых случаях такого скрипта нет, а есть уже готовый `makefile`. Тогда сразу же переходим ко второму этапу — вводим команду `make`, которая и скомпилирует программу. Если работа `make` завершилась без ошибок, устанавливаем программу (при этом происходит установка не только двоичных файлов, но и документации).

Для выполнения последнего шага понадобятся права суперпользователя (`root`), которые можно получить, введя команду `sudo`:

```
$ sudo make install
```

Чтобы удалить установленную таким образом программу, следует использовать команду `make uninstall`, хотя часто разработчики не используют в скриптах эту возможность.

## Конвертеры пакетов

Отдельно хотелось бы отметить наличие утилит, позволяющих конвертировать пакеты из одного формата в другой. Их возможности применения, естественно, ограничены, так как получить из пакета одного типа другой полноценный тип пакета невозможно. Кроме того, приложения, входящие в состав пакета, попросту могут не работать в другом дистрибутиве из-за различий версий библиотек, неудовлетворенных зависимостей и отличий в организации файловой системы. Поэтому их нельзя использовать для установки важных системных файлов. Однако в некоторых случаях (в первую очередь установка недвоичных пакетов,

например документация, словари и др.) их применение оправданно. Доступно несколько конвертеров; в контексте Linux Mint нас интересует alien (<http://kitenet.net/programs/alien/>). Устанавливаем ее из репозитория пакетов:

```
$ sudo apt-get install alien
```

Теперь, чтобы конвертировать rpm-пакет в deb, используем команду

```
$ alien --to-deb package.rpm
```

И наоборот, конвертируем deb в rpm:

```
$ alien --to-rpm package.deb
```

Но необходимость в alien возникает очень редко.

## Некоторые приложения, входящие в состав Linux Mint

Одной из проблем, с которой сталкиваются пользователи при переходе в Linux, - это отсутствие привычных программ. Да, здесь нет пакета Microsoft Office, Adobe Photoshop, WinAmp, Qip и тысячи других программ, к которым привык пользователь Windows. Кстати, в Linux можно найти некоторые из приложений, доступных и в Windows, в основном для работы с Интернетом — Mozilla Firefox, Opera, Mozilla Thunderbird, FileZilla и некоторые другие.

Но в Linux присутствуют полноценные аналоги большинства популярных программ, имеющие не только свои имена, но и несколько другую функциональность, особенности использования и т. д. Программы для управления системой мы рассмотрели в предыдущих главах. Теперь, чтобы упростить знакомство с системой, кратко расскажу об основных программах и утилитах, для выполнения большинства повседневных задач, входящих в комплект Linux Mint и доступных в репозитории.



### ПРИМЕЧАНИЕ

В Интернете можно найти несколько ресурсов — каталогов Linux-программ. Среди них — <http://linuxprogs.org.ua/>, <http://rus-linux.net/>, <http://www.nlxp.ru/> и др.

## Офисные пакеты

Трудно представить, что в Linux еще 10 лет назад не было полноценного офисного пакета, который мог конкурировать с Microsoft Office. Нет, конечно, решения были, но они часто не удовлетворяли всем требованиям, локализация была далека от идеала. Кроме того, в Linux был свой путь развития, который вполне устраивал разработчиков, а приложения в этой операционной системе, как вы

знаете, часто пишут «под себя». Все изменилось, когда компания Sun Microsystems изменила лицензию на разрабатываемый ею офисный пакет StarOffice на LGPL. В результате появился офисный пакет OpenOffice.org, который в итоге стал первым качественным продуктом, появившимся в Linux. Он позволяет работать с текстовыми документами, электронными таблицами, изображениями, базами данных и создавать презентации.

## Пакет OpenOffice.org

Сегодня над кодом OpenOffice.org работают как добровольцы со всего света, так и программисты корпорации Oracle Corporation (выкупившей Sun Microsystems, Inc), которая в основном и финансирует деятельность проекта OpenOffice.org. Пакет StarOffice распространяется под коммерческой лицензией и включен в некоторые коммерческие дистрибутивы Linux, а OpenOffice.org доступен под свободной лицензией, не требующей платы за его использование и с исходным кодом. С такими условиями распространения он является полноценным конкурентом дорогостоящего, хотя и более популярного, офисного пакета Microsoft Office. Именно OpenOffice.org включили разработчики Linux Mint в дистрибутив. И это вполне логично, ведь OpenOffice.org совместим со всеми широко распространенными форматами документов, электронных таблиц, презентаций и изображений, в том числе Microsoft Office последних версий. Документы выводятся качественно и практически без потерь в структуре. Хотя при сложном форматировании возможны искажения, но следует вспомнить, что документы, созданные в различных версиях Microsoft Office, тоже, бывает, отображаются не совсем корректно.



### ПРИМЕЧАНИЕ

Поскольку название OpenOffice уже является зарегистрированной торговой маркой, пакет следует называть именно OpenOffice.org.

Функционально OpenOffice.org состоит из нескольких приложений, которые можно устанавливать отдельно, выбрав таким образом действительно нужную функциональность.

В пакет OpenOffice.org входит несколько программ:

- OpenOffice.org Редактор текстов (OpenOffice.org Writer) — программа для работы с текстовыми документами (рис. 5.25) и по совместительству редактор веб-документов, редактор этикеток и визитных карточек, аналог Microsoft Word;
- OpenOffice.org Электронные таблицы (OpenOffice.org Calc) — программа для работы с электронными таблицами (рис. 5.26), аналог Microsoft Excel;
- OpenOffice.org Рисование (OpenOffice.org Draw) — очень простая по функциональности программа для создания и редактирования изображений в векторном формате (рис. 5.27), с возможностью экспорта в PDF и некоторые другие форматы;

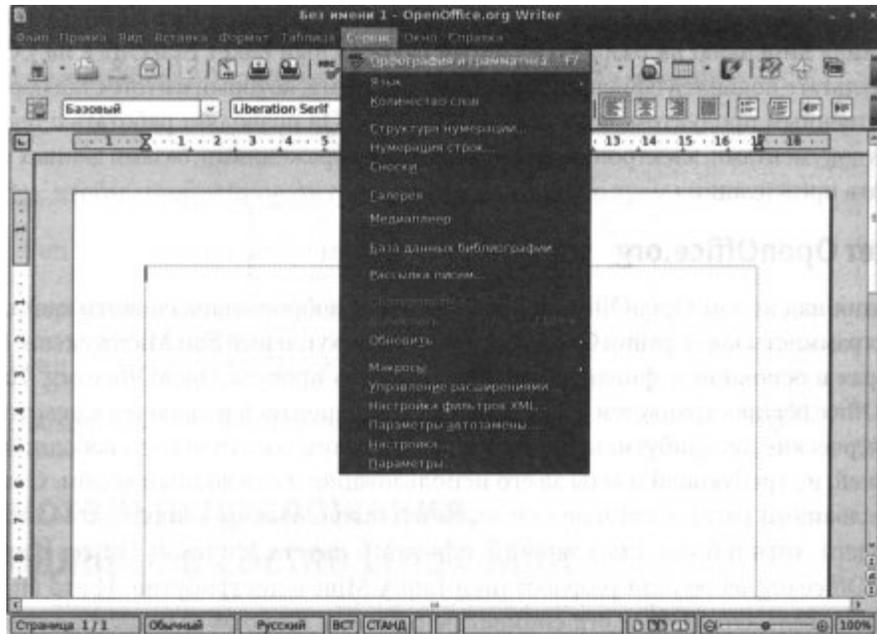


Рис. 5.25. Программа для работы с документами OpenOffice.org Writer

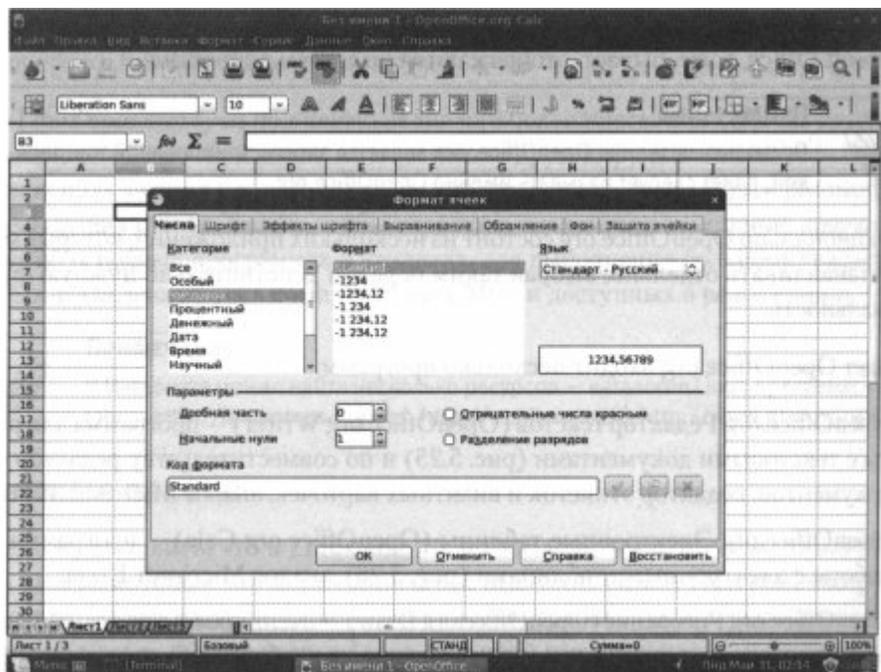


Рис. 5.26. Рабочее окно OpenOffice.org Calc

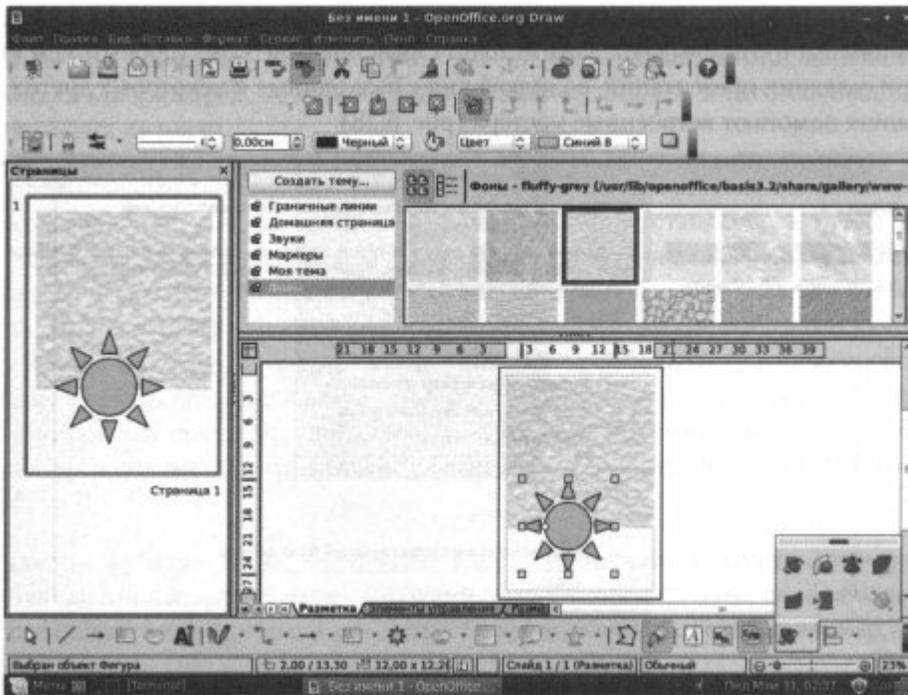


Рис. 5.27. Программа для редактирования векторных изображений из состава OpenOffice.org

- OpenOffice.org Презентации (OpenOffice.org Impress) — полнофункциональная программа для создания презентаций любого уровня сложности, аналог Microsoft PowerPoint;
- OpenOffice.org Базы данных (OpenOffice.org Base) — программа для создания и управления базами данных, поддерживающая не только Microsoft Access, но и еще 11 различных форматов баз данных. К сожалению, «родной» формат не будет понятен Microsoft Access; несмотря на простой с виду интерфейс, это очень удобная программа с большим количеством функций;
- OpenOffice.org Math — вспомогательная программа для работы с математическими формулами, аналог Microsoft Equation, его можно вызвать как из меню другого приложения, так и в качестве самостоятельной утилиты.

Пользователям, знакомым с пакетом Microsoft Office, освоить приложения из состава OpenOffice.org не составит труда. Следует также отметить отличную справку, переведенную на русский язык. Несмотря на бесплатность, любая программа, входящая в пакет, весьма функциональна и удобна в использовании. Интерфейс очень похож на Microsoft Word: кнопки на панели инструментов расположены аналогично, заголовки идентичны, и часто определить навскидку, что именно запущено — Calc или Excel, тяжело. Конечно, пользователи со стажем быстро

найдут отличия в содержании пунктов меню. Но практика показывает, что к изменениям быстро привыкают, даже находят OpenOffice.org более удобным, чем Word. При создании презентаций, подключении к базе данных и некоторых других операциях помогают встроенные мастера (рис. 5.28).

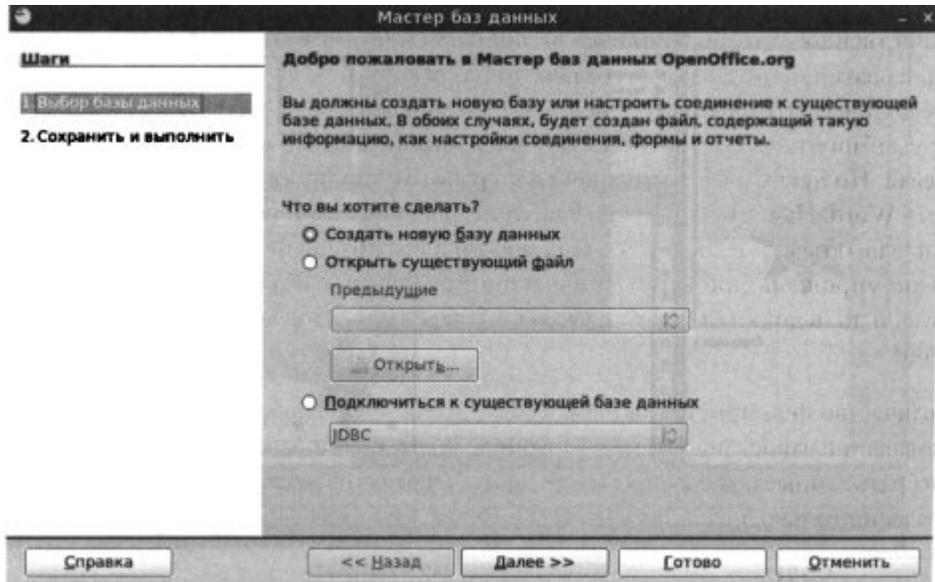


Рис. 5.28. Мастер создания баз данных

#### ПРИМЕЧАНИЕ



Официальный сайт проекта находится по адресу <http://www.openoffice.org/>, русскоязычную сборку можно скачать с <http://ru.openoffice.org/>. Существуют и альтернативные сборки, например <http://www.i-rs.ru/>. Последние две предпочтительней, так как в их комплект входят дополнительные словари для проверки орфографии, которые иначе придется устанавливать отдельно.

При беглом знакомстве с панелью инструментов бросается в глаза кнопка экспорта файла в формат PDF. Это весьма полезная функция, и, нужно сказать, она весьма востребованна — с OpenOffice.org нет необходимости дополнительно использовать программу-конвертер. Функции для форматирования документов и работы со шрифтами выполняются так же, как и в продукте от Microsoft. И кстати, возможность предпросмотра шрифтов в раскрывающемся списке на панели инструментов сначала появилась в Writer.

Остальную функциональность оценить сложнее, так как различные пользователи выполняют разные задачи, у них свои привычки и требования. Поэтому в обзорах часто можно встретить порой диаметрально отличающиеся мнения о продуктах, входящих в состав OpenOffice.org. Но среднестатистический пользователь вряд ли

найдет много недостатков, а учитывая цену OpenOffice.org, можно простить некоторые из них. Например, Writer может проверять правописание отдельных слов (при наличии словарей, которые входят в пакет **myspell-ru**), а несогласованные предложения не могут.

Единственное, что также может не понравиться, — это отсутствие возможности нарисовать таблицу, так как создавать ее можно только по стандарту, выбрав нужное количество строк и столбцов. Отдельные пользователи решают эту проблему путем импорта таблицы из Calc или воспользовавшись пунктом **Таблица • Разбить ячейки**. Но некоторые возможности по работе с таблицами сделаны даже лучше, чем в Word. Например, таблица в таблице, рисунок с помощью встроенных средств и другие объекты создаются очень просто и без потери форматирования. Аналогично упрощена процедура добавления различных объектов вроде рисунков, звука и др. в документ, хотя в **Сервис • Галерея** объектов меньше, чем в Microsoft Word.

Количество функций в Calc не уступает Excel. Реализовано автоматическое распознавание зависимостей между ячейками, очень удобен фильтр, позволяющий отобразить данные по некоторому критерию. Это можно использовать, например, для удаления только чисел.

#### СОВЕТ



По умолчанию каждый компонент OpenOffice.org сохраняет документы в своем формате, при переносе на другой компьютер следует убедиться, что данный файл будет прочитан. Выбрав **Сервис • Параметры • Загрузка/Сохранение • Общие** в поле **Тип документа** и **Всегда сохранять как**, можно указать, в каком из форматов всегда сохранять выбранный тип документов.

Все программы, входящие в пакет, тесно интегрированы друг с другом. При закрытии документа остается окно (рис. 5.29), в котором можно открыть любой документ, поддерживаемый OpenOffice.org, или вызвать любое приложение, входящее в его состав.

