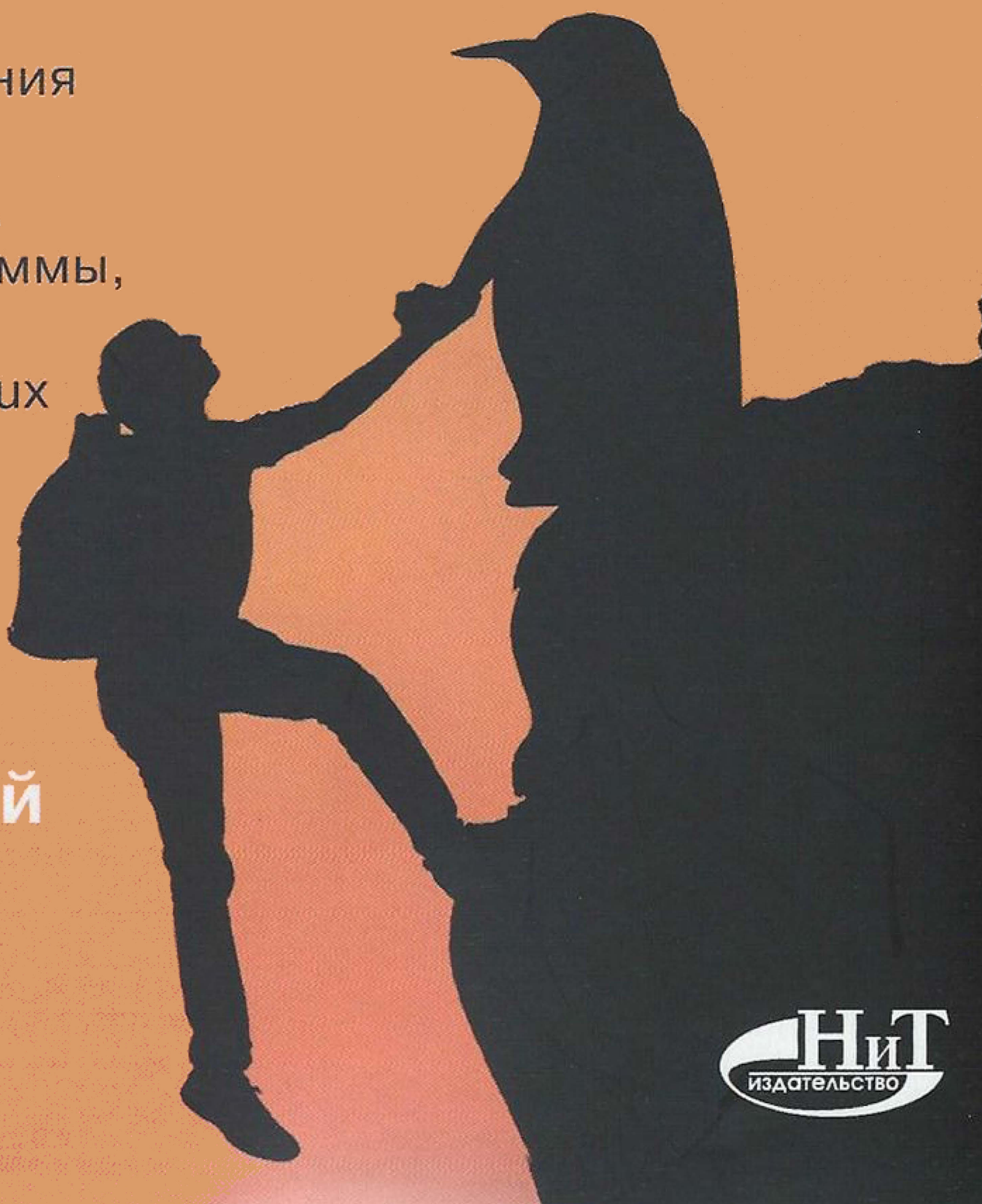
Донцов В. П., Сафин И. В.

LINUX

на примерах

- От основ работы до продвинутого администрирования
- Офисные пакеты, полезные программы, Интернет, безопасность Linux



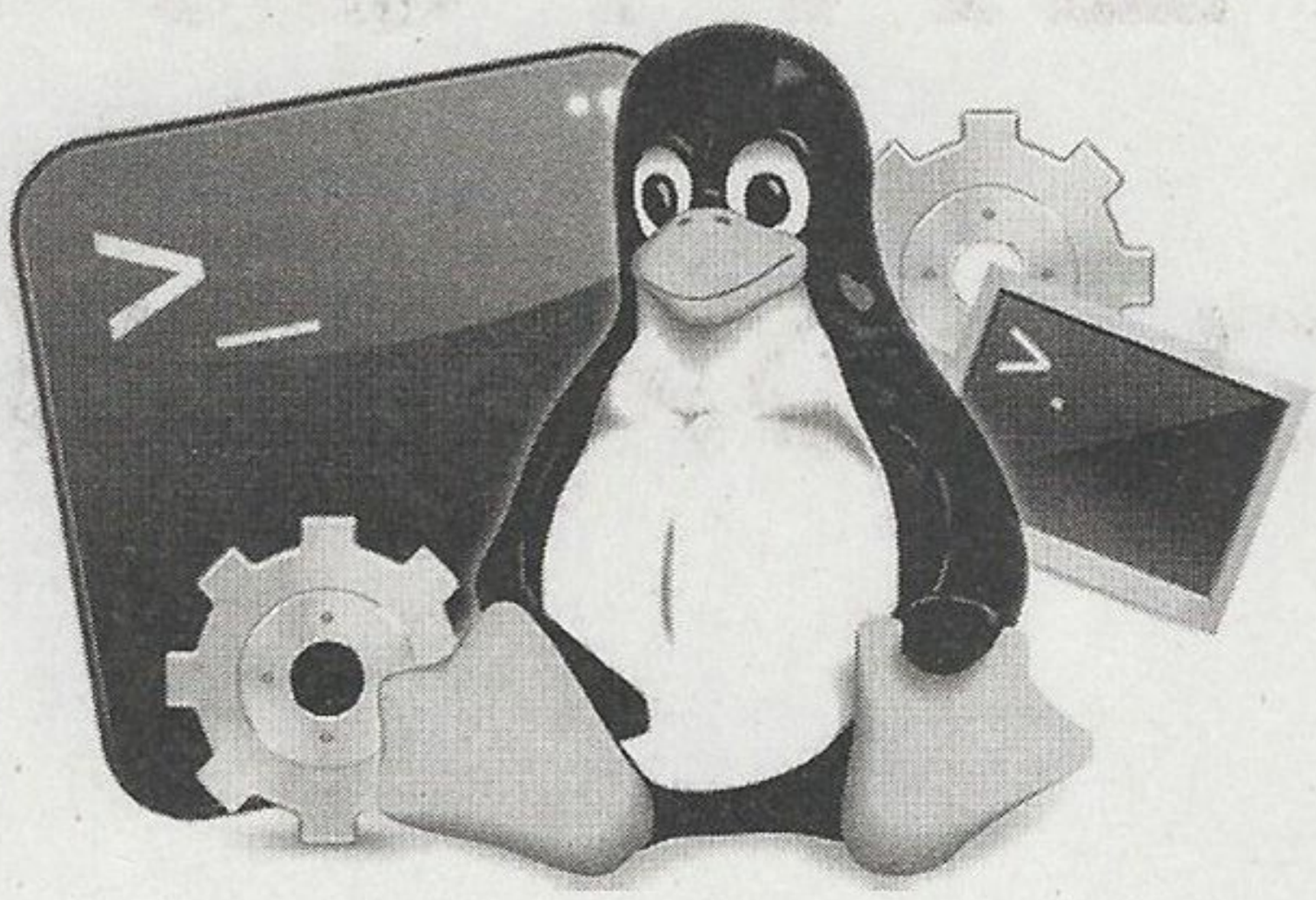
Практический курс по работе в Linux

НиТ издательство

СЕРИЯ — **ПРОСТО О СЛОЖНОМ** — СЕРИЯ

"Наука и Техника"

Санкт-Петербург



Донцов В. П., Сафин И. В.