#### ПРИМЕЧАНИЕ



В некоторых приложениях OpenOffice.org разработчики оставили скрытыми функции так называемые «пасхальные яйца». Например, в новом документе Writer набираем **StarWriterTeam** и нажимаем клавишу F3, в результате мы увидим команду разработчиков. В Calc записываем в любую ячейку `"=Game("StarWars")"`, нажимаем Enter и наслаждаемся игрой, спасая планету от нашествия инопланетян. А чтобы в Calc поиграть в крестики-нолики, достаточно в любой из ячеек внутри диапазона A1:C3 ввести `"=Game(A1:C3;"ТicТacToe")"`. Компьютер автоматически сделает следующий ход.

Кроме этого OpenOffice.org поддерживает расширения, наподобие плагинов к браузеру Firefox. На сайте проекта и в Интернете доступно большое их количество.



Рис. 5.29. Общее окно OpenOffice.org

К сожалению, несмотря на все усилия разработчиков, пакет несколько «тяжеловат», особенно это почувствуют пользователи, у которых компьютер с объемом оперативной памяти меньше 256 Мбайт. Хотя современные компьютеры редко оснащены меньшим объемом оперативной памяти.

## Офисные пакеты GNOME Office и KOffice

OpenOffice.org не единственный офисный пакет, доступный в Linux Mint. Кроме него есть приложения, которые выполняют единичные функции, и два офисных пакета — GNOME Office и KOffice. Первый пакет построен на библиотеках Gtk+ и ориентирован на среду GNOME, а значит, больше подходит для Linux Mint. Второй использует библиотеки Qt и является «штатным» для KDE, поэтому его рекомендуется использовать тем, кто работает в KDE версии Linux Mint. Хотя при необходимости никто не запрещает установить любую из описанных офисных сред в любом оконном менеджере.

Состав приложений, входящих в проект GNOME Office, не является строго фиксированным и постоянно меняется. С информацией по этому вопросу можно ознакомиться по адресу <http://live.gnome.org/GnomeOffice>. В настоящее время это четыре приложения. Рассмотрим их.

- AbiWord (<http://live.gnome.org/AbiWord>) — текстовый процессор с достаточно широкими функциональными возможностями, легкий и понятный в использовании и не требующий большого количества системных ресурсов. Одним из его неоспоримых достоинств является возможность импорта и экспорта в формат Microsoft Word, что, согласитесь, подчас является определяющим при выборе того или иного редактора для работы.
- Gnumeric (<http://www.gnumeric.org/>) — многоцелевая электронная таблица, предназначенная для замены подобных коммерческих приложений. Разработчики отмечают, что основной задачей при ее создании была реализация максимальной совместимости с Microsoft Excel (рис. 5.30).

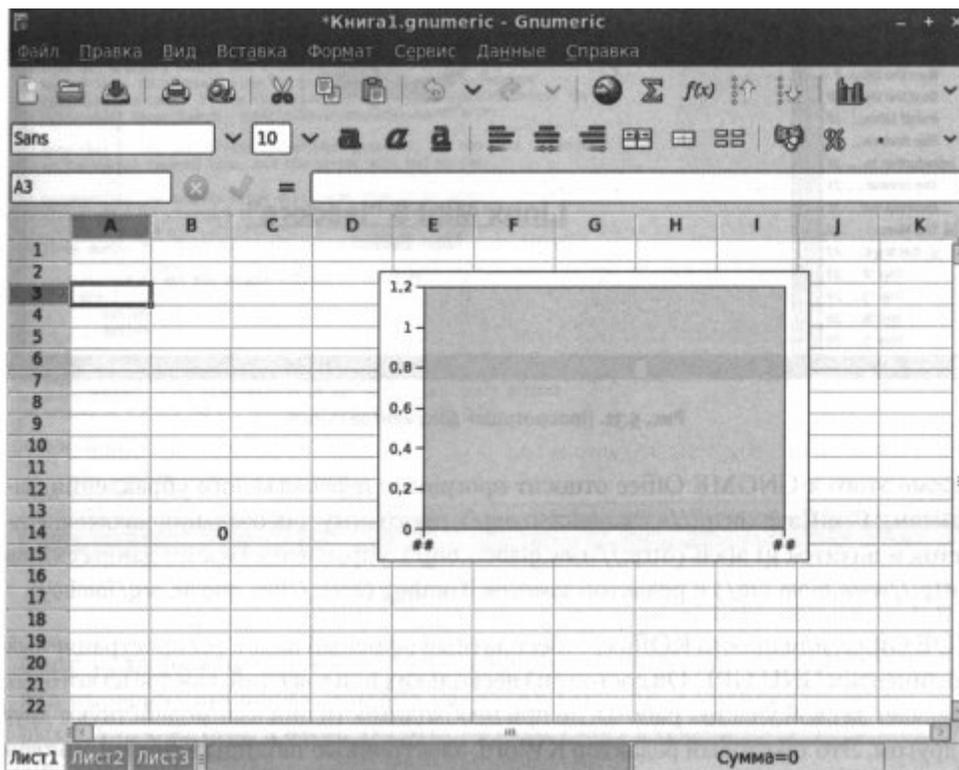


Рис. 5.30. Рабочее окно программы Gnumeric

- Evince (<http://live.gnome.org/evince>) — программа для просмотра документов в форматах PDF (рис. 5.31), PostScript, DjVu, а также TIFF и DVI, призванная заменить многочисленные просмотрщики в среде GNOME единым простым приложением. Программа уже установлена.
- Evolution (<http://live.gnome.org/Evolution>) — программа для работы с электронной почтой, контактами и временем.

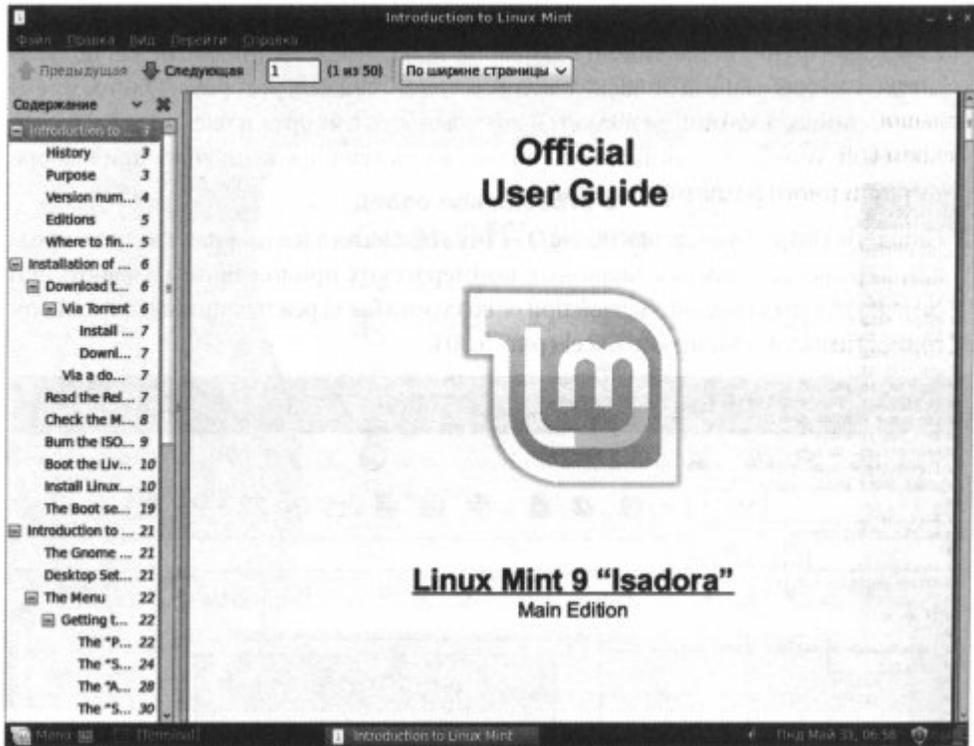


Рис. 5.31. Просмотрщик документов Evince

Кроме этого к GNOME Office относят программу персонального управления финансами GnuCash (<http://www.gnucash.org/>), программу для создания наклеек, этикеток и визиток gLabels (<http://www.glabels.org/>), управление базами данных Glom (<http://www.glom.org/>) и редактор заметок Tomboy (<http://live.gnome.org/Tomboy>).

KDE Office, или просто KOffice, — бесплатный офисный пакет, распространяемый по лицензии GNU GPL. Он состоит из нескольких приложений, каждое из которых отвечает за свой участок работы, но при необходимости они взаимодействуют друг с другом. Это текстовый редактор KWord, электронные таблицы KSpread, презентации KPresenter, управление базами данных Kexi, учет времени KPlato, диаграммы Kivio, редакторы растровой Krita и векторной Karbon графики.

Сайт проекта, на котором можно получить всю необходимую информацию, — <http://www.koffice.org/>.

## Текстовый редактор Gedit

Кроме указанных программ в репозитории Linux Mint доступно большое количество текстовых редакторов, ориентированных практически на все задачи — от обра-

ботки текстов до верстки сайтов и написания кода. По умолчанию в Linux Mint предлагается Gedit, построенный на библиотеках Gtk+, сайт проекта которого находится по адресу <http://www.gnome.org/projects/gedit/>. Это простой и не требующий большого количества системных ресурсов текстовый редактор с понятным интерфейсом (рис. 5.32).

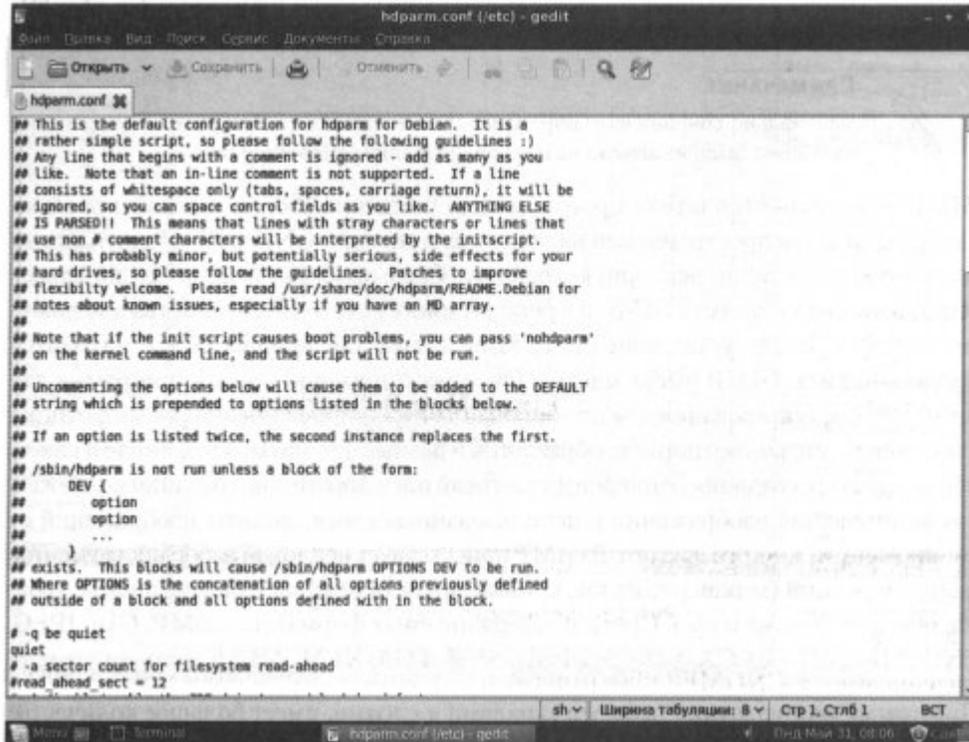


Рис. 5.32. Текстовый редактор Gedit

Gedit поддерживает проверку орфографии, подсветку синтаксиса многих языков программирования, языков разметки страниц и сценариев, среди которых C++, C#, Java, Pascal, HTML, LaTeX, PHP, Perl, Python, Ruby и многие другие.

При наведении курсора автоматически подсвечиваются открывающие и закрывающие скобки. Интерфейс переведен на несколько десятков языков, в числе которых есть и русский. Поддерживаются различные кодировки документов, при сохранении файла можно указать требуемую. Новый документ открывается во вкладке, поэтому одновременно работать с большим количеством файлов легко. Реализована проверка орфографии, автоматическая нумерация строк, функции поиска и замены текста, создание резервной копии перед сохранением, настраиваются шрифты и цвета, используемые редактором.

## Растровый редактор GIMP

Создание этого растрового редактора начато в 1995 г. двумя студентами Спенсером Кимбеллом (Spenser Kimball) и Питером Маттисом (Peter Mattis) как дипломный проект. В скором времени он получил известность и в настоящий момент поддерживается большой группой добровольцев со всего мира, а программа официально стала частью проекта GNU.



### ПРИМЕЧАНИЕ

Изначально сокращение GIMP означало **General Image Manipulation Program**, но затем оно было изменено на **GNU Image Manipulation Program**.

GIMP — мощная бесплатная программа для создания, сборки и редактирования изображений, распространяемая на условиях лицензии GNU GPL. Поддерживает растровую и частично векторную графику. Сайт проекта расположен по адресу <http://www.gimp.org/>, есть GIMP и в репозиториях пакетов большинства дистрибутивов Linux. После установки Linux Mint ярлык для запуска GIMP находится в меню Графика. GIMP умеет многое. Он может применяться как программа для рисования, ретуширования, масштабирования и кадрирования цифровых фотографий, раскраски, конвертации изображений в разные форматы, в том числе и пакетной обработки, создания баннеров для веб-сайтов и логотипов, создания коллажей, комбинирование изображений с использованием слоев, защиты изображений от копирования и многого другого. В GIMP присутствует неплохой набор инструментов цветокоррекции (микшер каналов, кривые, яркость-контраст, баланс цветов, уровни, обесцвечивание и др.). Среди поддерживаемых форматов — BMP, GIF, JPEG, PNG, TIFF, MNG, PCX, PDF, PS, PSD, SVG, TGA, XPM, ABR и много других.

Программа поддерживает работу с каналами и слоями, имеет большое количество инструментов выделения и рисования (кисть, карандаш, распылитель и штамп). Параметры любого инструмента можно настроить, изменив толщину линий, форму, прозрачность и др. Размеры обрабатываемого изображения ограничиваются только наличием свободного места на жестком диске. Количество одновременно открытых изображений также не ограничено. Поддерживаются графические планшеты. Возможности GIMP можно расширить, используя доступные плагины, которых насчитывается уже больше сотни. Частично поддерживаются и некоторые средства из Adobe Photoshop: файлы в формате PSD, файлы кистей, расширения в формате 8bf и действия (actions). Средств автоматической записи сценариев в GIMP нет, но этот недостаток компенсируется большим числом языков, на которых можно самостоятельно писать сценарии.

### СОВЕТ



Руководство пользователя на русском языке вы найдете по адресу <http://docs.gimp.org/ru/>. Кроме этого на русскоязычном сайте поддержки продукта <http://www.gimp.ru/> можно ознакомиться с несколькими обучающими статьями, рассчитанными как на начинающих, так и на опытных пользователей.

Новичков, знакомых с Adobe Photoshop, поначалу пугает необычный многооконный интерфейс программы. Но по прошествии некоторого времени к нему привыкаешь (рис. 5.33).

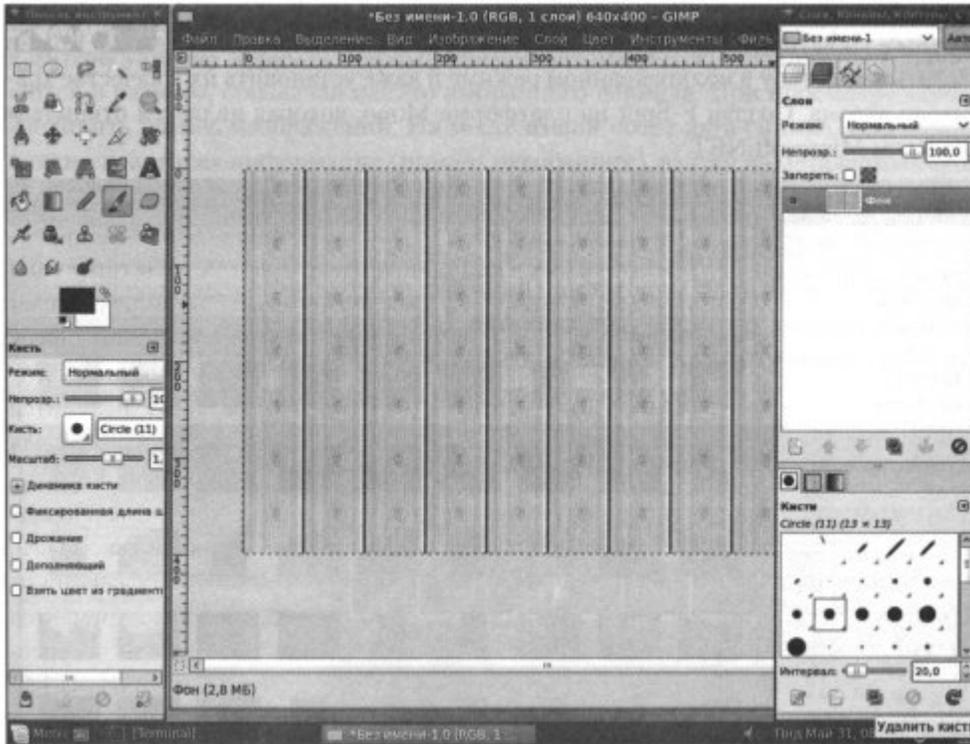


Рис. 5.33. Интерфейс GIMP

Многие считают, что GIMP является первым подтверждением того, что в процессе разработки свободного программного обеспечения может появиться продукт, которым смогут пользоваться не только хакеры. До этого свободные программы создавались хакерами и для хакеров. И можно сказать, что GIMP подготовил психологическую почву для многих других известных сегодня проектов: Mozilla Firefox, OpenOffice.org KDE, GNOME и др.

Цифровое фото сегодня уже не редкость, для работы с фотографиями в Linux так-же есть целый ряд специализированных решений.

## Менеджер фотографий F-Spot

Управляться с большим количеством фотографий очень непросто. Здесь на помощь пользователю придет программа F-Spot (<http://f-spot.org/>), которая доступна сразу после установки Linux Mint в меню **Графика**. Обладая простым и понятным

интерфейсом (рис. 5.34), программа позволяет организовать коллекцию фотографий. Поддерживаются все популярные графические форматы файлов JPEG, GIF, TIFF, RAW и др. Фотографии могут быть импортированы с жесткого диска, цифровой фотокамеры или iPod. Для удобства поиска и группировки фотографий по сходным событиям используется система меток. Из отобранных снимков можно сделать слайд-шоу в полноэкранном режиме и даже установить их в качестве хранителя экрана. Создан F-Spot на платформе Mono, которая является открытым аналогом Microsoft.NET.

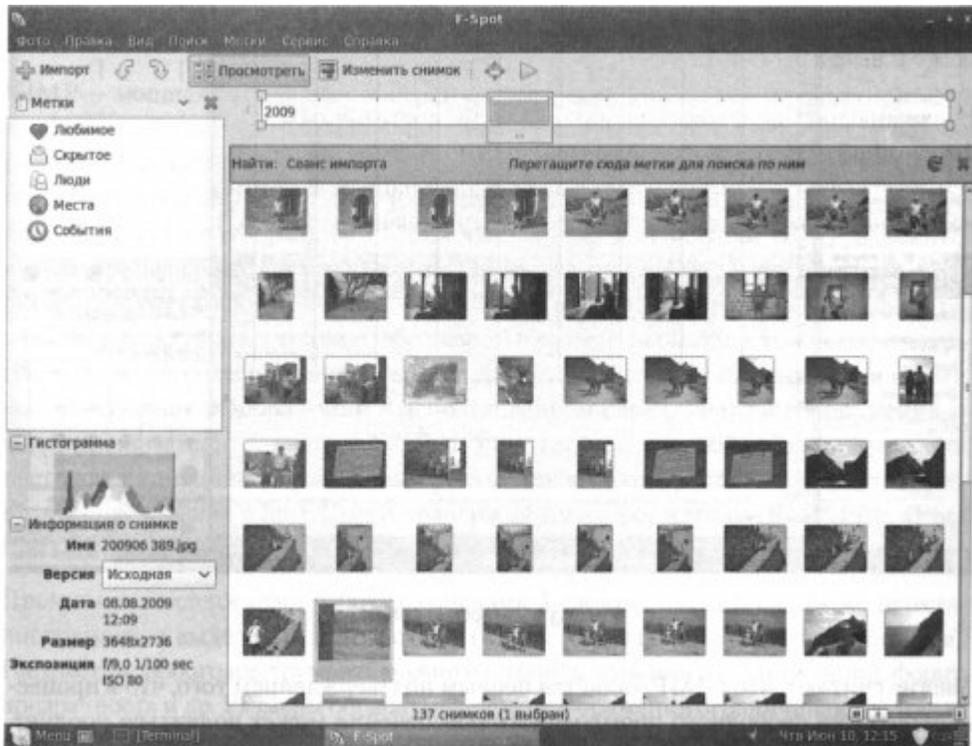


Рис. 5.34. Программа F-Spot

F-Spot — это проект с открытым исходным кодом, который распространяется по лицензии GNU GPL. После запуска программа предложит импортировать существующие фотографии. Для этого требуется указать каталог, в котором они хранятся, — подключенный фотоаппарат будет обнаружен автоматически. Проследите, чтобы был установлен флажок **Включая вложенные папки**. Файлы могут быть не только проанализированы, но и скопированы в отдельный подкаталог Photos, расположенный в домашнем каталоге пользователя. При этом все снимки будут заново рассортированы по папкам в соответствии с датой, которая берется из метаданных.

При большом количестве снимков этот процесс займет некоторое время, так что придется немного подождать. В дальнейшем импортировать снимки можно будет прямо из камеры, которые F-Spot распознает автоматически.

Сверху расположена шкала времени, используя которую можно быстро перейти к интересующему периоду. Первый снимок автоматически увеличится и уменьшится в размере, обозначая начало выбранного периода. При просмотре можно увеличить размер изображений. На месте имени стоит дата снимка, имя файла, а также некоторая информация (размер, экспозиция), взятая из метаданных файла. Выбрав снимок, получаем возможность его скопировать, повернуть, назначить метку и выставить оценку.

F-Spot предоставляет весь набор функций, достаточных для легкой обработки фотографий. Для правки снимка необходимо дважды щелкнуть кнопкой мыши на нем либо выбрать **Изменить снимок** на передней панели. К снимку можно добавить небольшой комментарий, провести его кадрирование (рис. 5.35).

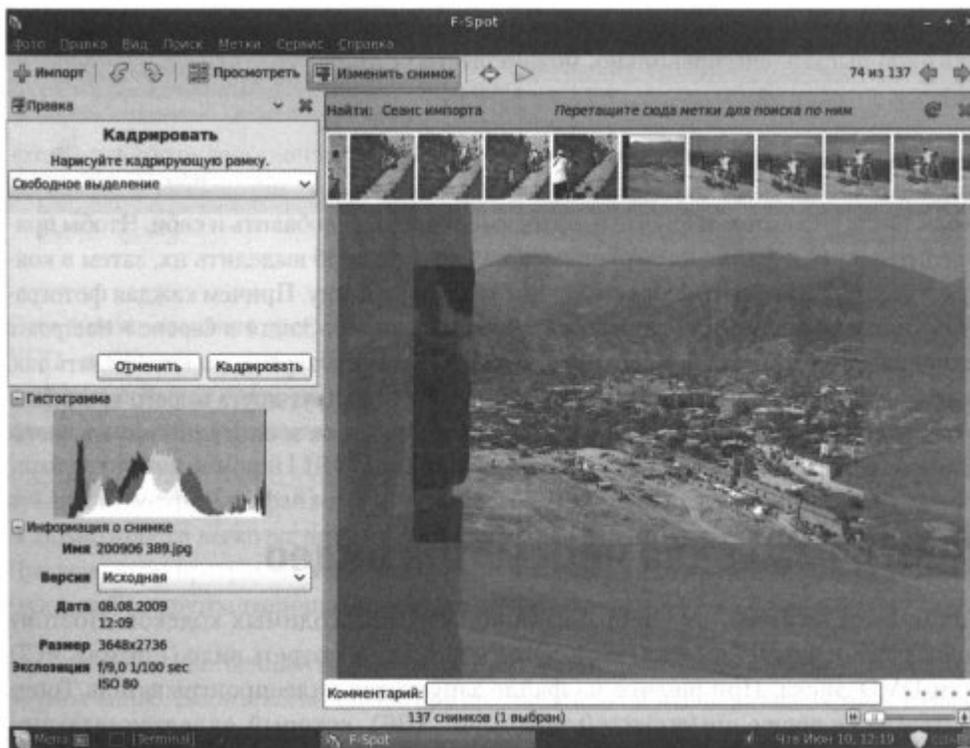


Рис. 5.35. Кадрирование снимка в F-Spot

Доступно свободное выделение участка, 4 x 3 (книга), 4 x 6 (открытка), 5 x 7, 8 x 10 и квадрат. При любом изменении у файла появляется вторая версия. Переход между

исходной и измененной версиями осуществляется при помощи выпадающего списка либо через меню **Файл • Версия**. При каждом изменении исходного файла будет добавляться новая его версия, общее количество версий не ограничено. Кроме того, F-Spot позволяет подстроить цвета снимка и устранить эффект красных глаз. Все изменения сразу же отображаются в снимке, поэтому очень легко подобрать результат. Всего одним нажатием кнопки снимок можно сделать черно-белым, и отдельная кнопка позволяет применить к выбранному снимку эффект сепии, при котором он станет похож на старое фото.

В меню **Правка** вы найдете еще пару функций, среди которых есть возможность увеличить резкость снимка и настроить время. Последняя функция весьма полезна, если время на фотоаппарате на момент фотографирования по неким причинам было установлено неверно. При редактировании времени можно обратиться к календарю. Причем, если для редактирования выбрано несколько снимков, появится дополнительно еще один пункт, позволяющий автоматически сдвинуть время последующих снимков на указанный интервал. Поэтому снимки, которые сделаны позже, всегда будут располагаться в надлежащей последовательности. Когда время снимков выставлено правильно, можно быстро отбирать их, используя функции поиска.

Чтобы упростить отбор однотипных снимков, в F-Spot использовано понятие **Метка**. По умолчанию программа имеет пять предустановленных меток: **Любимое**, **Скрытое**, **Люди**, **Места** и **События**. В случае необходимости можно добавить и свои. Чтобы прикрепить метку к файлу или группе файлов, необходимо выделить их, затем в контекстном меню выбрать **Добавить метку** и указать на метку. Причем каждая фотография может иметь несколько меток, это очень удобно. Зайдя в **Сервис • Настроить хранилище экрана** и активировав метку в поле **Хранилище экрана**, можно сделать так, чтобы выбранные фотографии периодически меняли друг друга вместо хранилища экрана. В этом же меню можно разрешить запись меток и описаний внутри метаданных, в файлах поддерживаемых форматов.

## Программы для просмотра видео

Linux Mint отличает от Ubuntu наличие всех необходимых кодеков. Поэтому после установки дистрибутива сразу же можно просмотреть видео с файла, VCD или DVD-диска. При щелчке на файле запустится видеопроигрыватель Totem (<http://www.gnome.org/projects/totem/>, рис. 5.36), который является штатным проигрывателем в среде GNOME (хотя ничто не мешает его запускать, например, в KDE), начиная с версии GNOME 2.10, официально входит в ее состав. Как и полагается, Totem хорошо интегрирован в GNOME и файловый менеджер Nautilus. Для вывода видео используются библиотеки GStreamer, но вместо них можно подключить Xine.

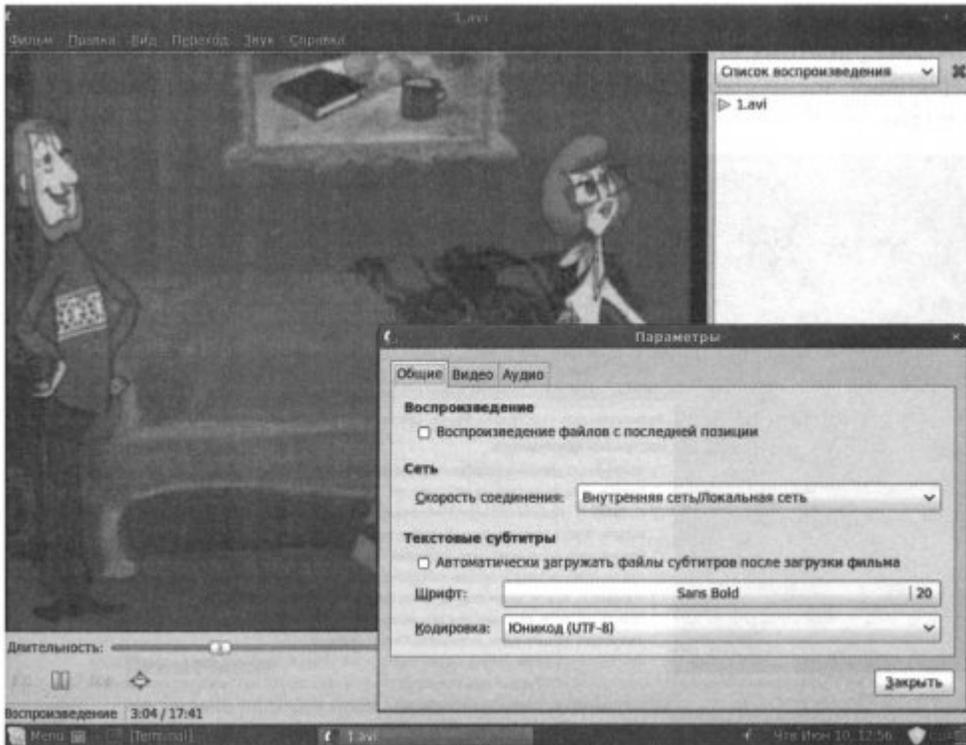


Рис. 5.36. Видеопроеигрыватель Totem

С его помощью можно просмотреть видеофайлы во многих стандартах, а также DVD-, VCD- и CD-диски, потоковое вещание в нескольких форматах. Поддерживаются плейлисты, вывод видео на несколько мониторов и ТВ-выход видекарты, полноэкранный видео, сохранение кадров, субтитры, мультисканальный звук, дистанционное управление при помощи LIRC, различные эффекты при воспроизведении аудиофайлов, просмотр видеофайлов и многое другое. Доступен плагин для браузера Mozilla. Возможности легко расширить за счет использования модулей (**Правка • Модули**). При всем этом программа очень проста в использовании и имеет минимум настроек (как и любое другое типичное приложение GNOME).

Но Totem не единственный видеопроеигрыватель, который идет в поставке Linux Mint по умолчанию. В контекстном меню при щелчке кнопкой мыши на видеофайле и **Menu** можно найти еще два пункта — один запускает GNOME MPlayer, второй — VLC.

## Видеопроеигрыватель MPlayer

На самом деле GNOME MPlayer (рис. 5.37, сайт проекта <http://code.google.com/p/gnome-mplayer/>) — это интерфейс к суперпопулярному консольному видеопроеигрывателю

MPlayer. Написан он на библиотеках Gtk+, хотя не зависит от других библиотек GNOME, поэтому его можно использовать в любой графической среде. Поддерживает воспроизведение аудио, видео, CD, DVD и VCD, потоковое аудио и видео.

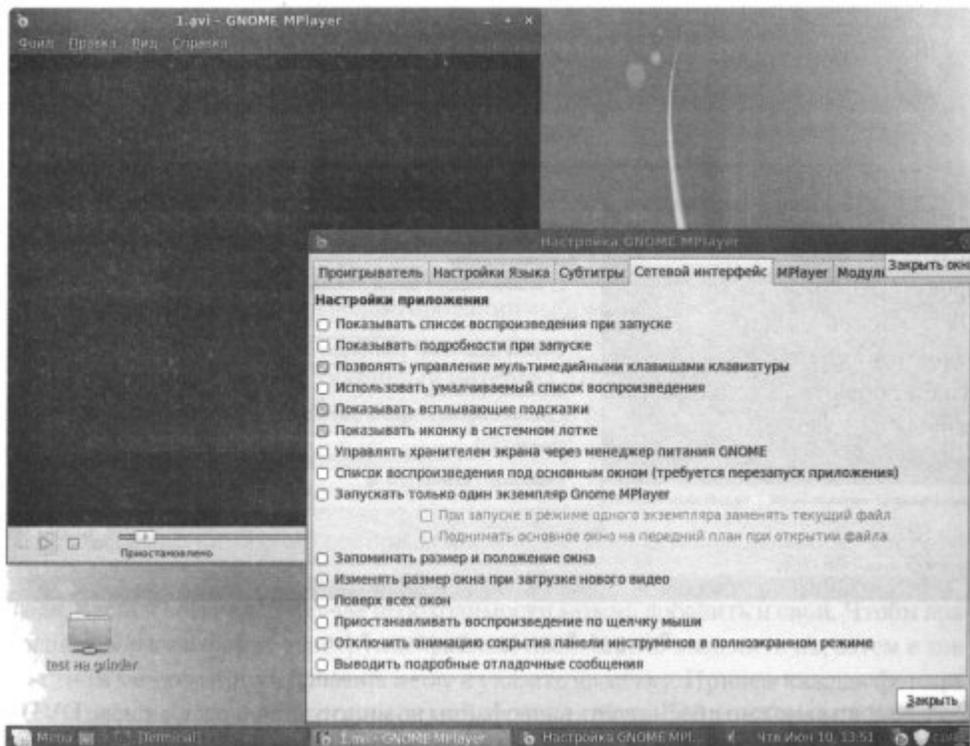


Рис. 537. Окно программы GNOME MPlayer

Программа очень проста в использовании. Поддерживается полноэкранное воспроизведение, субтитры, просмотр аналогового и цифрового ТВ. Информация об аудиоальбоме может быть загружена с Amazon.com. Для интеграции с браузерами и возможности воспроизведения видео на веб-страницах используется пакет `gecko-mediaplayer`, который в Linux Mint уже установлен.

Сам MPlayer пользуется гораздо большей популярностью. Проект начат в 2000 г. венгром Арпадом Гереффи (Arpad Gereffy), постепенно к проекту присоединились другие разработчики со всех уголков мира. Сайт проекта размещен по адресу <http://www.mplayerhq.hu/>. Изначально MPlayer ориентировался на поддержку максимального количества аудио- и видеоформатов, и сегодня их список впечатляет. Это диски CD, DVD, Video CD; файлы AVI, ASF/WMV/WMA, QT/MOV/MP4, RealMedia, Ogg Vorbis, Matroska, а также другие, записанные

с видеокодеками - DivX, MPEG-1, MPEG-2, MPEG-4, WMV, RealVideo, x264 и аудиокодеками — MP3, Vorbis, RealAudio, AC3/A52 (Dolby Digital), AAC, QuickTime, WMA и многими другими более-менее распространенными. Поддерживается потоковое воспроизведение через HTTP/FTP, RTP/RTSP, MMS/MMST, MPST и SDP.

**ПРИМЕЧАНИЕ**

Проект разрабатывает утилиту MEncoder, которая обеспечивает захват и запись телевизионного сигнала, кодирование файлов в различные форматы. Для ее установки необходимо ввести `sudo aptitude install mencoder`.

Большинство аудио- и видеоформатов поддерживаются свободной библиотекой libavcodec, выпускающейся для MPlayer. Эта библиотека использует код другого проекта FFmpeg, и ее возможностей вполне достаточно для воспроизведения DVD-дисков и файлов в различных форматах. Поддержка MPlayer закрытых форматов аудио- и видеокодеков реализована за счет подключения внешних DLL. Чтобы добавить поддержку нового формата, необходимо нужный кодек с расширением `dll` скопировать в каталог `/usr/lib/codecs` и перезапустить проигрыватель.