Linux

на примерах

"Наука и Техника"

Санкт-Петербург

УДК 004.451
ISBN 978-5-94387-742-1

Донцов В. П., Сафин И. В.

LINUX НА ПРИМЕРАХ — СПб.: "Наука и Техника", 2017. — 352 с., ил.

Серия «Просто о сложном»

Данная книга является отличным практическим руководством по работе в операционной системе Linux и ее администрированию. На большом количестве практических примеров показано выполнение огромного количества действий и задач, которые могут понадобиться пользователю Linux.

Книга может использоваться как практический справочник и как самоучитель для начинающих. Изложение ведется с учетом самых разных дистрибутивов Linux. Лучший выбор для эффективного изучения и использования Linux.

ISBN 978-5-94387-742-1

978-5-94387-7421

Контактные телефоны издательства:
(812) 412 70 26

Официальный сайт: www.nit.com.ru

© Донцов В.П., ПРОКДИ РГ
© Наука и техника (оригинал-макет)

ГЛАВА 1. ЗНАКОМСТВО С LINUX 23

ГЛАВА 2. СРАЗУ ПОСЛЕ УСТАНОВКИ 47

ГЛАВА 3. КОМАНДНАЯ СТРОКА..... 61

ГЛАВА 4. РАБОТА С ФАЙЛАМИ И КАТАЛОГАМИ 77

ГЛАВА 5. УПРАВЛЕНИЕ ПАКЕТАМИ
И ПРОЦЕССАМИ..... 101

ГЛАВА 6. ПОПУЛЯРНЫЕ ПРОГРАММЫ В LINUX..... 121

ГЛАВА 7. УПРАВЛЕНИЕ АППАРАТНЫМИ
СРЕДСТВАМИ 153

ГЛАВА 8. РАБОТА С ТЕКСТОМ 173

ГЛАВА 9. СЦЕНАРИИ КОМАНДНОЙ СТРОКИ 187

ГЛАВА 10. ПОЛЬЗОВАТЕЛИ И ГРУППЫ.
ОСНОВЫ БЕЗОПАСНОСТИ 213

ГЛАВА 11. УПРАВЛЕНИЕ УЧЕТНЫМИ
ЗАПИСЯМИ ПОЛЬЗОВАТЕЛЕЙ 225

ГЛАВА 12. ПРАВА ДОСТУПА К ФАЙЛАМ
И КАТАЛОГАМ 249

ГЛАВА 13. УПРАВЛЕНИЕ СЕТЕВЫМИ
ПОДКЛЮЧЕНИЯМИ 257

ГЛАВА 14. КАК ЗАГРУЖАЕТСЯ LINUX? 277

ГЛАВА 15. СИСТЕМЫ ИНИЦИАЛИЗАЦИИ 295

ГЛАВА 16. МОНИТОРИНГ СИСТЕМЫ 317

ГЛАВА 17. ЖУРНАЛИРОВАНИЕ..... 331

Содержание

ВВЕДЕНИЕ **18**
Что такое Linux? 18
Аудитория 18
Необходимые знания 20

ГЛАВА 1. ЗНАКОМСТВО С LINUX **23**

1.1. КРАТКАЯ ИСТОРИЯ LINUX **24**

1.2. ИЗ ЧЕГО СОСТОИТ ОПЕРАЦИОННАЯ СИСТЕМА. КОМПОНЕНТЫ СИСТЕМЫ **25**

1.3. ДИСТРИБУТИВЫ **27**
1.3.1. Debian 28
1.3.2. Fedora 28
1.3.3. CentOS 29
1.3.4. Ubuntu 29

1.4. ГРАФИЧЕСКИЕ ИНТЕРФЕЙСЫ **30**
1.4.1. Графическая среда GNOME 31
1.4.2. Графическая среда KDE 32
1.4.3. Оболочка LXDE 33
1.4.4. Оболочка Xfce 34
1.4.5. Оболочка Cinnamon 35
1.4.6. Оболочка Mate 36
1.4.7. E17 (Enlightenment) 37
1.4.8. Оболочка Razor-qt 38

1.5. ЛИЦЕНЗИЯ GPL **40**

1.6. РОЛИ ОПЕРАЦИОННОЙ СИСТЕМЫ **41**
1.6.1. Что такое сервер? Linux на сервере сети 41
1.6.2. Рабочая станция или домашний компьютер. Настольные и

портативные устройства 43
1.6.3. Настольные и портативные устройства 45

ГЛАВА 2. СРАЗУ ПОСЛЕ УСТАНОВКИ 47

2.1. ВХОД В СИСТЕМУ И ЗАВЕРШЕНИЕ РАБОТЫ 49

2.2. ПЯТЬ ВЕЩЕЙ, КОТОРЫЕ НУЖНО СДЕЛАТЬ СРАЗУ ПОСЛЕ УСТАНОВКИ 51

2.2.1. Настройка сети 51
2.2.2. Установка кодеков 52
2.2.3. Тонкая настройка графического интерфейса 54
2.2.4. Установка новых тем оформления графического интерфейса 57
2.2.5. Установка средства очистки системы 58

ГЛАВА 3. КОМАНДНАЯ СТРОКА 61

3.1. ВВЕДЕНИЕ В КОМАНДНУЮ СТРОКУ 62

3.2. ДОСТУП К КОМАНДНОЙ СТРОКЕ 66

3.3. ПОДСКАЗКА В КОМАНДНОЙ СТРОКЕ И ВЫЗОВ СПРАВКИ 68

3.4. ПЕРЕНАПРАВЛЕНИЕ ВВОДА/ВЫВОДА 70

3.5. НЕКОТОРЫЕ ПОЛЕЗНЫЕ КОМАНДЫ 72

ГЛАВА 4. РАБОТА С ФАЙЛАМИ И КАТАЛОГАМИ 77

4.1. ВВЕДЕНИЕ В ФАЙЛОВУЮ СИСТЕМУ 78

4.2. ФАЙЛОВЫЕ СИСТЕМЫ LINUX 80

4.3. СТАНДАРТНЫЕ КАТАЛОГИ КОРНЕВОЙ ФАЙЛОВОЙ СИСТЕМЫ 80

4.4. ФАЙЛЫ УСТРОЙСТВ 82

4.5. МОНТИРОВАНИЕ 83

4.6. ФАЙЛОВЫЙ МЕНЕДЖЕР NAUTILUS 86

4.6.1. Элементы окна Nautilus 86
4.6.2. Создание папки 89
4.6.3. Основные операции над файлами и папками 90
4.6.4. Поиск файлов 91
4.6.5. Создание архивов 91

4.7. КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И ПАПКАМИ 92
4.7.1. Команды для работы с файлами 92
4.7.2. Команды для работы с каталогами 95
4.7.3. Команда ln 97
4.7.4. Поиск файлов в консоли 98

ГЛАВА 5. УПРАВЛЕНИЕ ПАКЕТАМИ И ПРОЦЕССАМИ 101

5.1. ЗАПУСК ПРОГРАММ 102

5.2. УПРАВЛЕНИЕ ПАКЕТАМИ 105
5.2.1. Введение в пакеты 105
5.2.2. Менеджер пакетов dnf 107
Синтаксис и параметры 107
Конфигурационные файлы dnf 108
Очистка кэша 109
5.2.3. Менеджер пакетов apt 110
Синтаксис и команды 110
Конфигурационные файлы 111
5.2.4. Графический менеджер Synaptic 111
5.2.5. Ubuntu Software 114

5.3. УПРАВЛЕНИЕ ПРОЦЕССАМИ 115
5.3.1. Команда ps 115
5.3.2. Команда nice 117
5.3.3. Команда kill 117
5.3.4. Команда top 118

ГЛАВА 6. ПОПУЛЯРНЫЕ ПРОГРАММЫ В LINUX..... 121

6.1. ИНТЕРНЕТ-ПРИЛОЖЕНИЯ 122

6.1.1. Браузер Firefox 122

6.1.2. Почтовый клиент Thunderbird 125

6.1.3. Устанавливаем Skype и FileZilla 127

Установка Skype 127

Установка FileZilla 133

6.1.4. BitTorrent-клиент Transmission 133

6.2. МУЛЬТИМЕДИА-ПРИЛОЖЕНИЯ 137

6.3. ОФИСНЫЕ ПРИЛОЖЕНИЯ 141

6.4. ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С ГРАФИКОЙ 142

6.4.1. Shotwell Photo Manager 142

6.5. СЕРВЕРНЫЕ ПРИЛОЖЕНИЯ. СОБСТВЕННЫЙ ВЕБ-СЕРВЕР..... 147

ГЛАВА 7. УПРАВЛЕНИЕ АППАРАТНЫМИ СРЕДСТВАМИ..... 153

7.1. ПРОЦЕССОР И МАТЕРИНСКАЯ ПЛАТА 154

7.2. УПРАВЛЕНИЕ ДИСКАМИ 157

7.2.1. Различные интерфейсы 157

7.2.2. Добавление еще одного жесткого диска и его разметка 158

7.2.3. Сменные и оптические диски..... 166

7.3. МОНИТОРЫ И ГРАФИЧЕСКИЕ АДАПТЕРЫ 167

7.4. USB-УСТРОЙСТВА..... 170

ГЛАВА 8. РАБОТА С ТЕКСТОМ 173

8.1. ТЕКСТОВЫЕ РЕДАКТОРЫ 174

8.2. СТАТИСТИКА: ПОДСЧЕТ СИМВОЛОВ, СЛОВ, СТРОК..... 179

8.3. ПОТОКОВЫЙ ТЕКСТОВЫЙ РЕДАКТОР 179

8.4. СРАВНЕНИЕ ДВУХ ФАЙЛОВ 180

8.5. ТЕКСТОВЫЙ ФИЛЬТР 182

8.6. ФОРМАТИРОВАНИЕ ТЕКСТА 183

8.7. ЗАМЕНА СИМВОЛОВ ТАБУЛЯЦИИ ПРОБЕЛАМИ 183

8.8. РАЗБИВКА ТЕКСТА НА СТОЛБЦЫ 184

8.9. ВЫРАВНИВАНИЕ ТЕКСТА ПО ПРАВОМУ КРАЮ 184

8.10. ВЫВОД НАЧАЛЬНЫХ И ПОСЛЕДНИХ СТРОК ФАЙЛА 184

8.11. СОРТИРОВКА ФАЙЛОВ 185

8.12. РАЗБИВКА ТЕКСТОВЫХ ФАЙЛОВ НА НЕСКОЛЬКО ЧАСТЕЙ 185

ГЛАВА 9. СЦЕНАРИИ КОМАНДНОЙ СТРОКИ 187

9.1. ДОСТУПНЫЕ ИНТЕРПРЕТАТОРЫ КОМАНД 188

9.2. НАСТРОЙКА BASH 189

9.3. ВВЕДЕНИЕ В СЦЕНАРИИ BASH 192

9.4. ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ 194

9.5. СОБСТВЕННЫЕ ПЕРЕМЕННЫЕ 195

9.6. ПЕРЕДАЧА АРГУМЕНТОВ СЦЕНАРИЮ 195

9.7. УСЛОВНЫЕ ОПЕРАТОРЫ 196

9.8. ЦИКЛЫ 199

9.9. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ 201

9.10. ПРАКТИЧЕСКИЕ ПРИМЕРЫ 201

9.10.1. Переименование файлов 201

9.10.2. Мониторинг системы 203

9.10.3. Резервное копирование 204

ГЛАВА 10. ПОЛЬЗОВАТЕЛИ И ГРУППЫ. ОСНОВЫ БЕЗОПАСНОСТИ 213

10.1. УЧЕТНЫЕ ЗАПИСИ ПОЛЬЗОВАТЕЛЕЙ 214

10.2. ПОЛЬЗОВАТЕЛЬ ROOT 215

10.2.1. Максимальные права - это хорошо или плохо? 215

10.2.2. Команды sudo и su 216

10.2.3. Как включить root в Ubuntu 218

10.3. ФАЙЛ /ETC/PASSWD 218

10.4. ВВЕДЕНИЕ В ГРУППЫ 221

ГЛАВА 11. УПРАВЛЕНИЕ УЧЕТНЫМИ ЗАПИСЯМИ ПОЛЬЗОВАТЕЛЕЙ 225

11.1. СОЗДАНИЕ НОВЫХ УЧЕТНЫХ ЗАПИСЕЙ 226

11.1.1. Выбор надежного пароля 227

11.1.2. Управление группами 229

11.1.3. Создание учетной записи с помощью GUI-инструмента 230

11.1.4. Создание учетной записи с помощью командной строки 235

11.2. МОДИФИКАЦИЯ УЧЕТНОЙ ЗАПИСИ 238

11.2.1. Графические инструменты 238

11.2.2. Команды usermod и passwd 239

11.2.3. Команда visudo 242

11.3. УДАЛЕНИЕ УЧЕТНОЙ ЗАПИСИ 244

11.4. МОДУЛИ АУТЕНТИФИКАЦИИ 245

11.4.1. Знакомство с PAM 245

11.4.2. Установка минимальной длины пароля 245

11.4.3. Ограничиваем доступ к системе 246

11.4.4. Разрешаем регистрацию только в рабочее время 247

ГЛАВА 12. ПРАВА ДОСТУПА К ФАЙЛАМ И КАТАЛОГАМ 249

12.1. ВЛАДЕЛЕЦ И ПРАВА ДОСТУПА 250

12.2. КОМАНДА CHOWN 251

12.3. КОМАНДА CHMOD 251

12.4. СПЕЦИАЛЬНЫЕ ПРАВА ДОСТУПА 254

12.5. КОМАНДА CHATTR 254

ГЛАВА 13. УПРАВЛЕНИЕ СЕТЕВЫМИ ПОДКЛЮЧЕНИЯМИ 257

13.1. ГРАФИЧЕСКИЕ СРЕДСТВА НАСТРОЙКИ СЕТЕВОГО ПОДКЛЮЧЕНИЯ 258

13.2. УСТАНОВКА СТАТИЧЕСКОГО IP-АДРЕСА И ДРУГИХ ПАРАМЕТРОВ СЕТИ 259

13.3. ПОДКЛЮЧЕНИЕ К WI-FI СЕТИ 262

13.4. ПОЛУЧЕНИЕ ИНФОРМАЦИИ О СОЕДИНЕНИИ 263

13.5. ИЗМЕНЕНИЕ ИМЕНИ КОМПЬЮТЕРА 264

13.6. КОМАНДА IFCONFIG..... 264

13.7. СЕТЕВЫЕ ИНТЕРФЕЙСЫ В LINUX 267

13.8. ВВЕДЕНИЕ В МАРШРУТИЗАЦИЮ 269

13.8.1. Таблица маршрутизации 269

13.8.2. Изменение и сохранение таблицы маршрутизации..... 272

13.9. УТИЛИТЫ ДЛЯ ДИАГНОСТИКИ СЕТИ 275

ГЛАВА 14. КАК ЗАГРУЖАЕТСЯ LINUX? 277

14.1. НАЧАЛЬНАЯ ЗАГРУЗКА КОМПЬЮТЕРА И ЗАГРУЗЧИКИ LINUX 278

14.2. КОГДА НУЖНО ИЗМЕНЯТЬ КОНФИГУРАЦИЮ ЗАГРУЗЧИКА 279

14.3. УСТАНОВКА ЗАГРУЗЧИКА 280
14.4. КОНФИГУРАЦИЯ ЗАГРУЗЧИКА GRUB2 280
14.5. ВЫБОР МЕТКИ ПО УМОЛЧАНИЮ 288
14.6. ДОБАВЛЯЕМ ЗАГРУЗОЧНУЮ МЕТКУ WINDOWS 288
14.7. ЗАГРУЗКА С ISO-ОБРАЗА 289
14.8. ПАРОЛИ И ПАРАНОЙЯ 289
14.9. КАК ПЕРЕДАТЬ ПАРАМЕТР ЯДРА 292