**ПРИМЕЧАНИЕ**

Несколько кодеков предлагаются пакетом `w64codecs`, который установлен в системе. Список всех связанных с MPlayer проектов можно найти на странице <http://www.mplayerhq.hu/design7/projects-ru.html>.

Поддерживается вывод внешних субтитров в 12 форматах (MicroDVD, SubRip, OGM, SubViewer, Sami, VPlayer, RT, SSA (а также ASS), AQTitle, JACOsub, PJS и свой собственный MPsub) и субтитры DVD (потoki SPU, VOSub и Closed Captions), представленные в различных кодировках, в том числе и кириллической. Для вывода изображения MPlayer может использовать самые разнообразные устройства, такие как X11, Quartz Compositor, DirectX, VESA, SDL, VIDIX, DGA, Framebuffer, GGI, включая такое своеобразное устройство вывода, как ASCII-Art и Color ASCII-Art. Иными словами, имеется возможность смотреть видео в чистой консоли или вообще в текстовом режиме. Стандартная консольная версия, естественно, не имеет графического интерфейса — все управление осуществляется исключительно с помощью клавиатуры или пульта дистанционного управления (если настроен пакет LIRC — Linux Infrared Remote Control, <http://www.lirc.org/>). В этом режиме на экране отображается только окно с воспроизводимым видео. Чтобы проиграть фильм в таком варианте, достаточно ввести в консоли `mplayer имя_файла`.

Дополнительно в консоль (рис. 5.38) будет выводиться информация о загруженном файле, используемых кодеках, количестве пропущенных кадров и др.



Рис. 5.38. Работа MPlayer

## Проигрыватель VLC

На самом деле под VLC (<http://www.videolan.org/>) скрывается больше, чем просто аудио- и видеопроигрыватель. Проект, начатый французскими студентами, вырос до серьезного уровня и теперь поддерживается разработчиками более чем из 20 стран. VLC позволяет не только просматривать видео, с его помощью легко можно организовать трансляцию мультимедиапоток с различных источников: видеофайлов, цифровых видео и DVD-дисков, цифровых спутниковых и телевизионных каналов и веб-камер.

При первом запуске получим предупреждение политики конфиденциальности. Оптимальным здесь будет выбор варианта скачивания обложки альбома по запросу (рис. 5.39).

Управлять запущенным видео очень просто (рис. 5.40). Выбрав **Вид • Настройка интерфейса**, можно изменить внешний вид проигрывателя по своему усмотрению. Кроме стандартных функций поддерживаются и улучшения, например ускорение и замедление воспроизведения, переход к определенному времени, возможно управление с пульта при помощи пакета LIRC. В окне **Регулировки и эффекты** до-

ступны видео- и аудиоэффекты, здесь же настраивается сдвиг аудио и видео при воспроизведении. Программа поддерживает несколько интерфейсов управления — консоль, telnet, веб. Чтобы организовать трансляцию, следует открыть вкладку **Инструменты • Конфигурация VLM** и прописать в ней все параметры.

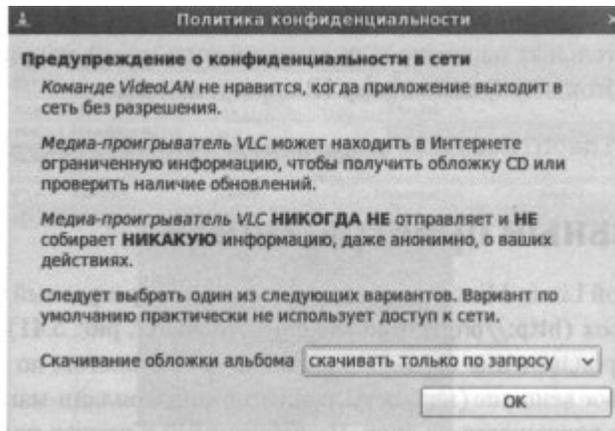


Рис. 5.39. Политика конфиденциальности VLC



Рис. 5.40. Окно видеопроигрывателя VLC

Поддерживается запуск в консоли. Чтобы посмотреть фильм, просто указываем название файла в качестве параметра:

```
$ vlc videofile.mpg
```

Программа сама подберет необходимый декодер, после чего начнется воспроизведение. Дополнительных параметров поддерживается много, например, чтобы отправить видеопоток по определенному IP-адресу, вводим:

```
$ vlc -vvv videofile.avi --sout udp://192.168.1.2 --ttl 1
```

## Музыкальные проигрыватели

Вместе с системой Linux Mint пользователь получает и прекрасный аудиопроигрыватель Rhythmbox (<http://projects.gnome.org/rhythmbox/>, рис. 5.41), который поддерживает воспроизведение не только с локальных каталогов, но и с сетевых ресурсов и потоковое вещание (подкасты, радиостанции и онлайн-магазины музыки) с возможностью сохранения на диск. Разработанный в рамках проекта GNOME, он интегрируется с некоторыми приложениями — Nautilus, Pidgin, XChat и др.

Их список легко отредактировать через меню **Музыка**. Управиться с большой коллекцией музыки с его помощью очень просто. Поддерживается поиск и сортировка мелодий, работа с ID3-тегами. Возможно задание правил отбора и создание «умных списков» воспроизведения, которые затем будут автоматически обновляться. Программа скачивает обложки альбомов и тексты песен. Пользователь может выставлять рейтинги к отдельным песням, которые затем будут учитываться при случайном воспроизведении (любимые песни будут проигрываться чаще). Внешний вид Rhythmbox можно менять, например, выбрав **Вид • Вечеринка**, мы уберем панель Рабочего **стола** внизу. Функциональность расширяется за счет подключения модулей, по умолчанию в **Правка • Модули** отключена половина из доступных. Например, некоторые плагины позволяют копировать и записывать AudioCD, управлять Rhythmbox при помощи LIRC, отправлять дорожки по электронной почте и т. д.

Кроме того, Rhythmbox поддерживает популярный сегодня iPod.

**Альтернативные музыкальные проигрыватели.** Естественно, кроме Rhythmbox в репозитории Linux Mint можно найти несколько десятков музыкальных проигрывателей в Linux. Кого не устраивает многофункциональность Rhythmbox, могут подобрать себе проигрыватель проще. Сторонники классического Winamp до недавнего времени использовали XMMS (X Multimedia System), который имел схожий с ним интерфейс. Проект практически прекратил свое существование. Однако его идеи возродились сразу в нескольких решениях — проигрывателях Audacious (рис. 5.42), BMPx, Beep Media Player (BMP) и XMMS2. Например, Audacious ([http://audacious-media-player.org/Main\\_Page](http://audacious-media-player.org/Main_Page)) представляет собой пере-

работанный XMMS с обновленным интерфейсом под библиотеки GTK+ 2. Основное преимущество его по сравнению со многими решениями — легкость.

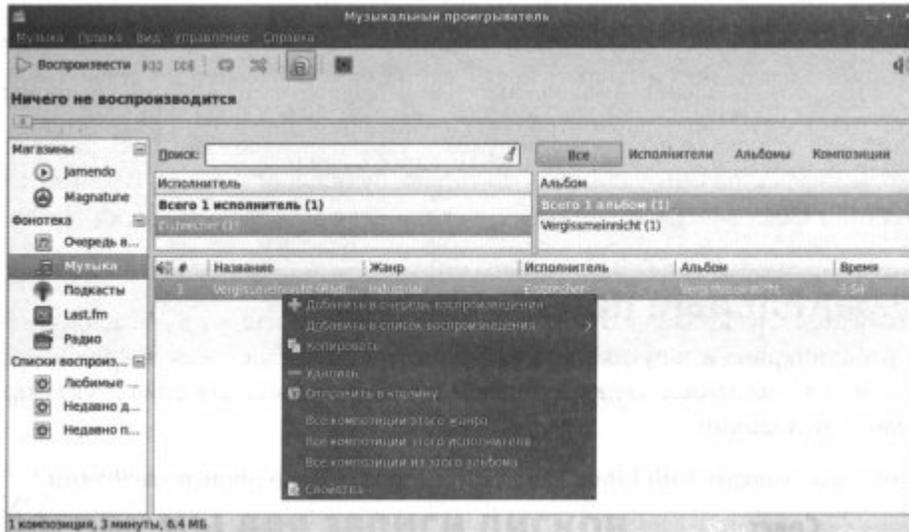


Рис. 5.41. Окно проигрывателя Rhythmbox



Рис. 5.42. Интерфейс Audacious

Поэтому его можно советовать пользователям, у которых слабые компьютеры. Как и в XMMS, поддерживаются скины (классического стиля) от Winamp 2. Для этого достаточно скачать понравившийся скин (с расширением wsz) с сайта <http://www.winamp.com/skins> и скопировать его в каталог `~/audacious/Skins`, после чего перезапустить проигрыватель и выбрать новый скин в списке **Параметры • стили**. Поддерживается воспроизведение музыкальных файлов во всех популярных форматах. В комплект включено несколько модулей — last.fm, поддержка LIRC, различных звуковых эффектов и визуализации. Есть даже будильник — Alarm. В репозитарии Linux Mint пакет audacious уже есть. Для установки всех его компонентов вводим

```
$ sudo aptitude install audacious
```

Пользователи, работавшие с XMMS/Winamp, вряд ли найдут в Audacious что-то экстраординарное и запутаются в настройках. Просто щелчком кнопкой мыши в левом углу вызываем меню, выбираем пункт **Воспроизвести** файл и указываем каталог с мелодиями.

Кроме того, репозитарий Linux Mint богат консольными проигрывателями.

#### СОВЕТ



Найти все проигрыватели можно при помощи команды `sudo apt-cache search sound`.

Так, на слабых компьютерах (начиная от процессора i486) для воспроизведения MP3 можно использовать консольную утилиту `mpg123` (<http://www.mpg123.org/>) или аналог `mpg321` (<http://mpg321.depooter.org/project/>). Достаточно перейти в каталог, в котором находятся такие файлы, и ввести команду `mpg123 ./*.mp3` - и все файлы, находящиеся в нем, будут воспроизведены. Несмотря на кажущуюся простоту, это довольно мощные утилиты, умеющие многое. Кстати, многие проигрыватели, в том числе и Winamp, используют `mpg123` для воспроизведения MP3-файлов. Аналогично есть консольная утилита для файлов Ogg Vorbis-`ogg123`. Есть консольные утилиты, имеющие псевдографический интерфейс. Например, `mp3blaster` (<http://mp3blaster.sourceforge.net/>) — диалоговая программа с простым интерфейсом (рис. 5.43), умеющая воспроизводить аудиофайлы не только формата MP3, но и Ogg Vorbis, WAV и SID. Это один из немногих текстовых аудиопроигрывателей, допускающий полное интерактивное управление в процессе воспроизведения с возможностью создания очень гибких списков воспроизведения, причем не прерывая воспроизведение. Проигрыватель воспроизводит большинство форматов MP3 и имеет встроенный полноценный миксер.

В качестве альтернативы `mp3blaster` могут выступать `playmp3list` (<http://rucus.ru.ac.za/~urban/projects/playmp3list/>) или MOC (Music On Console, <http://moc.daper.net/>), являющиеся неплохими консольными фронт-эндами для `mpg123`.

В плане работы с мультимедиа хочется также указать на два проекта — MythTV и Freevo, которые позволяют превратить компьютер в настоящий медицентр.

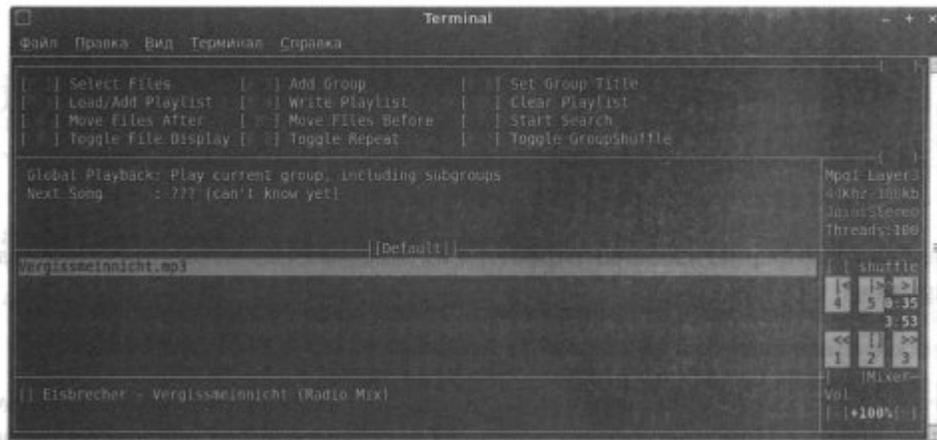


Рис. 5.43. Проигрыватель mp3blaster

## Программы для записи дисков

Устройства для записи CD и DVD давно уже вошли в стандартную комплектацию компьютера. Чтобы записать диск, нужна соответствующая программа. Команда `sudo aptitude search burn` выдает длинный список приложений. Все программы с графическим интерфейсом являются фронт-эндами к консольным утилитам `mkisofs`, `cdrecord` и `dvd+rw-tools`.

Сначала скажу пару слов о том, как записать диск в консоли. Этот процесс в Linux разбит на два этапа. В первую очередь создаем ISO-образ:

```
mkisofs -r -o cd_image.iso путь_к_файлам
```

Для записи образа на компакт-диск используем следующую команду:

```
$ cdrecord -v dev=0,0,0 speed=4 -data cd_image
```

Параметр `dev` указывает на идентификатор привода для записи дисков. Чтобы его узнать, достаточно ввести команду `cdrecord -scanbus`. При помощи `cdrecord` можно создать и аудиодиск:

```
$ cdrecord -v speed=4 dev=0,0,0 -audio track1.wav track2.wav
```

Для записи DVD используется другая команда:

```
$ growisofs -Z /dev/dvd=disk.iso
```

хотя `growisofs` позволяет записывать диски на лету, без создания промежуточного образа:

```
$ growisofs -dvd-compat -Z /dev/dvd -dvd-video ./temp
```

Кроме того, из командной строки можно быстро отформатировать перезаписываемый DVD или CD. Например, чтобы полностью отформатировать DVD-RW, достаточно ввести команду

```
$ dvd+rw-format -force=full /dev/dvd
```

Для быстрого форматирования параметр `full` не указываем. Аналогично форматируется и CD-RW:

```
$ cdrecord dev=0,0,0 blank=fast
```

Эти команды имеют много параметров, но обычный пользователь вряд ли захочет запоминать их. Наверное, это и не нужно, в Linux Mint после установки поставляется программа для записи дисков — Brasero (<http://projects.gnome.org/brasero/>). Как и любая другая программа, разрабатываемая в рамках проекта GNOME, Brasero имеет простой и понятный даже новичку интерфейс, поэтому с созданием диска сможет справиться каждый. После запуска выводится меню, в котором пользователь должен выбрать тип записываемого диска (рис. 5.44).

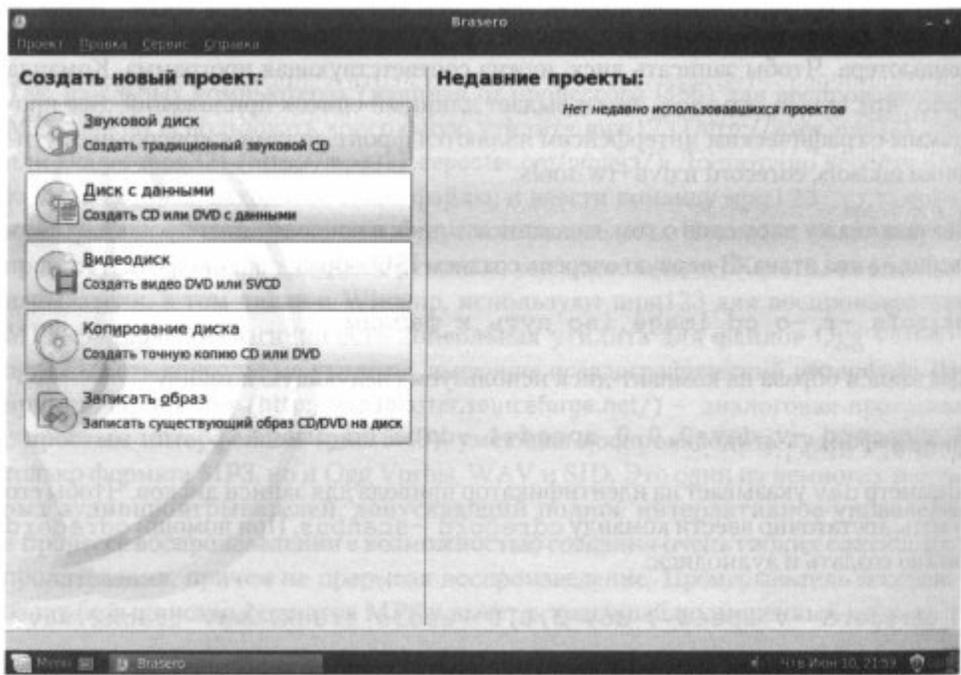


Рис. 5.44. Рабочее окно программы Brasero

Поддерживается запись Audio CD, Video CD или DVD, дисков с данными. Файлы в проект можно добавить при помощи встроенного файлового менеджера или

простым перетаскиванием. Возможно переименование файлов и каталогов в проекте, использование фильтров для отбора файлов. С помощью Brasero легко создать точную копию диска, записать подготовленный ранее образ с расширением .iso или .cue, стереть перезаписываемый компакт-диск. Записанный диск можно проверить после записи. Кроме этого в Brasero встроен простой редактор обложек.

## Редактирование ID3-тегов

Редко когда пользователи задумываются о необходимости правильного заполнения информационных тегов в музыкальных файлах. Но с ростом коллекции найти что-либо становится тяжело. Можно, конечно, рассортировать файлы вручную, переименовывая и копируя каждый, но проще заполнить поля ID3-тегов, и всю сортировку или поиск можно будет затем выполнить автоматически. Практически все программы для копирования аудиодиска умеют автоматически заполнять поля тегов, основываясь на информации в специальных базах. В репозитории Linux Mint находится более десятка программ, позволяющих отредактировать ID3-тег, найти их просто:

```
$ sudo apt-cache search id3
```

В результате получим длинный список программ с самым разным интерфейсом — консольные, псевдографические и графические. Но мы остановимся на Easy Tag (<http://easytag.sourceforge.net/>) — удобной программе для просмотра и редактирования тегов в файлах форматов MP3, MP2, FLAC, MusePack, MP4/AAC, Monkey Audio и Ogg Vorbis с понятным и локализованным интерфейсом, построенным на библиотеках Gtk+ (рис. 5.45). В поставке ее нет, но из репозитория установить ее просто:

```
$ sudo aptitude install easytag
```

После запуска Easy Tag вначале просканирует домашний каталог пользователя, чтобы найти файлы в поддерживаемых форматах. Во вкладках **Общие** и **Картинки** Easy Tag очень просто отредактировать любое поле, доступное в тегах: **Название**, **Исполнитель**, **Альбом**, **Год выпуска**, **Номер трека**, **Жанр**, **Комментарии**, **Композитор**, **Первый исполнитель**, **Авторское право**, **Ссылка (URL)**, **Кодировщик** и **Прикрепленное изображение**. При этом реализовано рекурсивное заполнение тегов, в том числе и в подкаталогах, удаление, переименование и сохранение. При вводе некоторых данных, вроде названия альбома, даты и др., реализовано автозаполнение полей. Кроме информации в тегах пользователю выводится информация о файле (битрейт, время звучания, размер, режим стерео и др.). При наличии соединения с Интернетом для заполнения полей тегов можно использовать один из серверов CDDb, для чего достаточно нажать кнопку **Поиск CDDb**. Возможны автоматическое заполнение тегов исходя из названий файлов и обратная операция — переименование каталогов и файлов исходя из содержимого тегов. Для этого могут использоваться шаблоны или специальный текстовый файл.

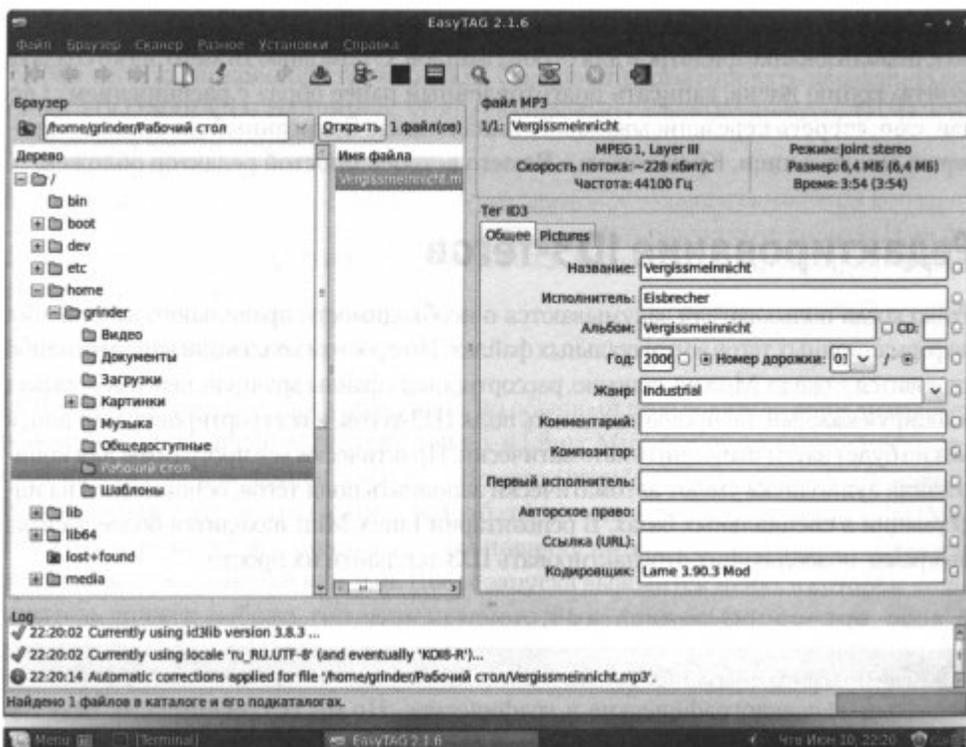


Рис. 5.45. Программа EasyTAG

Кроме этого при помощи сканера тегов и имен файлов информация в выбранных полях тегов легко конвертируется в строчные, заглавные, доступны транслитерация символов и изменение кодировки. Пробелы или подчеркивания в названиях полей легко конвертируются друг в друга. Если пользователь ошибся при вводе, ситуацию может исправить многоуровневая отмена действий. Имеется встроенный древообразный браузер просмотра, возможен вывод файлов по исполнителю, альбому. Доступны многочисленные варианты сортировки (по тегу или свойствам файла) и поиска нужных файлов. Из отобранных файлов можно составить M3U плейлист, для чего достаточно выбрать **Разное • Сохранить список записей**. При необходимости **прямо** из окна программы можно запустить внешний проигрыватель и прослушать выбранные мелодии.

## Видеоредактор OpenShot

Еще в начале XX века для Linux практически не существовало удобных и стабильно работающих приложений, позволяющих захватывать и редактировать видео. Сегодня ситуация в корне отличается. Есть для этой системы и нелинейные видеоредакторы, которые не стесняются использовать профессионалы, и редакторы

попроще, и масса приложений, позволяющих перекодировать видеофайл из одного формата в другой. Поиск в репозитории Linux Mint выдаст длинный список приложений — Avidemux, Kino, Cinelerra, Kdenlive, PiTiVi и OpenShot (<http://www.openshotvideo.com/>). Последний устанавливается вместе с системой. Интересно, что проект возник в начале 2008 г., когда его автор Джонатан Томас установил Ubuntu, в котором его устраивало все, кроме редактора видео. Редактор написан с нуля. И сегодня этот проект признан разработчиками нескольких дистрибутивов, которые устанавливают его по умолчанию.

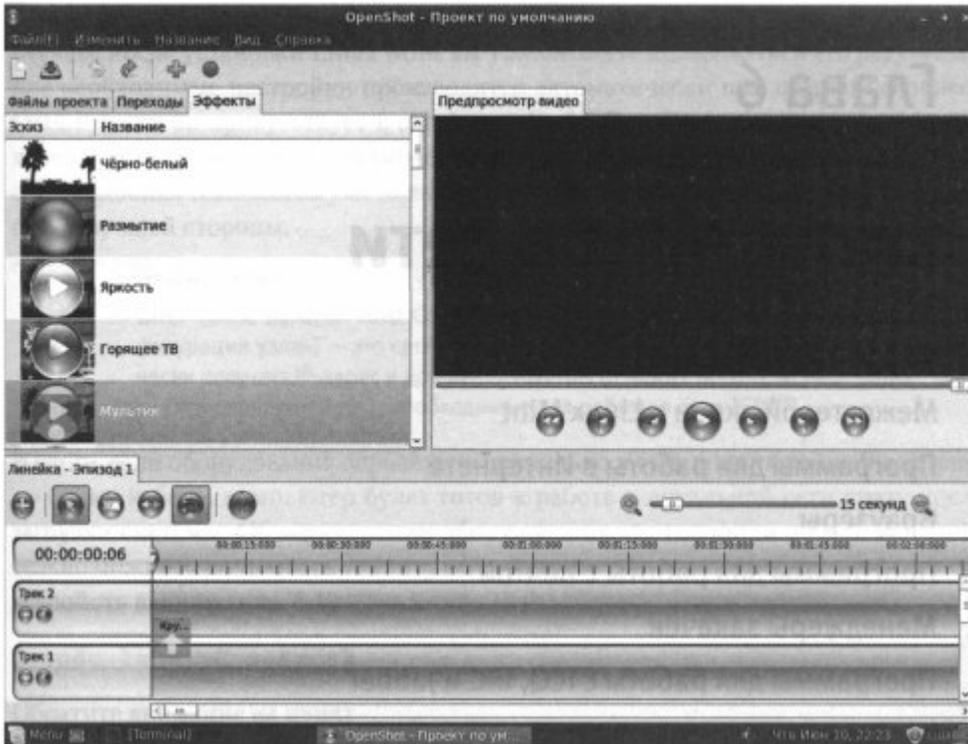


Рис. 5.46. Видеоредактор OpenShot

Это простой в использовании видеоредактор (рис. 5.46), с работой в котором может разобраться любой желающий. Поддерживается большое количество видео- и аудиоформатов, возможна вставка переходов и использование эффектов, как видео (яркости, контрастности, баланса белого, сепия, эффекта старого фильма и т. д.), так и звуковых. Программа легко интегрируется в среду GNOME, файлы в проект можно добавлять простым перетаскиванием. Экспорт файлов упрощает наличие готовых профилей.

# Глава 6

## Linux Mint в сети

Подключаемся к сети

Межсетевой экран в Linux Mint

Программы для работы в Интернете

Браузеры

Программы для работы с почтой

Менеджеры закачек

Программы для работы с ICQ, IRC и Jabber

Пользователю в Linux Mint предоставляется полный набор утилит и приложений для выхода в Интернет и подключения к его различным ресурсам — веб-сайтам, системам мгновенного обмена сообщениями, электронной почте и т. д. В дальнейшем будем считать, что читатель уже имеет опыт работы с этими сервисами, просто разберем процесс подключения и некоторые приложения.

## Подключаемся к сети

Если компьютер подключен к локальной сети Ethernet или WiFi, то в большинстве случаев после установки Linux Mint вы уже сможете пользоваться его ресурсами. Все необходимые настройки производятся автоматически при помощи сервисов DHCP (если он присутствует в сети) или Avahi. Сервис Avahi (<http://avahi.org/>) позволяет программам автоматически обнаруживать сервисы и узлы в локальной сети, подобная технология уже давно используется в Mac OS X и зарекомендовала себя с лучшей стороны.



### ПРИМЕЧАНИЕ

**DHCP** (англ. Dynamic Host Configuration Protocol — «протокол динамической конфигурации узла») — это сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры (IP-адрес шлюза по умолчанию, адреса DNS-серверов и др.), необходимые для работы в сети TCP/IP.

В итоге, если оборудование определено правильно и есть подключенный к сетевому блоку кабель, компьютер будет готов к работе в локальной сети сразу после загрузки системы. Об определении оборудования мы говорили в гл. 3, дополнительно можно использовать утилиту `lshw`, например для вывода состояния сетевых устройств вводим (рис. 6.1):

```
$ sudo lshw -C network
```

Обратите внимание на пункт

```
logical name: eth0
```

В этом пункте показано логическое имя сетевого устройства, под которым к нему следует обращаться в системе. Кроме того, рассмотрим следующую команду:

```
$ ifconfig -a
```

```
eth0      Link encap:Ethernet  HWaddr 00:0c:29:4e:7f:7d
          inet  addr:192.168.17.146    Bcast:192.168.17.255
          Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe4e:7f7d/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```



Появившееся окно состоит из пяти вкладок, в которых настраиваются проводные (рис. 6.2), беспроводные соединения, VPN, DSL и подключение к широкополосным 3G-сетям.

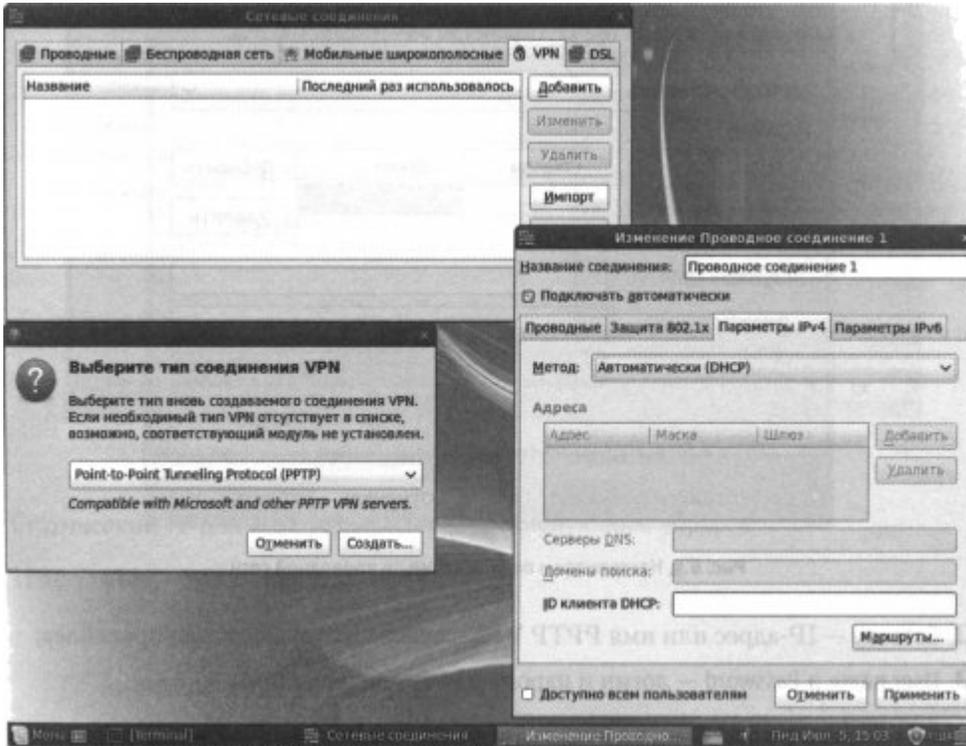


Рис. 6.2. Окно настроек при помощи NetworkManager

Просто выбираем нужный пункт и нажимаем кнопку **Добавить**. После этого действуем согласно подсказкам мастера или заполняем параметры в предложенном окне. Например, настроим популярное сегодня PPTP VPN-подключение к Интернету. Вначале выбираем Проводные • **Добавить**, отмечаем имеющееся в списке соединение или создаем новое. Вводим название соединения, во вкладке Проводные указываем MAC-адрес сетевой карты, для которой производится настройка, и на вкладке Параметры IPv4 настраиваем IP-адрес, сетевой шлюз, сетевую маску и маршруты (все это можно узнать у провайдера, рис. 6.3).

Хотя если эти данные получаются при помощи DHCP, то настраивать, скорее всего, эту часть не придется. Не забываем установить флажок **Подключать автоматически**, чтобы соединение устанавливалось после перезагрузки системы. Сохраняем настройки (потребуется ввести пароль) и переходим во вкладку **VPN**. Чтобы настроить новое подключение, нажимаем **Добавить**, в открывшемся окне отмечаем тип подключения PPTP. Затем в появившейся вкладке указываем:

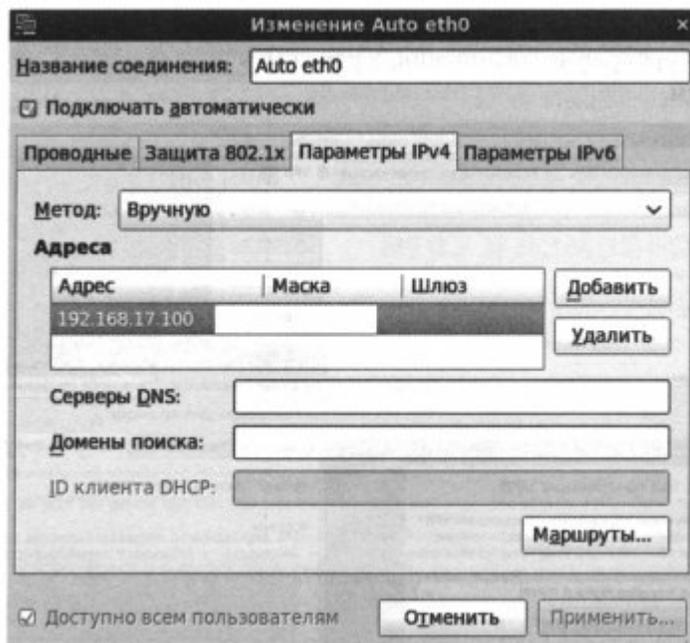


Рис. 6.3. Настраиваем подключение по проводной сети

- Gateway — IP-адрес или имя PPTP VPN-сервера, которые указал провайдер;
- User name и Password — логин и пароль для аутентификации.

Нажав кнопку Advanced, можно указать протоколы, которые следует использовать для аутентификации, разрешить MPPE-шифрование (рис. 6.4). После нажатия кнопки **Применить** будет создано новое соединение. Все очень просто.

Аналогично при помощи NetworkManager настраиваются и другие подключения. Теперь рассмотрим процесс изнутри, чтобы понять суть. Такие знания помогут разобраться в случае, если что-то не получится настроить.



#### ПРИМЕЧАНИЕ

В Linux Mint и некоторых других дистрибутивах для выхода в Интернет пользователь должен входить в группу `dip`.

## Настройка Ethernet-соединения

Настройки сетевых интерфейсов сохраняются в конфигурационном файле `/etc/network/interfaces`. Его структура хорошо расписана на многих ресурсах Интернета, она проста и понятна. Например, укажем, что для сетевого устройства `eth0` IP-адрес назначается динамически при помощи сервиса DHCP:

```
iface eth0 inet dhcp
```

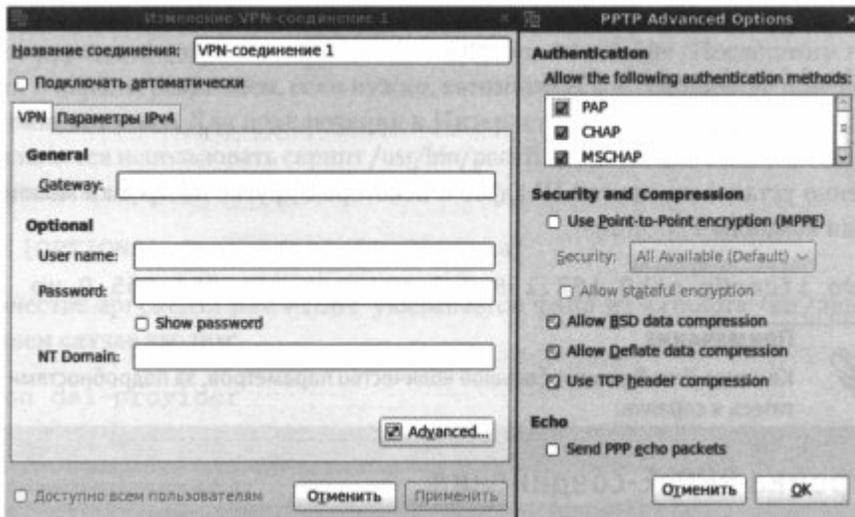


Рис. 6.4. Настройка PPTP-соединения

Статический IP для eth1 устанавливается следующим образом:

```
iface eth1 inet static
    address 192.168.0.25
    network 192.168.0.0
    gateway 192.168.0.1
    netmask 255.255.255.0
    mtu 1492
```

Во втором случае мы указали полностью все параметры — IP-адрес, сетевую маску, IP-адрес шлюза и т. д. Кроме того, обязательно указываем в файле `/etc/resolv.conf` адрес хотя бы одного DNS-сервера (а лучше двух), иначе разрешение имени в Интернете производиться не будет:

```
$ sudo gedit /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
```

В примере использован адрес Google Public DNS-серверов, хотя список IP-адресов обычно выдает провайдер. Чтобы перезагрузить настройки сетевых интерфейсов, используйте команду

```
$ sudo /etc/init.d/networking force-reload
```

Один интерфейс включить и выключить можно так:

```
$ sudo ifconfig eth0 down
```

```
$ sudo ifconfig eth0 up
```

Временно установить другой IP-адрес и некоторые другие настройки можно при помощи команды `ifconfig`:

```
$ sudo ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```



#### ПРИМЕЧАНИЕ

Команда `ifconfig` имеет большое количество параметров, за подробностями обратитесь к справке.