ГЛАВА 15. СИСТЕМЫ ИНИЦИАЛИЗАЦИИ 295

15.1. ЧТО ТАКОЕ СИСТЕМА ИНИЦИАЛИЗАЦИИ 296
15.2. КЛАССИЧЕСКАЯ СИСТЕМА ИНИЦИАЛИЗАЦИИ SYSV INIT 298
15.2.1. Файл конфигурации inittab 298
15.2.2. Сценарии запуска служб 302
15.3. СОВРЕМЕННАЯ СИСТЕМА ИНИЦИАЛИЗАЦИИ SYSTEMD 303
15.3.1. Как она работает? 303
15.3.2. Настраиваем systemd 305
15.3.3. Цели 308
15.4. УПРАВЛЕНИЕ СЛУЖБАМИ 309
15.4.1. Графические менеджеры 309
15.4.2. Управление службами в init 313
15.4.3. Управление службами в systemd 314

ГЛАВА 16. МОНИТОРИНГ СИСТЕМЫ 317

16.1. ИНФОРМАЦИЯ О ДОСТУПНОМ МЕСТЕ НА ДИСКЕ 318
16.2. ИНФОРМАЦИЯ ОБ ИСПОЛЬЗОВАНИИ ОПЕРАТИВНОЙ ПАМЯТИ 319
16.3. ИНФОРМАЦИЯ О НАГРУЗКЕ СИСТЕМЫ 319
16.4. ЗАРЕГИСТРИРОВАННЫЕ ПОЛЬЗОВАТЕЛИ 321