## Настройка PPPoE-соединения

PPPoE (Point-to-Point Protocol over Ethernet) — протокол передачи фреймов PPP через Ethernet-соединения, который используется в первую очередь для организации подключения с традиционной связкой логин/пароль, поэтому и популярен в xDSL и подобных сервисах, где нет встроенных механизмов аутентификации пользователя.

Подключение по PPPoE реализовано подобно обычному PPP-соединению с использованием демона `pppd`. Данные настройки поэтому будут актуальны и для тех пользователей, которые выходят в Интернет при помощи модемных или DSL-устройств.



#### ВНИМАНИЕ

Для установки PPPoE-соединения потребуется наличие в системе следующих пакетов: `ppp`, `pppoe` и `pppoeconf`. По умолчанию в Linux Mint все необходимое уже есть.

Когда все готово, переходим непосредственно к настройке клиентского PPPoE-соединения. Опять же, вариантов настройки несколько. Самый простой — ручное редактирование двух файлов: `/etc/ppp/pap-secrets` (либо `/etc/ppp/chap-secrets`, если провайдер использует CHAP-аутентификацию, формат файлов одинаков) и `/etc/ppp/peers/dsl-provider`. Имя второго используется в качестве аргумента команды `ppp` и может быть выбрано любое.

В состав Linux Mint для удобства настройки включена утилита `pppoeconf`. Запускаем `sudo pppoeconf`, появится псевдографическое меню настройки (рис. 6.5).

Теперь просто следуем указаниям утилиты, в большинстве случаев подтверждая предлагаемые по умолчанию параметры. Утилита попытается сама найти подходящий Ethernet-интерфейс, используя пакеты PADI (PPPoE Active Discovery

Initiation), с помощью которых идет поиск активных концентраторов доступа. Далее `pppoeconf` сохраняет оригинальный файл `dsl-provider`. После этого вводим логин и пароль, разрешаем, если нужно, автоматическое соединение при загрузке системы, вот и все. Для подключения к Интернету вручную по протоколу PPPoE предлагается использовать скрипт `/usr/bin/pon`. В общем виде вызов `pon` выглядит следующим образом:

```
pon [OPTIONS] [provider] [arguments]
```

В качестве аргумента `provider` указывается файл из каталога `/etc/ppp/peers`, в нашем случае вводим:

```
$ pon dsl-provider
```

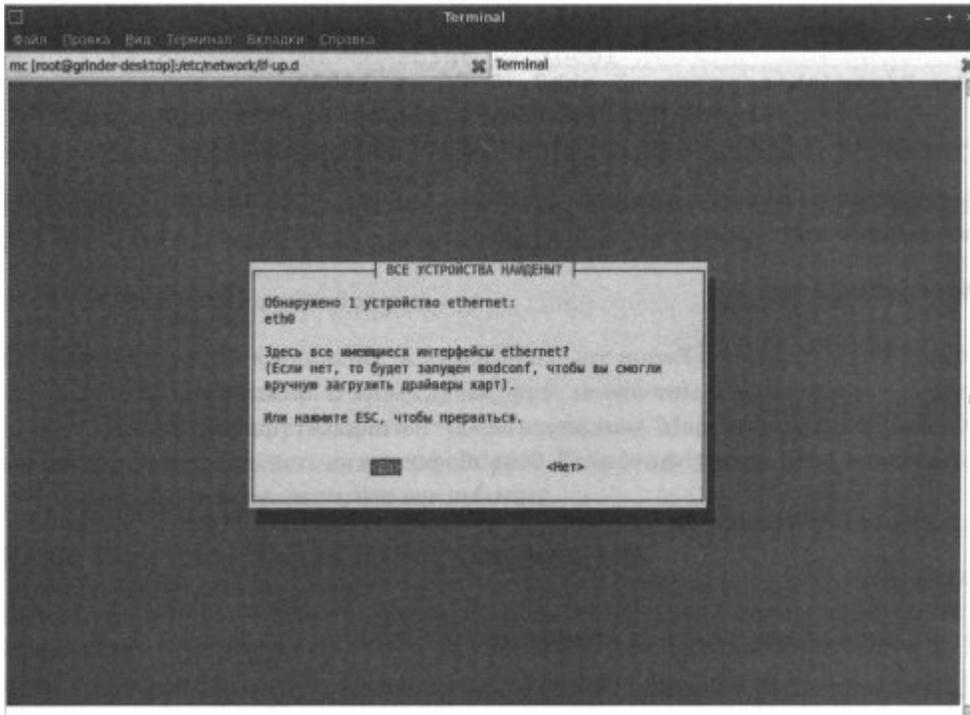


Рис. 6.5. Утилита `pppoeconf`

## Конфигурационные файлы PPPoE

Начнем с самого простого файла `/etc/ppp/pap-secrets` (или `chap-secrets`). Открыв его, мы увидим свои логин и пароль для соединения (результат работы `pppoeconf`) в следующем виде:

```
user * password
```

Здесь достаточно проверить правильность введенных данных или вписать такую строку самому, если `pppoeconf` не использовался. Собственно параметры соединения описываются в `/etc/ppp/peers/dsl-provider`.

**СОВЕТ**

Сохранив файлы `/etc/ppp/ppp-secrets` и `/etc/ppp/peers/dsl-provider` (как, впрочем, и другие описываемые в книге) во время переустановки системы и восстановив их из резервной копии, мы избежим повторной настройки.

Параметров внутри может быть много, остановлюсь лишь на самых популярных.

```
$ sudo gedit /etc/ppp/peers/dsl-provider
```

```
# интерфейс и параметры, нужен если в системе несколько сетевых карт
```

```
pty "/usr/sbin/pppoe -I etho -T 80 -m 1452"
```

```
noipdefault
```

```
# В таблице маршрутизации сделать данное соединение маршрутом по умолчанию
```

```
defaultroute
```

```
replacedefaultroute
```

```
hide-password
```

```
#lcp-echo-interval 30
```

```
#lcp-echo-failure 4
```

```
noauth
```

```
# Восстановить связь в случае разрыва
```

```
persist
```

```
# Использовать максимальный размер передаваемого пакета в 1492 байт
```

```
mtu 1492
```

```
usepeerdns
```

Чтобы PPPoE автоматически устанавливалось при загрузке системы в файл `/etc/network/interfaces`, добавляем:

```
auto dsl-provider
    iface dsl-provider inet ppp
    provider dsl-provider

auto eth0
    iface eth0 inet manual
    pre-up /sbin/ifconfig eth0 up
```

Для контроля PPPoE-соединения используем команду

```
$ ifconfig ppp0
ppp0 Link encap:Point-to-Point Protocol
    inet addr:157.33.34.178 P-t-P:192.168.101.1
Mask:255.255.255.255
```

Плюс за ходом подключения можно следить в системном журнале `/var/log/messages`. Для чего в другой консоли вводим перед началом подключения

```
$ tail -f /var/log/messages
```

Случается ситуация, когда адрес получен, вроде все нормально, но доступ к ресурсам получить невозможно. В этом случае будет нелишним, введя `route -n`, убедиться в том, что маршрутизация настроена правильно. Маршрут, помеченный как *default*, должен показывать на интерфейс `ppp0`. Если это не так, вводим `route add default ppp0` и проверяем, что все работает.

## Настройка нескольких PPPoE-соединений

Бывает, что пользователям требуется работать больше, чем с одним сервером, использующим PAP (например, PPPoE и резервное модемное соединение, VPN и др.). Если логин для каждого сервиса отличается, обычно проблем не возникает, PPP самостоятельно способен разрешить ситуацию. Но многие пользователи предпочитают использовать один и тот же логин для всех серверов, к которым они подключаются. А вот это уже может вызвать проблемы, поскольку демон должен правильно выбрать нужную строку из файла `pap-secrets` для аутентификации. Здесь ему нужно немного помочь. Для начала в файле `/etc/ppp/pap-secrets` заменяем знак астериска `*` условным именем сервера. Примерно так:

```
user  server1      password1
user  server2      password2
```

Создаем два файла, взяв за основу файл `dsl-provider`, и в каждом из них при помощи параметров `name` и `remotename` задаем логин и сервер для подключения:

```
name user

remotename server1
```

Теперь просто указываем нужный файл в качестве аргумента `rop`. Это, кстати, не единственный вариант решения.

В дистрибутиве Linux Mint для настройки PPPoE предлагается более простой вариант с использованием пакета `RP-PPPoE`. Пакет состоит из нескольких скриптов и утилит: `pppoe-relay`, `pppoe-setup`, `pppoe-start`, `pppoe-stop`, `pppoe-connect`, `pppoe-server`, `pppoe-sniff`, `pppoe-status`, `pppoe-wrapper` и `tkpppoe`. Чтобы инициировать соединение, достаточно набрать в консоли `pppoe-start`, чтобы остановить — `pppoe-stop`, чтобы просмотреть статистику — `pppoe-status`, а снова запустить конфигурационный скрипт можно с помощью `pppoe-setup`.

## Подключаемся по PPTP

Настройка VPN-подключения по протоколу PPTP (Point-to-Point Tunneling Protocol) во многом напоминает подключение по PPPoE. До недавнего времени в Linux из-за опасения лицензионных преследований по поводу протокола MPPE отсутствовала нормальная поддержка PPTP, что вызывало проблемы и требовало лишних манипуляций вплоть до пересборки ядра.



### ПРИМЕЧАНИЕ

Полная поддержка протокола PPTP появилась начиная с версии 2.6.13, официальная была начата с 2.6.14, а в 2.6.15 уже был включен модуль шифрования PPP MPPE.

Чтобы процесс был понятней, разберем ручную настройку, а потом посмотрим на GUI. Для поиска в репозитории пакетов, относящихся к PPTP, вводим команду:

```
$ sudo apt-cache search pptp
```

Клиентскую часть обеспечивает проект PPTP Client (<http://pptpclient.sourceforge.net>), который совместим со всеми серверами, работающими по протоколу PPTP: Windows VPN, Linux PopTop, Cisco PIX и некоторыми другими. Пакет уже установлен в Linux Mint. Теперь необходимо указать настройки в конфигурационном файле:

```
$ sudo gedit /etc/ppp/options.pptp
```

```
lock noauth nobsdcomp nodeflate
```

```
# Отключаем ненужные проверки
```

```
refuse-pap
refuse-eap
#refuse-chap
refuse-mschap
persist
# Количество попыток подключения в случае обрыва соединения
maxfail 10
defaultroute
replacedefaultroute
```

В файл `/etc/ppp/chap-secrets` прописываем логин и пароль в следующем виде:

```
user pptp password *
```

Для входа в домен запись должна выглядеть так:

```
domainWuser pptp password *
```

Создаем описание подключения, как это делали чуть выше для PPPoE:

```
$ sudo gedit /etc/ppp/peers/pptp
# Указываем адрес PPTP сервера
pty "pptp 10.100.0.1 --nolaunchpppd"
connect /bin/true
name user
# Идентификатор из второго поля в chap-secrets
remotename pptp
# Подключаем файл настроек
file /etc/ppp/options.pptp
#require-mppe-128
require-mppe-40
ipparam pptp
```

Вот, собственно, и все настройки. Теперь подключаемся

```
$ pon pptp
```

и смотрим вывод команды `ifconfig`. Если соединение установлено и адрес получен, то считаем, что процесс настройки прошел успешно. В случае неудачи выполняем:

```
$ pon pptp debug dump logfd 2 nodetach
```

В результате получим все параметры соединения и лог ошибок. Если есть необходимость в автоматическом подключении при загрузке системы, правим **`/etc/network/interfaces`**:

```
$ sudo mcedit /etc/network/interfaces
```

```
auto tunnel
```

```
iface tunnel inet ppp
```

```
    provider pptp
```



#### ПРИМЕЧАНИЕ

В репозитории Linux Mint доступны еще приложения для управления PPTP-подключением, найти их можно при помощи команды `sudo aptitude search pptp`.

## Межсетевой экран в Linux Mint

В операционной системе Linux в качестве фильтра пакетов используется Netfilter (<http://www.netfilter.org/>), который входит в состав ядра и разрабатывается группой Netfilter Core Team. Для управления его настройками используется утилита командной строки **`iptables`**, разрабатываемая параллельно этим же проектом. Процесс настройки требует некоторого понимания протоколов, используемых при обмене информации в Интернете, и назвать его простым нельзя. Хотя тема настройки **`netfilter/iptables`** уже не является новой, в глобальной сети можно найти не один десяток документов, описывающих как его устройство, так и сами команды. Кроме этого написано несколько хороших графических утилит (KMyFirewall, Firewall Builder, Firestarter и др.), помогающих самостоятельно создавать правила неподготовленному пользователю.

В Ubuntu, начиная с версии 8.04, для управления правилами netfilter используется UFW (Uncomplicated firewall), поэтому процесс настройки здесь выглядит несколько иначе, чем в других дистрибутивах. Позже UFW перекочевал в другие дистрибутивы, в том числе и в Linux Mint. Важно понять, что UFW не заменяет iptables, а является лишь удобной высокоуровневой надстройкой над этой утилитой. Про-

цесс создания новых правил планируется сделать максимально понятным для обычного пользователя. Кроме этого упрощена интеграция приложений с межсетевым экраном. Разработчик может создавать готовые правила, которые будут автоматически активироваться при установке сервиса, разрешая нужные сетевые соединения.

Файлы настроек UFW находятся в каталоге `/etc/ufw`, синтаксис команд внутри несколько проще и понятнее, чем `iptables` (рис. 6.6).

```

Terminal
grinder@grinder-desktop ~ $ cat /etc/ufw/ufw.conf
# /etc/ufw/ufw.conf
#
# Set to yes to start on boot. If setting this remotely, be sure to add a rule
# to allow your remote connection before starting ufw. Eg: 'ufw allow 22/tcp'
ENABLED=yes
#
# Please use the 'ufw' command to set the loglevel. Eg: 'ufw logging medium'.
# See 'man ufw' for details.
LOGLEVEL=low
grinder@grinder-desktop ~ $ sudo cat /etc/ufw/before.rules
#
# rules.before
#
# Rules that should be run before the ufw command line added rules. Custom
# rules should be added to one of these chains:
#   ufw-before-input
#   ufw-before-output
#   ufw-before-forward
#
# Don't delete these required lines, otherwise there will be errors
*filter
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-not-local - [0:0]
# End required lines
#
# allow all on loopback
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-output -o lo -j ACCEPT
#
# quickly process packets for which we already have a connection
-A ufw-before-input -m state --state RELATED,ESTABLISHED -j ACCEPT
-A ufw-before-output -m state --state RELATED,ESTABLISHED -j ACCEPT
#
# drop INVALID packets (logs these in loglevel medium and higher)

```

Рис. 6.6. Правила UFW

По умолчанию UFW всегда отключен, и перед запуском демона его необходимо активировать, заменив в файле `/etc/ufw/ufw.conf` строку

```
ENABLED=no
```

```
ENABLED=yes
```

Иначе при попытке его запуска командой

```
$ sudo /etc/init.d/ufw start
```

получим отказ

```
* Skipping firewall: ufw (not enabled)..
```

Кроме непосредственного редактирования конфигурационных файлов настройки можно производить при помощи консольной утилиты `ufw`. Например, чтобы активировать `ufw` и разрешить ее загрузку, при старте системы можно поступить следующим образом:

```
$ sudo ufw enable
```

```
Firewall started and enabled on system startup
```

Отключить также просто:

```
$ sudo ufw disable
```

```
Firewall stopped and disabled on system startup
```

Существует две глобальные политики: все разрешено и все запрещено. Первая активируется при помощи команды

```
$ sudo ufw default allow
```

Политика по умолчанию `incoming` изменена на `'allow'`

(не забудьте соответствующим образом обновить ваши правила)

Чтобы запретить все подключения, используем

```
$ sudo ufw default deny
```

При помощи `ufw` очень просто разрешить или запретить входящие соединения для сервиса, описанного в `/etc/services`, или конкретного порта/протокола.