16.5. МОНИТОРИНГ ИНТЕРНЕТ-КАНАЛА 322

16.5.1. Программа iftop 322

16.5.2. Программа nload 323

16.5.3. Программа iptraf-ng 324

16.6. ПРОГРАММА NETHOGS 326

16.7. МОНИТОРИНГ АППАРАТНЫХ ПРОБЛЕМ 327

16.8. ПРОВЕРКА ПРОИЗВОДИТЕЛЬНОСТИ ДИСКА 328

ГЛАВА 17. ЖУРНАЛИРОВАНИЕ 331

17.1. ВВЕДЕНИЕ В ЖУРНАЛИРОВАНИЕ 332

17.2. ВАЖНЫЕ ЖУРНАЛЫ 334

17.2.1. Журнал auth.log 334

17.2.2. Сообщения ядра - kern.log 336

17.2.3. Системный журнал syslog 338

17.3. РОТАЦИЯ ЖУРНАЛОВ 341

17.4. ПРОСМОТР ЖУРНАЛОВ С ПОМОЩЬЮ JOURNALCTL 342

17.4.1. Что такое journald.service 342

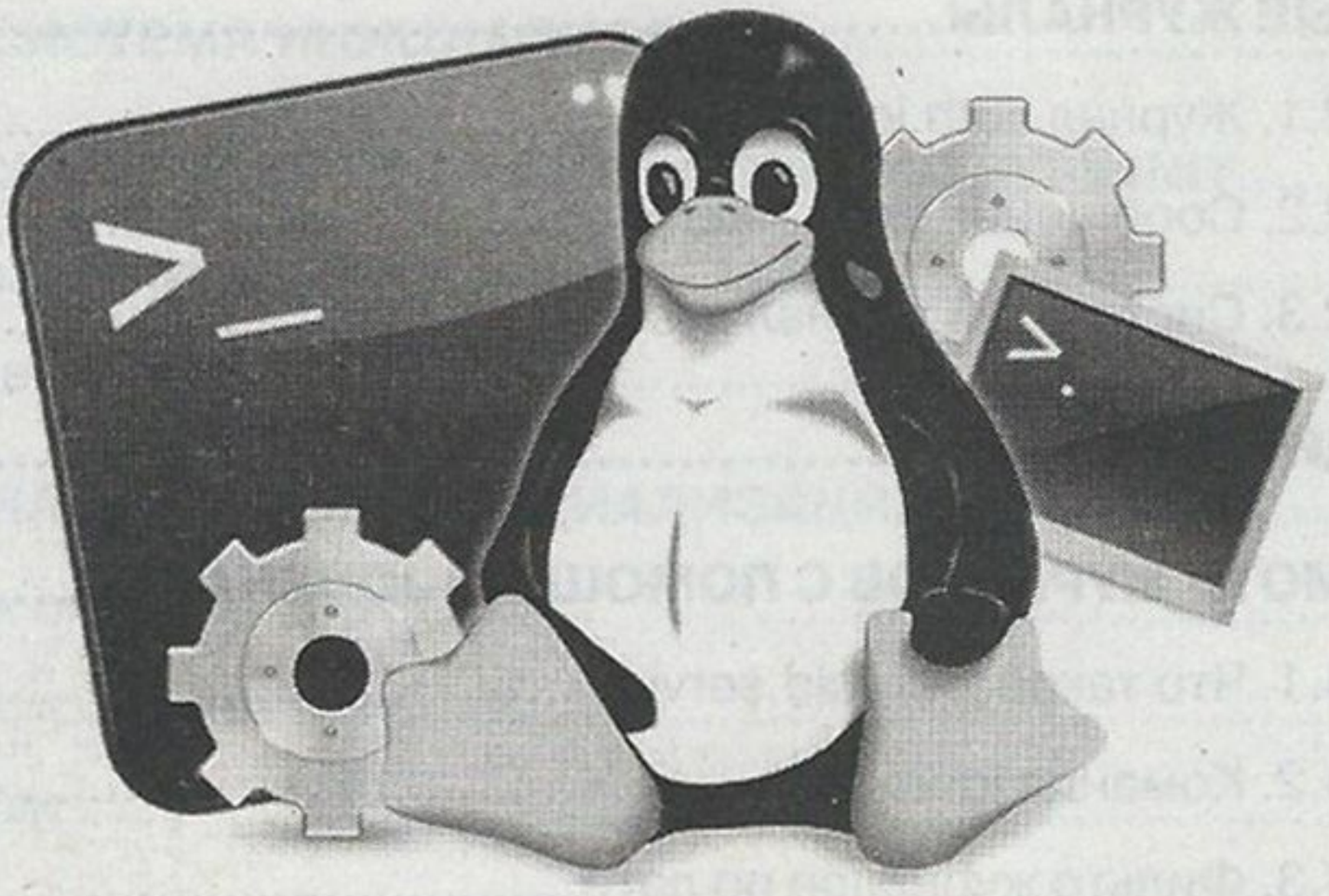
17.4.2. Команда journalctl 343

17.4.3. Фильтр журналов по дате 344

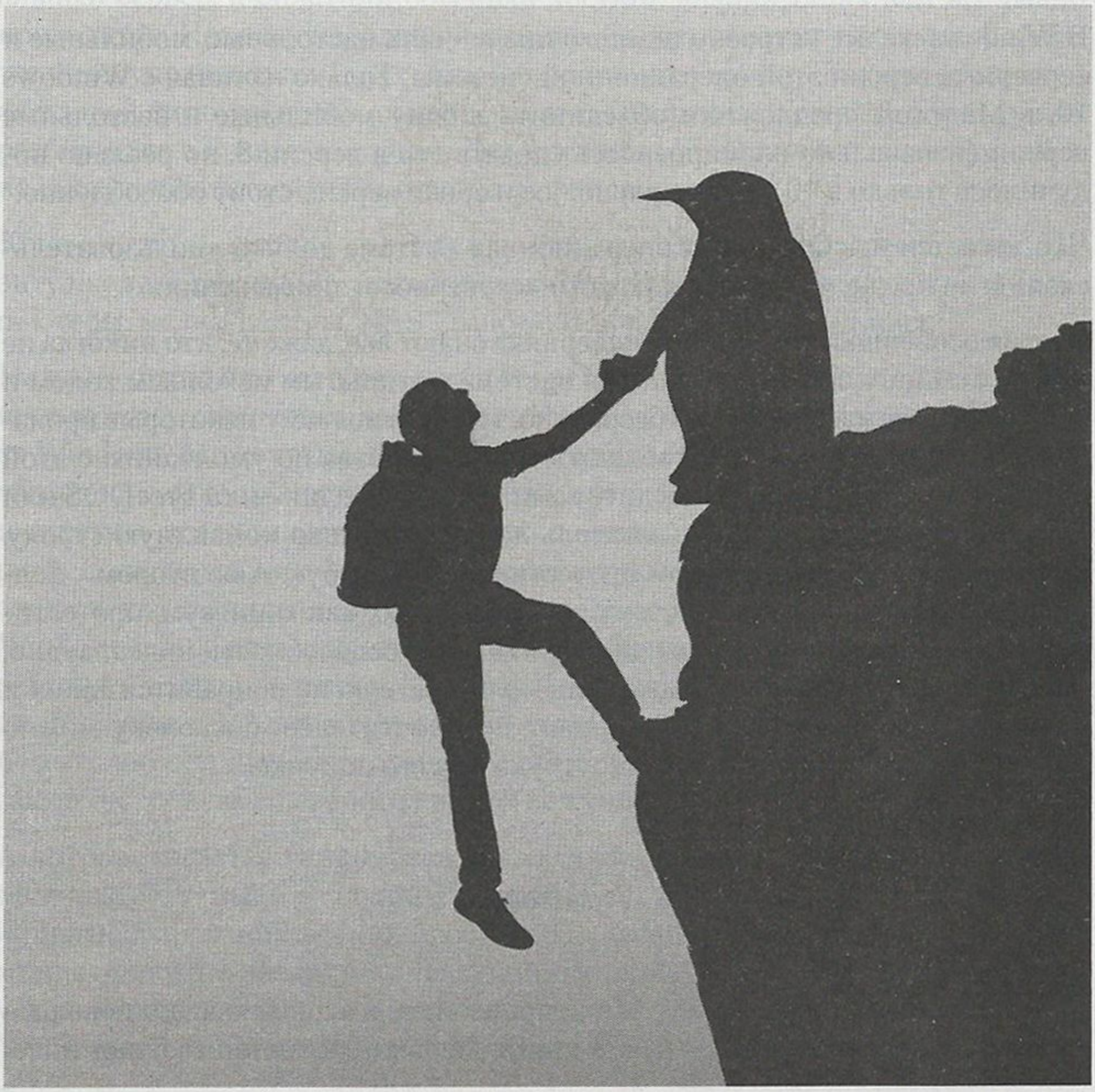
17.4.4. Фильтр по службе 344

17.4.5. Фильтр по уровню ошибки 345

ВМЕСТО ЗАКЛЮЧЕНИЯ 346



Введение



Что такое Linux?

Windows, macOS, Linux - три самые популярные операционные системы наших дней. Данная книга посвящена как раз Linux. Уникальность этой операционной системы заключается в ее универсальности. Вы можете установить ее и на десктоп, и на ноутбук, и на сервер. Многие встроенные устройства работают под управлением Linux.

В Windows же все устроено немного иначе - есть настольные, мобильные и серверные версии этой операционной системы. Только начиная с Windows 10, в Microsoft предложили объединить в одну мобильные и настольные версии (изначально планировалось сделать это в версии 8, но реально получилось только в 10), но все равно - серверная версия стоит обособленно.

Что касается macOS, то эта операционная система только «пользовательская» и не предусматривает никакого «серверного» применения.

Вторая особенность, о которой наверняка знают все, даже те, кто никогда не работал с Linux. Эта операционная система с открытым исходным кодом и распространяется абсолютно бесплатно. Именно поэтому некоторые производители ноутбуков предоставляют свои устройства по умолчанию с этой операционной системой. Посудите сами: выбор невелик - или FreeDOS или Linux. В первом случае пользователь получает только командную строку, не позволяющую даже толком протестировать ноутбук. А во втором - полноценную операционную систему, позволяющую, как минимум, протестировать ноутбук при покупке - проверить работоспособность видео, аудио, сетевого соединения и т.д. Даже если пользователю не понравится Linux и он установит Windows, такой вариант предпочтителен, поскольку можно убедиться в работоспособности ноутбука при его покупке.

Аудитория

Более подробное знакомство с Linux произойдет в первой главе, а пока разберемся, кому будет интересна эта книга. Первым делом книга будет инте-

ресна самым обычным пользователям, которые купили компьютер с предустановленной Linux и хотят разобраться в этой операционной системе.

Вторая категория пользователей - это системные администраторы, которым приходится по долго службы использовать эту операционную систему. Ведь предприятие может немало сэкономить, если все компьютеры (в том числе и сервер) будут работать под управлением этой операционной системы.

Третья категория - веб-разработчики. Большинство виртуальных серверов (VPS/VDS) работают под управлением Linux, так что книга будет полезна не только администраторам и пользователям именно Linux, но и веб-разработчикам, которым по роду своей работы приходится работать с виртуальными серверами, которые обычно работают под управлением этой операционной системы. Даже если вы не хотите арендовать целый виртуальный сервер, а оплатили обычный хостинг, знания Linux вам все равно пригодятся. Приведу небольшой пример. Сервер хостинг-провайдера работает под управлением Linux. Как минимум, вам нужно ориентироваться в файловой системе, чтобы понять, где и какие файлы находятся, вам нужно разбираться в правах доступа, иначе сценарии (программы) на вашем сайте не смогут обратиться к тем или иным файлам и т.д.

Конечно же, Linux (и, разумеется, эта книга) будет интересна всем IT-энтузиастам, поскольку большинство небольших «девайсов» - мини-компьютеры, медиа-центры, NAS - работают под управлением Linux.

Недавно читал статью о том, что в мире Linux не очень популярна: в статье приводилась статистика, что на настольных компьютерах чаще используется Windows. Может это и так (проверить приведенную статистику у меня, к сожалению, возможности нет), но если брать ситуацию на IT-рынке в целом и охватить все компьютеры - от мала до велика - то тут еще можно поспорить. Если сервер под управлением серверной версии Windows вполне реален, но где вы видели тот же NAS или мини-компьютер под управлением Windows? Большинство аппаратов самообслуживания, банкоматов также работают под управлением Linux. Выбор операционной системы не случаен. В случае с NAS и мини-компьютерами использовать Windows просто не выгодно - стоимость операционной системы может превысить стоимость самого устройства и его тогда никто не купит.