В общем случае команда выглядит так:

```
ufw allow|deny [service]
```

Иными словами, чтобы разрешить подключение к веб-серверу, работающему на 80-м порту, поступаем следующим образом:

```
$ sudo ufw allow 80/tcp
```

Чтобы просмотреть правила `iptables` без их активации, добавляем в команду параметр `-dry-run`. Параметр `status` позволит узнать текущие настройки UFW без заглядывания внутрь `iptables`:

```
$ sudo ufw status
```

Удалить разрешение на подключение к выбранному порту можно так же просто:

```
$ sudo ufw delete allow 80/tcp
```

Запрещающее правило создается аналогично разрешающему, только вместо `allow` используем `deny`:

```
$ sudo ufw deny 53
```

Теперь мы блокировали доступ к 53-му порту. При этом если ранее было создано, например, разрешающее правило, которое теперь нужно заменить на блокирующее, то это лучше производить в два этапа. Вначале отключаем первое правило, а затем устанавливаем второе.

Вместо номера порта можно назвать сервис по имени. Смотрим, как называется нужный сервис:

```
$ cat /etc/services | less
```

и включаем его в правило:

```
$ sudo ufw allow ssh
```

В правилах UFW можно задавать IP-адреса (источника и назначения). Например, чтобы разрешить подключение с внутренней сети 192.168.1.0/24, используем:

```
$ sudo ufw allow 192.168.1.0/24
```

или запрещающее правило:

```
$ sudo ufw deny from 10.20.30.40
```

Опционально можно указать порт и протокол. Разрешим подключение по SSH только с одного IP-адреса:

```
$ sudo ufw allow from 192.168.0.20 to any port 22
```

Для включения/отключения регистрации используется команда `logging`. Включаем:

```
$ sudo ufw logging on
```

Файл `/etc/ufw/sysctl.conf` задает некоторые системные переменные (аналог общесистемного `/etc/sysctl.conf`). Например, чтобы разрешить перенаправление пакетов, снимаем комментарий со строки

```
net/ipv4/ip_forward=1
```

В состав Linux Mint включен и графический интерфейс UFW — GUFW (<http://gufw.tuxfamily.org/>, рис. 6.7).

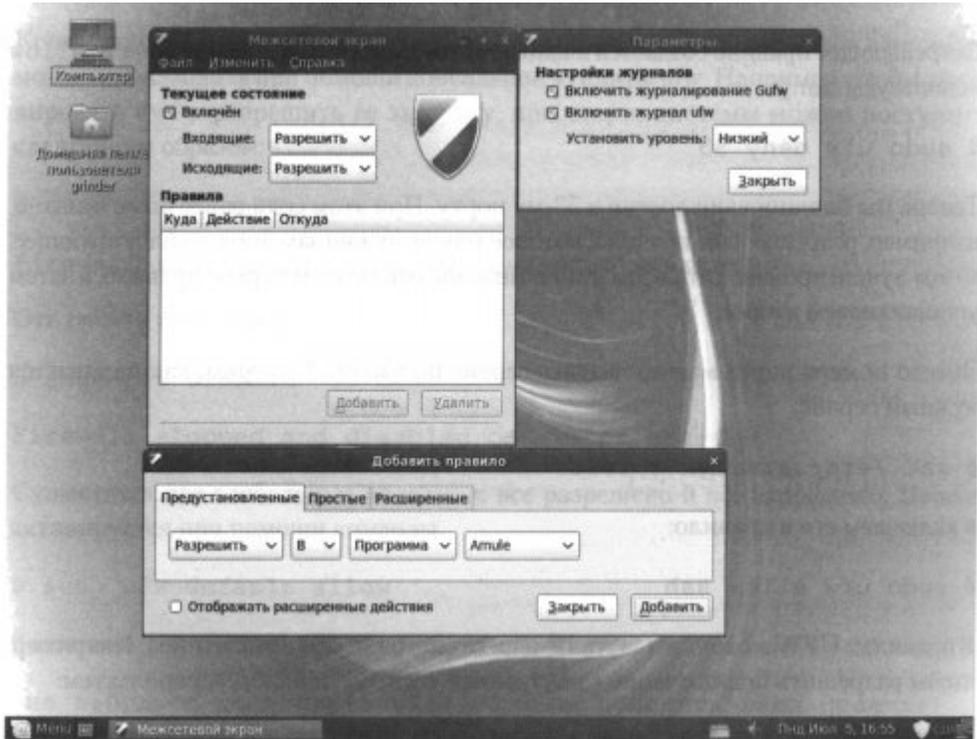


Рис. 6.7. Программа настройки UFW — GUFW

Возможностей у него немного, но их вполне достаточно. С его помощью можно просмотреть установленные правила, создать новые, одним движением мышки запретить или разрешить входящий трафик.

## Программы для работы в Интернете

Традиционно в Linux очень много удобных программ для работы в Интернете, ориентированных как на профессиональное использование, так и на обычного пользователя, консольных и имеющих графический интерфейс. Все, конечно, мы рассматривать не будем, только основные.

## Браузеры

Так уже сложилось, что в Linux нет своего доминирующего браузера, который встраивается в ядро, как в Windows, и пользователю так или иначе приходится с ним

работать. Более того, в некоторых дистрибутивах после установки пользователю часто доступны сразу несколько браузеров и есть возможность выбрать любой, исходя из эстетических соображений, личных пристрастий, задач и системных ресурсов. Еще больше браузеров доступно в репозиториях дистрибутивов. Некоторые из браузеров уже знакомы пользователям Windows — Mozilla и Mozilla Firefox (установлен по умолчанию), Opera и Google Chrome. Функциональность их ничем не отличается от версии Windows, поэтому при желании всегда можно использовать привычный браузер.



#### ПРИМЕЧАНИЕ

Версию Opera для Linux можно скачать по адресу <http://www.opera.com/download/index.dml?platform=linux>.

Команда `sudo aptitude search browser` в Linux Mint покажет большой список приложений, библиотек и всевозможных расширений к браузерам.

## Консольный Интернет

Без консоли, как всегда, не обошлось даже среди браузеров. Многие разработчики используют такие браузеры, чтобы проверить качество своего HTML-кода, любят эти браузеры и системные администраторы. Они экономны в трафике. Сайты некоторых проектов GNU оптимизированы для просмотра именно в консольных браузерах. Из полученного списка можно выделить два самых популярных решения — Lynx и Links. Причем второй имеет версию (пакет `links2`), запускающуюся как в консоли, так и в X Window.

Это легкий браузер Lynx, понимающий практически все стандарты, принятые во Всемирной паутине сегодня. Позволяет обращаться к документам, находящимся в кэше, но не отображает правильно фреймы и таблицы. Вполне пригоден к использованию, особенно на слабых машинах.

Еще одна особенность выделяет данный браузер — он весьма щепетильно относится к интерпретации кода, поэтому если есть серьезные изъяны, то он просто откажется работать. При запуске можно использовать дополнительные ключи, все они описаны в документации. Например, ключ `-dump` выводит как сам документ (без работающих гиперссылок), так и отдельно все гиперссылки, имеющиеся в нем, что позволяет отобразить необходимые файлы и загрузить их, используя один из менеджеров закачки. Параметр `-source` выведет вместо документа его исходный текст.

Браузер Links (рис. 6.8) отлично справляется с выводом таблиц, здесь уже можно активировать меню, которое к тому же русифицировано. В остальном же он практически не отличается от Lynx.

Рис. 6.8. Страница сайта <http://tux.in.ua/> в браузере Links2

## Браузеры для графической среды

В дистрибутиве Linux Mint в качестве основного и единственного браузера предложен Firefox. Но для GNOME штатным является другой браузер — Epirhany (от англ. — «прозрение»). Проект Epirhany (<http://www.gnome.org/projects/epiphany/>) начат в 2002 г. одним из разработчиков браузера Galeon — Марко Песенти Гритти, который ушел ввиду разногласий по поводу будущего этого браузера. Сегодня разработка Galeon фактически прекращена и планируется, что его наиболее существенные возможности, отсутствующие в Epirhany, будут реализованы в виде расширений к последнему. Этот браузер построен на движке Gecko, который используется в Firefox, SeaMonkey, K-Meleon и некоторых других браузерах. Поэтому Epirhany отображает веб-страницы аналогично Firefox, поддерживает работу с Cookies, блокировку всплывающих окон. Хотя есть информация, что в будущем движок, возможно, будет заменен новым — WebKit.



### ПРИМЕЧАНИЕ

WebKit — компонент для просмотра HTML, разработанный компанией Apple для браузера Safari, используемого в Mac OS X. Основан на коде библиотек KHTML и KJS проекта KDE. В самом KDE WebKit появится в версии 4.1.

Как и среда GNOME, идеология разработки Eripanu — простота (рис. 6.9). Этот браузер лишен всей посторонней функциональности (почтового клиента, редактора веб-страниц и др.) и прост в использовании.

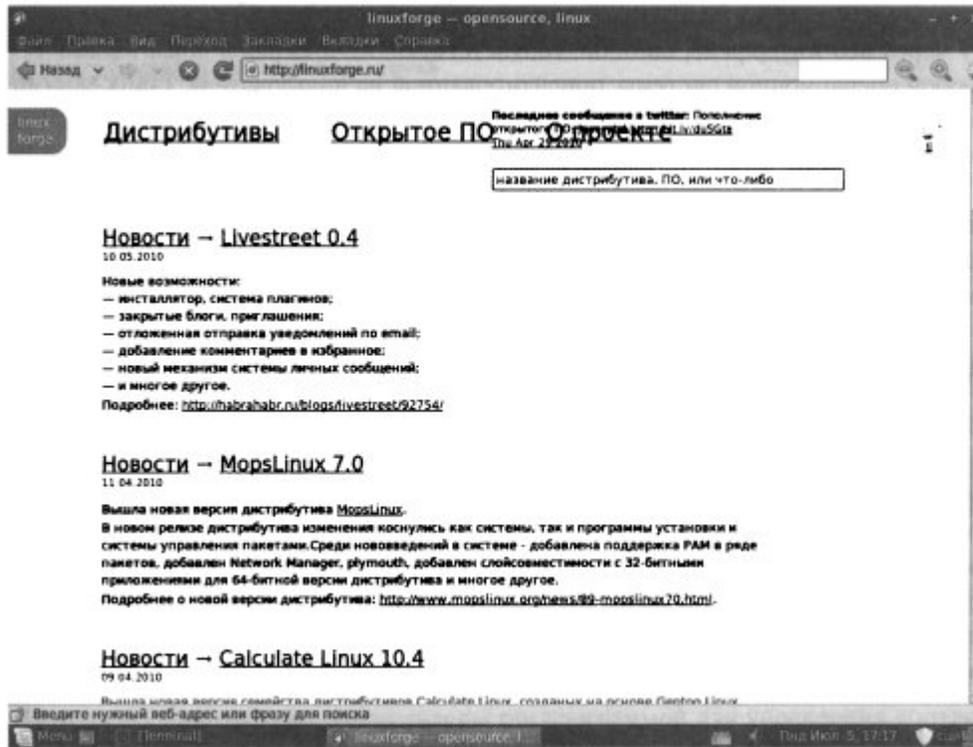


Рис. 6.9. Браузер Eripanu с окном модулей расширения

Вместо иерархического дерева каталогов с закладками, используемого в большинстве современных браузеров, в Eripanu каждой закладке присваивается одна или несколько категорий, после чего дерево категорий формируется автоматически. Наиболее часто посещаемые закладки сохраняются отдельно. Некоторое время придется привыкать, но потом такая схема становится удобной. Помимо этого Eripanu поддерживает так называемые умные закладки: закладке может передаваться в качестве аргумента некоторая текстовая строка, которая будет вставлена в ссылку в указанном месте. Такой подход позволяет, например, использовать закладки для быстрого обращения к поисковым системам. Если такую умную закладку поместить на панель инструментов, рядом с кнопкой закладки появляется текстовое поле ввода.

Как и Firefox, Eripanu поддерживает расширения. Некоторые расширения входят в основной пакет. В Linux Mint часть расширений к Eripanu включена в отдельный пакет `eripanu-extensions`.

## Программы для работы с почтой

Одним из наиболее широко используемых и любимых сервисов Интернета является электронная почта. Поэтому, хотя часть пользователей и предпочитает работу через веб-интерфейс, наличие соответствующих программ обязательно для любой операционной системы, ориентированной на пользователя. Linux Mint не является исключением. Необходимые приложения в этой системе доступны как отдельным пакетом, так и интегрированы в состав другого приложения (Mozilla, Opera) или программы для организации групповой работы. Среда GNOME и KDE имеют интегрированные почтовые клиенты, хотя в некоторых дистрибутивах их заменяют альтернативными приложениями. Например, в Linux Mint установлен Mozilla Thunderbird.

### Mozilla Thunderbird

Кросс-платформенный почтовый клиент и приложение для работы с группами новостей, Mozilla Thunderbird (<http://www.mozillamessaging.com/ru/thunderbird/>) является составной частью проекта Mozilla. Поддерживает все необходимые протоколы: SMTP, POP3, IMAP, NNTP и RSS. Интерфейс Thunderbird, как и браузера Mozilla Firefox, основан на технологии XUL, разработанной в Mozilla Foundation. В результате пользовательский интерфейс не отличается от других приложений, разработанных для конкретной платформы, структура меню и настроек во многом повторяет браузер (рис. 6.10).

При первом запуске программа просит указать учетные данные для работы с почтой (логин и пароль), после чего самостоятельно настраивает подключение.

По умолчанию внешний вид напоминает интерфейс почтового клиента Outlook Express, при желании его можно изменить при помощи тем. По адресу <https://addons.mozilla.org/thunderbird/themes/> доступно большое количество тем, при помощи которых внешний вид Thunderbird можно изменить до неузнаваемости. На этом же сайте доступны различные расширения, увеличивающие функциональность почтового клиента. Некоторые из расширений имеются в репозитории в виде пакетов. Например, чтобы иметь возможность шифровать сообщения, необходимо установить расширение `enigmail`. Хотя темы и расширения можно установить и обновить прямо из окна Thunderbird, достаточно выбрать в меню **Инструменты** нужный пункт.

Поддерживается работа с виртуальными папками. Сообщение может отображаться в нескольких папках на основании заданных пользователем фильтров. При этом письмо существует в единственном экземпляре, не занимая лишнего места на диске. Полученные из разных ящиков сообщения могут храниться в отдельных для каждого ящика папках или в общей для всех. Сообщения можно набирать как в обычном текстовом формате, так и в HTML.

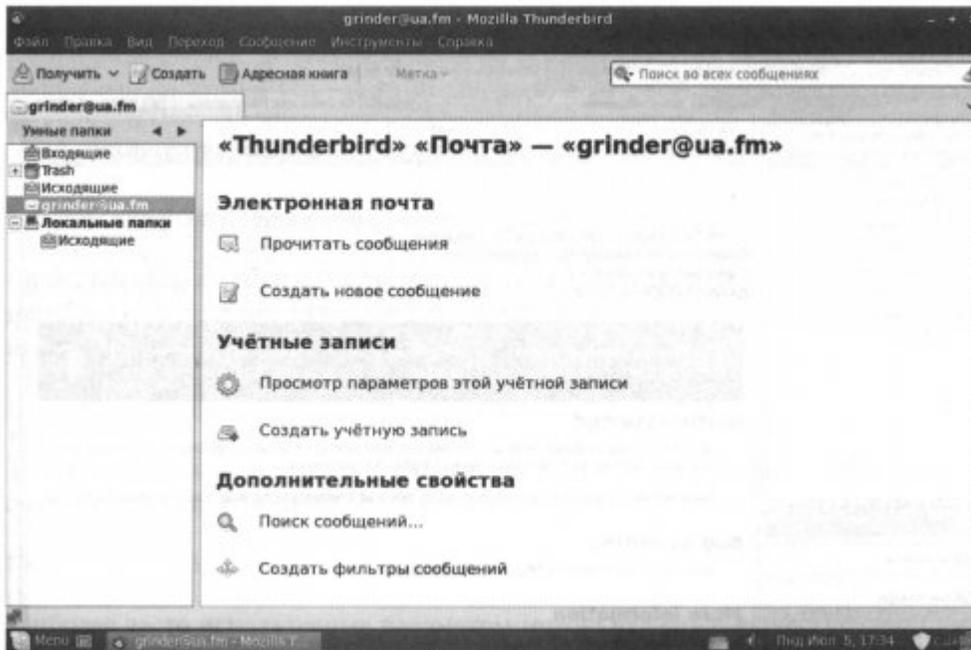


Рис. 6.ю. Окно Mozilla Thunderbird

## Почтовый клиент Evolution

Почтовым клиентом и по совместительству программой для управления контактами и временем для оконной среды GNOME является Evolution (<http://www.gnome.org/projects/evolution/>). Изначально он разработан и поддерживается фирмой Novell, с сентября 2004 г. входит в состав GNOME. Возможна интеграция Evolution в панель GNOME, что позволяет быстро получить доступ к задачам. Очевидно, из-за происхождения идеология Evolution (рис. 6.11) несколько не соответствует простоте, принятой в этой среде.

Хочется также вспомнить Sylpheed (<http://sylpheed.sraoss.jp/en/>) — очень популярный в Linux почтовый клиент, отличающийся легковесностью и быстротой, базирующийся на библиотеках GTK.

Кроме этого в репозитории можно найти несколько утилит для проверки почты, которые выполнены в виде небольших апплетов, которые после запуска прячутся в панели задач и напоминают о себе только при появлении нового сообщения в почтовом ящике. Это GMailWatch (<http://www.employees.org/~ashokn/gmailwatch/>), KCheckGMail (<http://kcheckgmail.sourceforge.net/>), GNOME CheckGmail (<http://checkgmail.sourceforge.net/>), mailtc (<http://mailtc.sourceforge.net/>), GNUbiff (<http://gnubiff.sourceforge.net/>) и др.