А вот что касается банкоматов, то здесь идет речь о стабильности операционной системы, о ее неприхотливости и о простоте администрирования и диагностирования всяких проблем. По своей работе мне приходится администрировать различного рода компьютерные системы - так вот найти и устранить проблему в Linux часто гораздо проще, чем в той же Windows. В Windows все просто, пока работает. Но если что-то сломалось, то на устране-

ние проблемы может быть затрачено больше времени, чем в случае с Linux. Взять бы те же конфигурационные файлы. Во-первых, все они текстовые и их можно редактировать в любом текстовом редакторе. Во-вторых, один конфигурационный файл отвечает за один сервис. Конечно, у сервиса может быть несколько конфигурационных файлов, но все они лежат в одном подкаталоге каталога /etc и вы знаете, где искать проблему. В Windows каталог /etc заменяет реестр, найти проблему в котором часто очень сложно, поскольку конфигурация различных компонентов системы разбросана по всему реестру, это уже не говоря о различных конфликтах настроек.

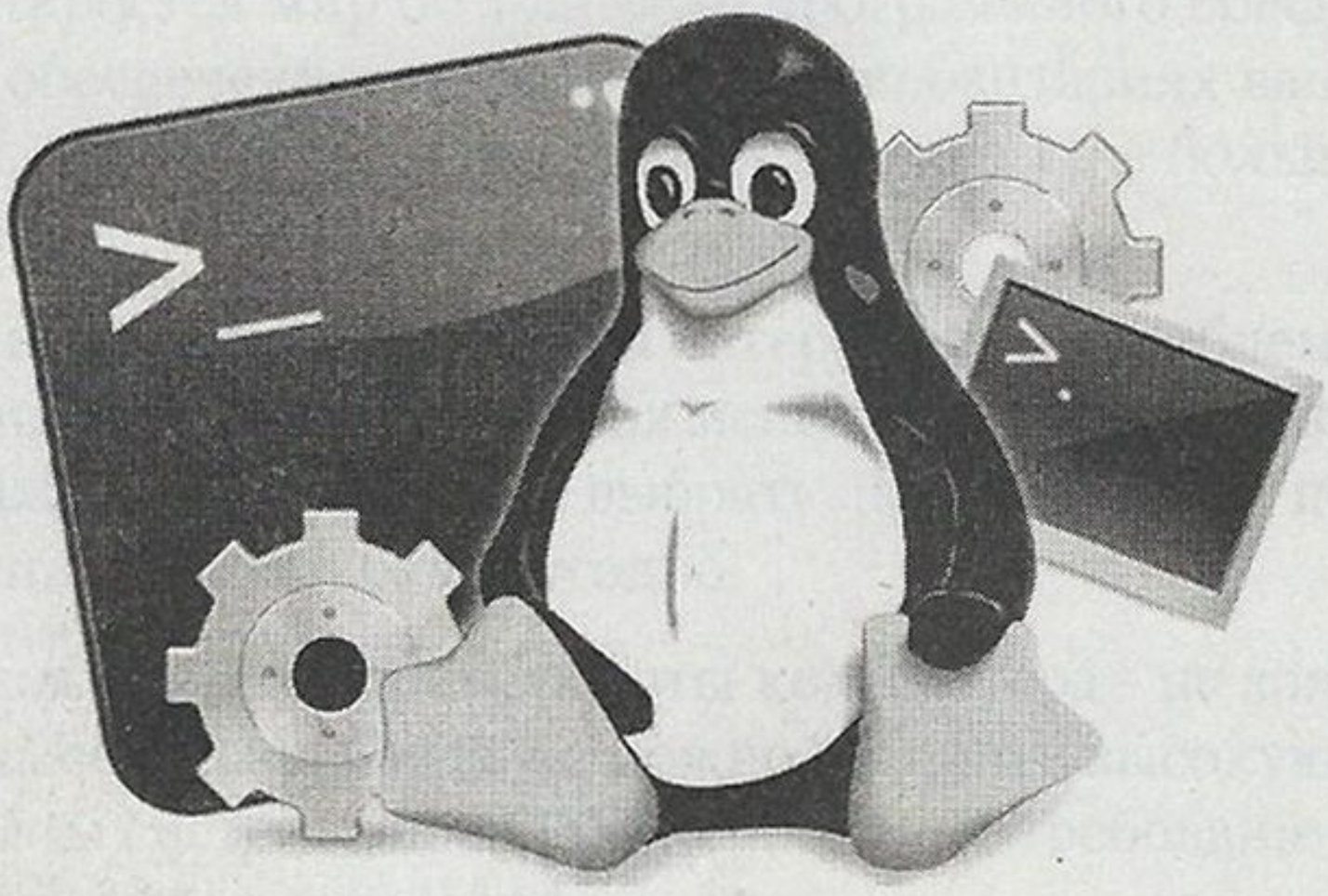
Установив Linux на свой компьютер, вам открывается действительно неограниченные возможности:

- Вы сможете запускать как Linux, так и Windows-программы (в эмуляторе или в виртуальной машине).
- Перед вами откроется мир бесплатного программного обеспечения. Все программное обеспечение, находящееся в репозиториях вашего дистрибутива (о том, что это такое - вы узнаете в главе 1), в большинстве случаев бесплатное.
- Кардинально изменить графический интерфейс - не проблема. Для Linux предусмотрено более 20 графических менеджеров (не тем оформления, а именно менеджеров) - вы можете выбрать любой из них и переключаться между ними, когда вам будет угодно.
- Получите высокую надежность работы компьютера - не зря Linux устанавливается на серверах. Сюда же можно добавить высокую производительность работы системы, поскольку системные требования Linux ниже и работает она быстрее, чем Windows.
- Высокая степень безопасности - это реальность. Всякие вирусы и прочие вредоносные программы вроде программ-шифровальщиков теперь для вас - не проблема. Я не говорю, что для Linux нет вирусов, но для нее вирусов гораздо меньше и даже те, что есть из-за устройства системы безопасности вряд ли смогут причинить системе существенный вред.
- Будете не как все. Сможете удивить своих друзей совершенно другой операционной системой.

Необходимые знания

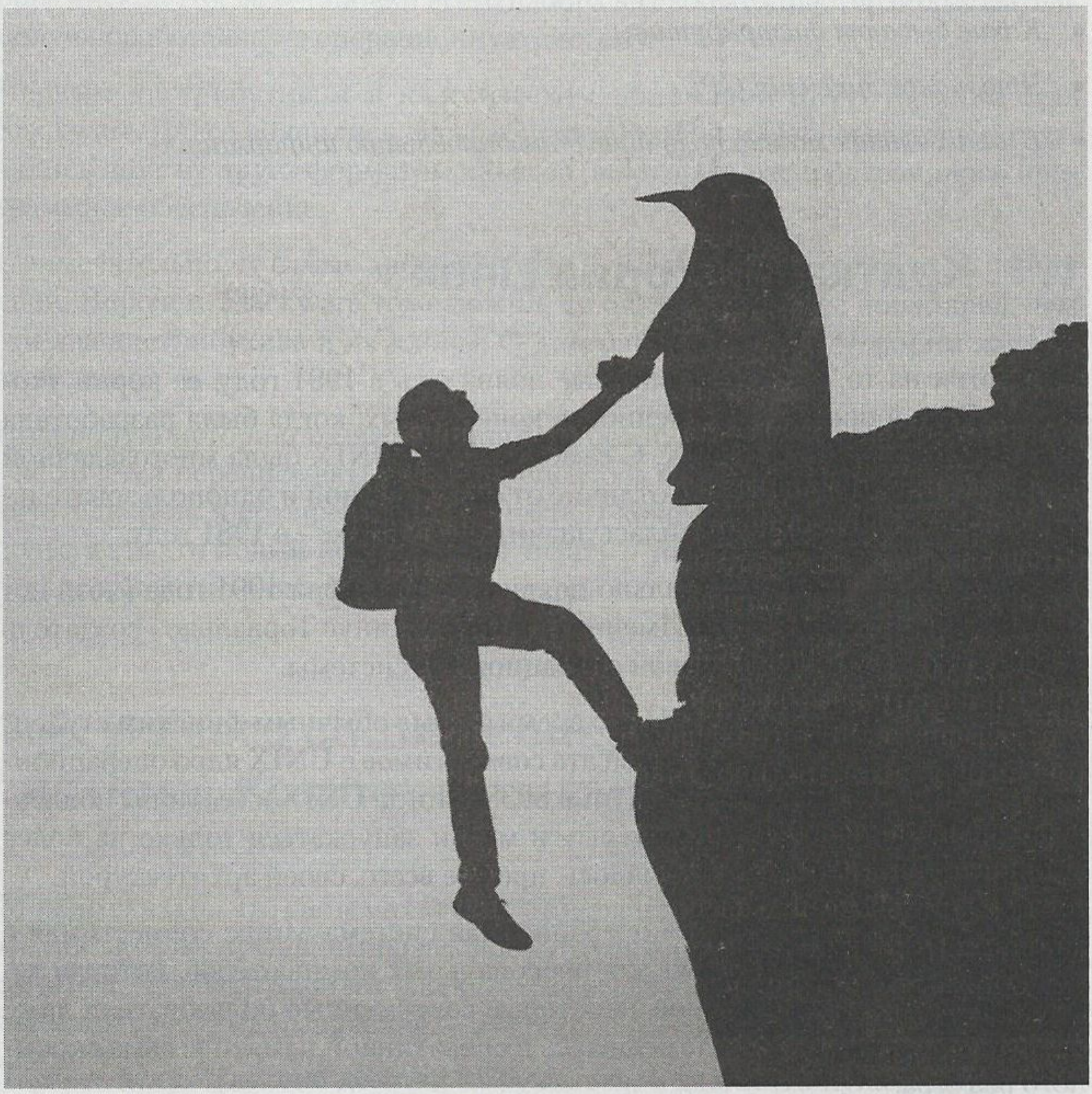
Данная книга не относится к самоучителю работы с компьютером - никто не будет объяснять, как работать с мышкой, объяснять назначения клавиш

на клавиатуре и т.д. Основ здесь не будет. Как минимум, чтобы успешно работать с этой книгой, вам нужно уметь пользоваться компьютером.



ГЛАВА 1.

Знакомство с Linux



В этой главе вы узнаете:

- *В каком году была создана Linux*
- *Из чего состоит Linux*
- *В каких каталогах находятся пользовательские и системные утилиты*
- *За что отвечает тот или иной компонент системы*
- *Какие бывают графические интерфейсы*
- *Какие бывают дистрибутивы*
- *Что такое лицензия GPL*
- *В каких главах можно получить дополнительную информацию*

1.1. Краткая история Linux

Не смотря на то, что Linux впервые появилась в 1991 году, ее корни уходят глубоко в прошлое, а именно - в конец 60-ых, когда была разработана операционная система UNIX. С самого начала UNIX была многозадачной и многопользовательской - в отличие от однозадачной и однопользовательской MS DOS, которая появилась значительно позже - в 1981 году.

Первая версия Linux официально появилась 5 октября 1991 года (хотя номер версии тогда был 0.02). Именно 5 октября Линус Торвальдс - создатель Linux - объявил о выходе новой операционной системы.

Кто такой Линус Торвальдс? На то время он был обычным финским студентом, который увлекся идеей написать совместимое с UNIX ядро операционной системы, но для процессора Intel 80386. Тогда UNIX и ее клоны поддерживали только «суперкомпьютеры» и могли запускаться только на более мощном железе, которое отличалось, прежде всего, своей архитектурой.

Прототипом ядра послужила операционная система Minix: совместимая с UNIX операционная система для персональных компьютеров, которая загружалась с дискет (наверное, некоторые современные пользователи даже не знают, что это такое) и помещалась в оперативной памяти весьма скромного размера.

В 1991 году речь шла только о создании ядра. Как уже отмечалось, тогда была выпущена версия ядра 0.02, а версия 0.99 (почти версия 1.0) появилась лишь в декабре 1993 года. Но одного только ядра для полноценной операционной системы недостаточно. Нужен еще набор утилит.

Благодаря проекту GNU Линус Торвальдс сразу получил возможность использовать в Linux множество свободных утилит - интерпретатор команд bash, компилятор hcc, архиваторы tar и gzip и многие другие. Все эти утилиты могли работать с его UNIX-совместимым ядром.

Позже энтузиасты начали выпускать так называемые дистрибутивы Linux: ядро и набор программ, при этом набор программ в разных дистрибутивах мог отличаться. Первые дистрибутивы Linux распространялись в виде набора дискет. После установки всего набора на свой компьютер пользователь получал работающую операционную систему.

Первым дистрибутивом в классическом понимании этого термина стал Slackware. Далее появились дистрибутивы RedHat и Debian, которые отличались друг от друга форматом пакетов, в которых распространялось программное обеспечение.

Примечательно, те самые дистрибутивы RedHat и Debian дожили до наших дней. Впрочем, Slackware тоже дожил, но он гораздо менее популярен, чем эти два дистрибутива и их клоны. О дистрибутивах мы поговорим далее в этой главе.

Подробную историю развития Linux вы без проблем найдете в Интернете - не хочется переписывать то, о чем писалось и говорилось много раз. Да и читателю (как я его прекрасно понимаю!) хочется быстрее узнать, как использовать его новый ноутбук, а не читать о том, в каком году Линус Торвальдс выпустил очередную версию ядра.

1.2. Из чего состоит операционная система. Компоненты системы

Можно сказать, что система состоит из ядра и программ (утилит). От части это так, но я же предпочитаю немного углубиться. Ведь программы бывают разными - драйвер дисплея и текстовый редактор - по сути, программы, но у каждой свое предназначение.

Первым делом система должна загружаться, а для этого нужен загрузчик. На данный момент в Linux используется загрузчик GRUB2 (GRand Unified

Bootloader). Именно он отвечает за загрузку ядра Linux в память и передает ему параметры, указанные пользователем. Загрузчик GRUB2 будет рассмотрен в главе 14, а ядро и его параметры - в главе 16.

Ядро не может содержать в себе код, обеспечивающий поддержку всех существующих в природу аппаратных средств - попросту говоря драйверов для «железа». Тогда бы ядро получилось огромным и вряд ли Linux могла запускаться на обычных персональных компьютерах (не говоря уже о встраиваемых устройствах). Именно поэтому драйверы, обеспечивающие поддержку того или иного оборудования, загружаются в виде модулей ядра.

Вот видите, мы уже знаем три компонента системы - загрузчик, ядро, модули ядра. Но и это еще далеко не все. Как и в Windows, в Linux есть службы - специальные программы, выполняющиеся в фоновом режиме и предоставляемые различные сервисы. Отсюда, кстати, и второе название этих программ - сервисы (services). Есть еще и более старое название, пришедшее еще со времен UNIX - демоны (daemons), именно поэтому названия многих таких программ заканчиваются на «d», например, httpd, mysqld и т.д. Примеры служб - веб-сервер, FTP-сервер, сервер имен, сервер баз данных и т.д.

Запуском служб управляет система инициации, задаваемая параметром ядра init. Во всех современных дистрибутивах используется система инициализации systemd, которая будет рассмотрена в главе 15. Но в более старых версиях может использоваться init или некоторые другие системы - init-ng (довольно редко), upstart (некоторые версии дистрибутива Ubuntu).

Наконец, настало время для программ. Для самых обычных программ. В отличие от Windows, программы устанавливаются не с помощью программы-инсталлятора setup.exe, а распространяются в виде пакетов. Пакет - это файл с «расширением» .rpm (RedHat, CentOS, Fedora, ALT Linux, openSUSE и некоторые другие) или .deb (Debian, Ubuntu и их клоны). Пакет можно представить себе, как архив, содержащий саму программу, дополнительные файлы (например, документацию), а также инструкции по установке этой программы.