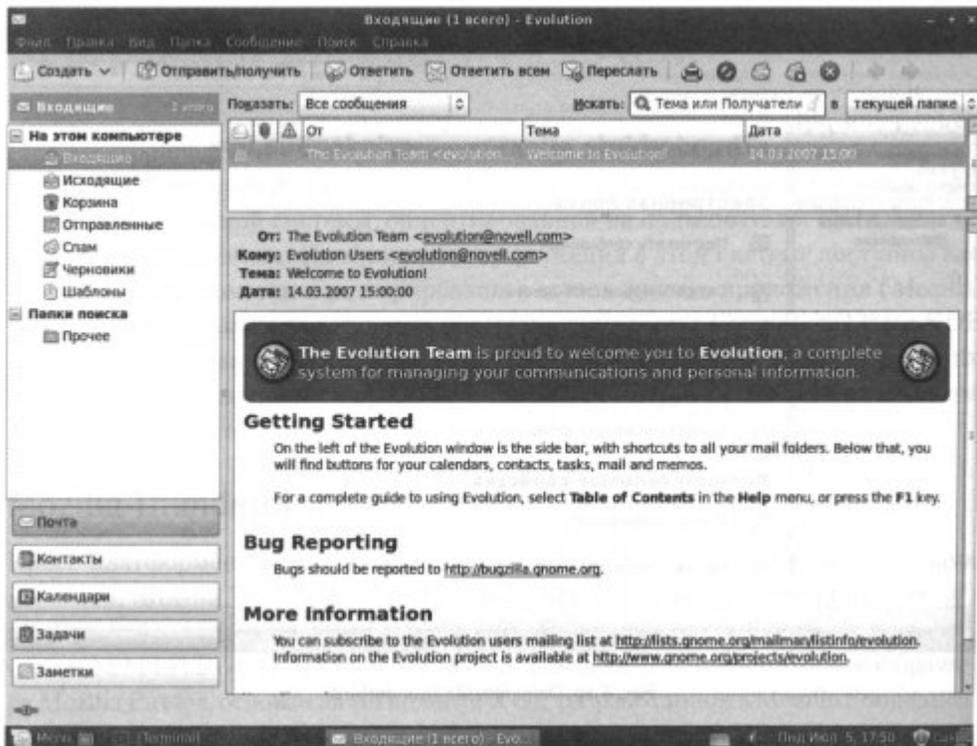


Рис. 6.и. Почтовый клиент Evolution

## Менеджеры зачек

Прежде чем начать с обзора специализированных решений, хотелось бы напомнить, что такие браузеры, как Opera и Firefox, имеют неплохие менеджеры зачек. Кроме этого в Firefox имеется плагин FlashGot (<http://www.flashgot.net/>), по популярности занимающий первое место среди плагинов для этого браузера и умеющий автоматически определять установленные в системе менеджеры зачек, перехватывать ссылки с файлами определенных в настройках расширений и передавать их выбранному менеджеру зачек. Применительно к Linux FlashGot умеет совместно работать с такими менеджерами, как Aria, cURL, Gnome Gwget, KDE Kget и Downloader 4 X.

## Закачка с командной строки

Самым популярным инструментом для закачки файлов и целых сайтов является утилита GNU — wget, которая, как правило, устанавливается по умолчанию во многих дистрибутивах Linux. Wget поддерживает все популярные протоколы

HTTP/HTTPS и FTP, умеет работать через прокси-сервер HTTP. Запустив `wget`, о ней можно вообще забыть. Повлиять на ее работу можно, только убив процесс (например, нажатием `Ctrl + D` в той консоли, в которой запущена программа). Чтобы скопировать весь сайт целиком при помощи `wget`, достаточно ввести команду

```
$ wget http://linuxforge.ru/
```

и через некоторое время в текущем каталоге появится полная копия сайта. Чтобы при обрыве соединения загрузка возобновлялась, добавляем параметр `-c`. Так можно скачать отдельный файл:

```
$ wget -c http://linuxforge.ru/file.rar
```

Параметр `-r` указывает на рекурсивный обход, а `-l` на его глубину. Обойдем рекурсивно все каталоги сайта, на глубину не более 10:

```
$ wget -r -l10 http://linuxforge.ru/
```

Наиболее часто используемые параметры можно указать в конфигурационном файле `.wgetrc`, который создается в домашнем каталоге пользователя. Очень удобно скачивать файлы по расписанию. Например, создадим задачу для загрузки файла в 02:00 (необходимо, чтобы в системе был запущен демон `atd`).

```
$ at 0200
```

```
at> wget -c http://server.org/file.rar
```

```
at> Нажимаем Ctrl+D
```

Еще одна программа — `cURL` (<http://curl.haxx.se/>), которая поддерживает на порядок большее количество протоколов: FTP, FTPS, HTTP, HTTPS, SCP, SFTP, TFTP, TELNET, DICT, FILE и LDAP. Знает о сертификатах SSL, прокси, Cookies, возможна аутентификация пользователей по имени и паролю.

## Графические менеджеры зачек

Одна из самых любимых программ для загрузки файлов с графическим интерфейсом — `D4X` (`Downloader for X`). Хотя у `D4X`, наверное, не самый удобный GTK+ интерфейс (рис. 6.12), но многие пользователи думают совсем иначе. Те, кто ранее пользовался `Download Master`, найдут много знакомого в `D4X`, все на своих местах и там, где ожидаешь найти нужную опцию. Предусмотрен также запуск программы без графического интерфейса. `D4X` умеет закрывать файлы по протоколам FTP и HTTP, в том числе рекурсивно, ведет отдельный каталог для каждой загрузки, работает через прокси-сервер.

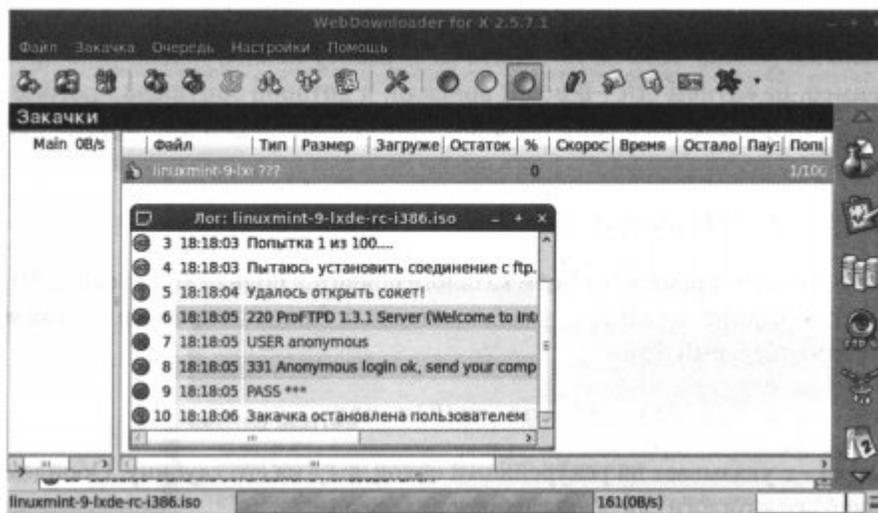


Рис. 6.12. Окно D4X

Среди полезных возможностей можно выделить встроенный планировщик для выполнения различных операций по расписанию, выборочный фильтр при закачке по HTTP, FTP-поиск для нахождения ближайшего сервера, ограничение скорости закачки и скачивание файла в несколько потоков для увеличения скорости. Реализована функция дистанционного управления программой, добавление файлов перетаскиванием, возможность ввода паролей для доступа к сайтам.

## Программы для работы с ICQ, IRC и Jabber

Программы для мгновенного обмена сообщениями являются не менее популярными, чем электронная почта и другие сервисы Интернета. После установки в меню Linux Mint находим кросс-платформенный клиент для IRC-сетей XChat (рис. 6.13) и Pidgin.

Pidgin (<http://pidgin.im/>) — это целый комбайн, который является универсальным средством общения, поддерживающим такие протоколы, как AIM, MSN, Yahoo!, XMPP (Jabber), ICQ, IRC, SILC, SIP/SIMPLE, Novell Group Wise, Lotus Sametime, Bonjour, Zephyr, MySpaceIM, Gadu-Gadu и QQ. Написанная с использованием библиотек GTK+, эта программа является мультиплатформенной, она доступна для большинства UNIX-подобных систем. Есть версия и для Windows. Хорошо интегрируется в GNOME и KDE. Именно Pidgin предлагается в качестве штатного средства работы в IM-сетях в дистрибутиве Ubuntu. Интерфейс переведен и внешним видом напоминает классический ICQ, поэтому проблем с освоением быть не должно.

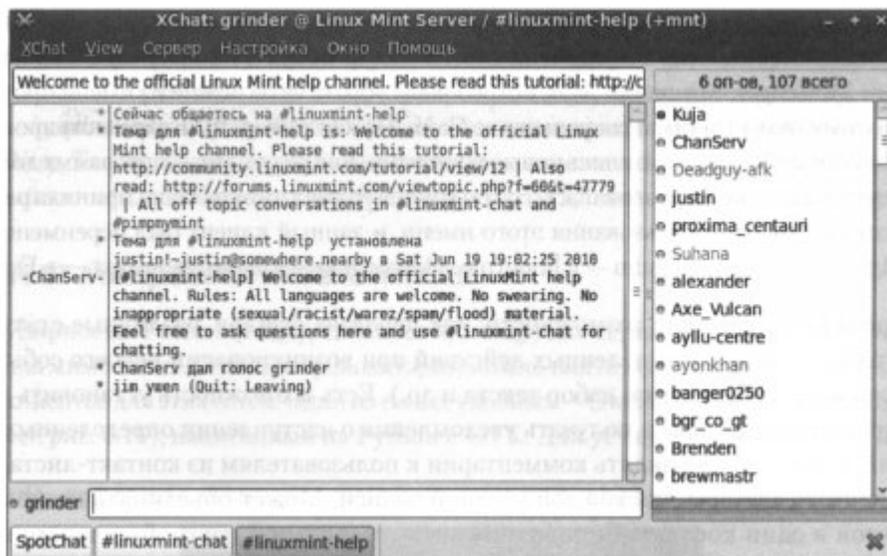


Рис. 6.13. Окно XChat

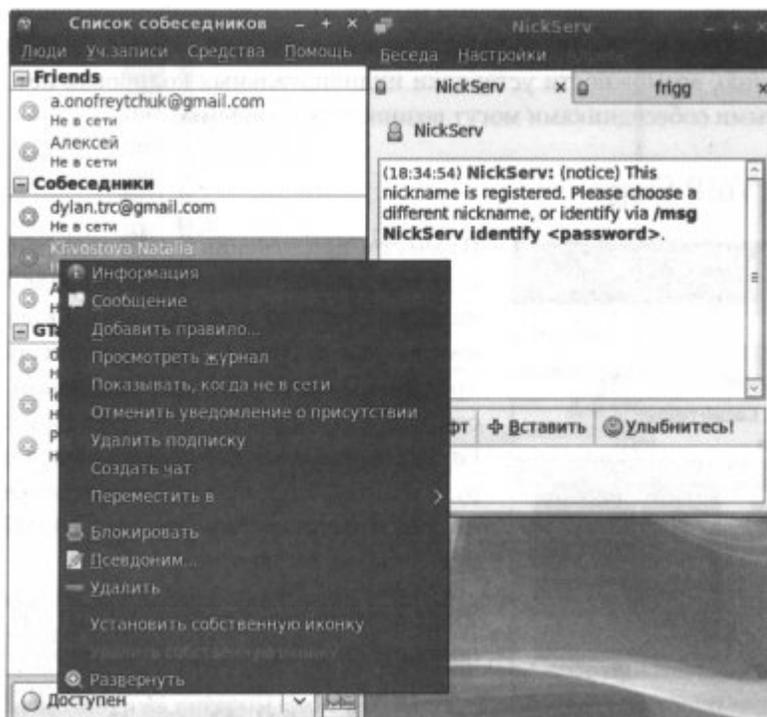


Рис. 6.14. Окно Pidgin

Первые версии пользователи не очень любили, но в этом не было вины разработчиков. На начальном этапе своего развития проект Pidgin носил длинное имя GTK+ AOL Instant Messenger. Компания AOL потребовала убрать ее название из имени продукта, поэтому название было сокращено до Gaim. Но как только AOL зарегистрировала торговую марку AIM, появились новые претензии. Как результат — программа долгое время практически не развивалась. Затем группа разработчиков Gaim приняла решение отказаться от использования этого имени, и данный клиент был переименован в Pidgin, библиотека libgaim — в libpurple, а консольный клиент gaim-text — в Finch.

Поддерживаются такие возможности, как передача файлов, различные статусы, программирование определенных действий при возникновении некоего события (отключение собеседника, набор текста и др.). Есть возможность установить слежение за собеседником и получать уведомления о наступлении определенных событий. Позволяет добавлять комментарии к пользователям из контакт-листа, устанавливать аватары для каждой учетной записи. Может объединять несколько номеров в один контакт. Есть возможность раздельной настройки параметров конфиденциальности для всех учетных записей. Функциональность легко расширяется за счет установки дополнительных плагинов, позволяющих использовать проверку орфографии, расширенную систему оповещения, ведения бесед во вкладках одного окна, управление жестами мышки. После установки все модули отключены. Чтобы выбрать и активировать нужные, следует зайти в **Сервис • Модули**. К сожалению, возможности установки индивидуальных кодировок нет, поэтому с некоторыми собеседниками могут возникнуть проблемы.

## Клиент VoIP Ekiga



Рис. 6.15. Окно программы Ekiga

В качестве приложения для IP-телефонии и проведения видеоконференций в среде GNOME наиболее удобно использовать Ekiga (<http://www.ekiga.org/>, рис. 6.15). Первая версия этой программы была написана Демиеном Сандрасом в качестве дипломной работы и называлась GnomeMeeting, он сегодня является одним из руководителей проекта. В настоящее время Ekiga является частью рабочей среды GNOME, в которую легко интегрируется.

Ekiga поддерживает протоколы SIP и H.323 (при помощи OpenH323) и способна взаимодействовать с другими SIP-совместимыми клиентами, в том числе и Microsoft NetMeeting. Поддерживает множество аудио- и видеокодеков высокого качества. При наличии Novell Evolution можно обменивать-

ся листами контактов. Менеджер учетных записей позволяет легко работать с несколькими серверами SIP. При первом запуске вас встретит мастер первоначальной настройки, который поможет сконфигурировать Ekiga, автоматически обнаруживаются звуковые карты, устройства Video4Linux, видеокамеры, подключенные к порту FireWire. Так же, как и большинство остальных клиентов, Ekiga поддерживает текстовый чат между собеседниками SIP, вывод статуса, поиск пользователей и др.

## Клиент микроблогинга Gwibber

Популярность Twitter (<http://twitter.com/>) и других сервисов микроблогинга не прошла мимо Linux. Сегодня в репозитории можно найти большое количество удобных клиентов для этих сетей. Один из самых удобных — Gwibber (<http://live.gnome.org/Gwibber>, рис. 6.16), написанный на Python и GTK. Для установки следует ввести

```
$ sudo aptitude install gwibber gwibber-themes
```



Рис. 6.16. Программа микроблогинга Gwibber

Поддерживает сервисы микроблогинга Twitter и Jaiku, фотосервис flickr, социальные сети Digg и Facebook, умеет следить за RSS- и Atom-каналами сайтов. После настройки программа прячется в панели задач и напоминает о себе при появлении нового сообщения, щелкнув на котором кнопкой мыши, можно сразу написать ответ.

В клиенте реализован поиск по ключевым словам. Гиперссылка автоматически укорачивается. Возможно изменение внешнего вида при помощи тем оформления.

**Яремчук Сергей Акимович**  
**Linux Mint на 100 %**

Заведующий редакцией  
Руководитель проекта  
Ведущий редактор  
Литературный редактор  
Художник  
Корректоры  
Верстка

*А. Буглак*  
*К. Галицкая*  
*Е. Каляева*  
*Ю. Крайцова*  
*Л. Адуевская*  
*О. Андросик, В. Субот*  
*А. Барцевич, О. Махлина*

Подписано в печать 14.09.10. Формат 70x100/16. Усл. п. л. 19,35. Тираж 1500. Заказ 23837.

ООО «Лидер», 194044, Санкт-Петербург, Б. Сампсониевский пр., 29а.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 95 3005 — литература учебная.

Отпечатано по технологии СtP в ОАО «Печатный двор» им. А. М. Горького.  
197110, Санкт-Петербург, Чкаловский пр., 15.

Эта книга позволит вам освоить на 100% Linux Mint — популярнейший дистрибутив операционной системы GNU/Linux. «From freedom came elegance» («От свободы к эlegантности») — девиз данного дистрибутива, который совместим с оригинальным Ubuntu и вместе с тем избавлен от многих его недостатков и недочетов, а также оснащен оригинальным интерфейсом и набором собственных приложений, таких как mintinstall, mintUpdate, mintMenu и т. д. Данное руководство рассчитано, в первую очередь, на читателей, не имеющих опыта работы в этой операционной системе. Тем не менее содержащиеся в книге обширный справочный материал и советы профессионалов будут полезны и опытным пользователям ОС Linux. Многие темы и вопросы, раскрываемые в издании, заинтересуют также пользователей дистрибутивов, родственных Linux Mint, в частности Linux Ubuntu, Debian GNU/Linux.

С помощью «Linux Mint на 100 %» вы с легкостью освоите самую эlegантную ОС современности и, безусловно, полюбите свежий мятный вкус свободного ПО!

Тема: **Операционные системы/Linux**

Уровень пользователя: **начинающий/опытный**



**Заказ книг:**

197198, Санкт-Петербург, а/я 127  
тел.: (812) 703-73-74, postbook@piter.com  
61093, Харьков-93, а/я 9130  
тел.: (057) 758-41-45, 751-10-02, piter@kharkov.piter.com

**www.piter.com** — вся информация о книгах и веб-магазин