Для управления пакетами (установка, удаление, обновление, вывод информации и т.д.) используется менеджер пакетов. В зависимости от дистрибутива менеджер пакетов может быть разный, например, в последних версиях Fedora используется dnf, в предыдущих версиях Fedora и CentOS - yum, в Debian и Ubuntu используется apt, в openSUSE - zypper. Подробно об установке программ мы поговорим в главе 5.

После того, как программа установлена, ее можно запустить командой, которая обычно соответствует полному или сокращенному имени программы,

например, для запуска файлового менеджера Midnight Commander используется команда mc.

Программы, относящиеся к базовой системы, базовые утилиты, такие как bash (командный интерпретатор), date (вывод или установка даты), cat (вывод файла), cp (копирование файла) находятся в каталоге /bin. Запустить программы из этого каталога, пусть с некоторыми ограничениями, может любой пользователь. Системные утилиты, например, e2fsck (проверка файловой системы), cfdisk (конфигурирование диска) находятся в каталоге /sbin. Запустить их может только пользователь root или пользователь с привилегиями sudo (подробнее о правах доступа мы поговорим в главах 11 и 12).

Программы, не относящиеся к базовой системе - как правило, это программы, устанавливаемые пользователем с помощью менеджера пакетов, устанавливаются в каталоги /usr/bin (для всех пользователей) и /usr/sbin (для привилегированных пользователей).

Как видите, система состоит из множества компонентов, а не только их ядра и программ. Хотя, если рассматривать все упрощенно, то все сводится к ядру и программам - ведь система инициализации, менеджер пакетов, загрузчик - все это программы, а без модулей ядра можно жить, если перекомпилировать ядро и включить весь необходимый код в ядро - можно отказаться от модулей.

1.3. Дистрибутивы

Как уже отмечалось ранее, существуют три самые первые дистрибутивы, дожившие до наших дней - Slackware, RedHat и Debian. Примечательно, что все эти дистрибутивы появились в 1993 году. Конечно, было много разных дистрибутивов и довольно таки успешных, которые до наших дней не дожили. Из довольно древних и удачных вспоминается BlackCat - он основывался на RedHat, но обладал отличной русификацией и был очень популярен на территории бывшего СССР. Также был довольно неплохой французский дистрибутив Mandrake (Mandriva), который прекратил свое существование в 2012-ом году.

Мы сейчас не будем останавливаться на различных устаревших и экзотических дистрибутивах, а рассмотрим современные дистрибутивы, основанные на RedHat и Debian.

1.3.1. Debian

Начнем с Debian. Не потому что это лучший дистрибутив (все современные дистрибутивы очень даже неплохие), а просто потому, что мне так захотелось. Дистрибутив появился 15 августа 1993 года. Название дистрибутива происходит от имен создателя проекта Яна Мердока (Ian Murdock) и его подруги Дебры (Debra).

Дистрибутив отличается своей надежностью и стабильностью. В его состав не включают нестабильные версии программ, поэтому есть два побочных эффекта. Первый - дистрибутив очень надежен, второй - вряд ли вы найдете в нем самые последние версии программ.

Данный дистрибутив подойдет, как для сервера, так и для домашнего компьютера, если его владелец хочет просто работать и не любит всякие экспериментальные штуки.

1.3.2. Fedora

Раньше дистрибутив RedHat устанавливался, как на серверы, так и на рабочие станции. Позже от него откололась пользовательская ветка - Fedora Core. Официальной датой рождения этого дистрибутива считается 22 сентября 2003 года, когда компания Red Hat объявила о разделении дистрибутива на два субпроекта — развиваемый при участии сообщества Fedora Linux; и коммерческий Red Hat Enterprise Linux. Коммерческая версия, как понятно из ее названия, устанавливается на корпоративные серверы и рабочие станции. Этого дистрибутива мы касаться не будем, так как есть маленькая вероятность, что он вам встретиться. Все переходят на Linux, чтобы не платить, поэтому вероятность того, что вам придется работать с платным (коммерческим) дистрибутивом очень низка.

Проект Fedora Core позже переименовали просто в Fedora. Он поддерживается исключительно добровольцами и изначально возник в Гавайском университете. Дистрибутив содержит только свободное программное обеспечение.

Сейчас есть две основных ветки этого дистрибутива - Рабочая станция (Workstation) и Сервер (Server). Из названия веток, думаю, понятно, что первая оптимизирована под использование на рабочей станции или домаш-

нем компьютере, а вторая - на сервере (по умолчанию, даже не предлагается к установке графический интерфейса - на сервере он будет лишним).

1.3.3. CentOS

CentOS (Community ENTerprise Operating System) — дистрибутив Linux, основанный на коммерческом Red Hat Enterprise Linux компании Red Hat и совместимый с ним. Другими словами, это бесплатная версия Red Hat Enterprise Linux. Срок поддержки каждой версии CentOS составляет 10 лет. Каждая версия обновляется каждые 6 месяцев для поддержки новых аппаратных средств.

Данный дистрибутив очень популярен (в силу своей специфики) на серверах, хотя никто и ничто не может помешать установить его на рабочую станцию. Лично мне очень нравится этот дистрибутив, хотя я использовал его в основном для построения сервера виртуализации - в связке с OpenVZ - для запуска на реальном физическом сервере нескольких виртуальных серверов.

1.3.4. Ubuntu

Наверное, самый молодой и самый популярный дистрибутив Linux. Первая версия этого дистрибутива появилась 20 октября 2004 года и основана на Debian. Фактически, это форк (ответвление) дистрибутива Debian, который перерос в полноценный самостоятельный дистрибутив.

Поскольку дистрибутив создавался в 2004 году он еще не успел «пропитаться» всем архаизмом древних дистрибутивов и все древние решения попросту в него не вошли. Зато разработчики постарались сделать его удобным, чем Debian. Компактность дистрибутива охраняется принципом «одна задача - одна программа». По умолчанию в Ubuntu устанавливается только одна программа (наиболее успешная по мнению разработчиков) для решения одной задачи. То есть у вас не будет сразу после установки двух почтовых клиентов, трех браузеров и т.д. Будут только те программы, которые вы будете использовать. Конечно, вы можете доустановить любые программы из репозитария, но сначала дистрибутив не будет пожирать ваш диск программами, которые вам не нужны.

По этой же причине произошло разделение Ubuntu на две ветки - Ubuntu и Kubuntu. Второй дистрибутив вместо графической среды GNOME использовал KDE. А ведь другие дистрибутивы или предлагали выбрать сразу при установке или (что хуже), устанавливали обе среды сразу с возможностью переключения между ними.

Говорить об этом дистрибутиве можно очень долго. Его визитная карточка - это графический интерфейс Unity, который был разработан специально для Ubuntu. Первые версии Ubuntu поставлялись с GNOME, но, начиная с версии 11.04 (которая вышла в 2011-ом году), оболочка GNOME была заменена на Unity.

Вторая особенность этого дистрибутива заключается в том, что учетная запись root по умолчанию отключена из соображений безопасности, а выполнение административных задач производится через команду sudo (гл. 11).

В отличие от Debian, в состав Ubuntu традиционно включаются нестабильные версии программ. Да, они самые новые, содержат самые новые возможности, но могут работать нестабильно. Именно по этой причине не рекомендуется использование Ubuntu на серверах, хотя серверная ветка этого дистрибутива все же существует - Ubuntu Server и я знаю примеры довольно успешной работы Ubuntu на реальных серверах.

1.4. Графические интерфейсы

В Linux очень много различных графических оболочек. Установив ту или иную оболочку, вы можете кардинально поменять внешний вид пользовательского интерфейса. В этом разделе мы рассмотрим некоторые (наиболее популярные) оболочки, а также разберемся, как их можно установить в Ubuntu. Почему именно в Ubuntu? Учитывая популярность этого дистрибутива и его ориентированность на пользователя, есть огромная вероятность, что вашим первым дистрибутивом станет именно Ubuntu. На рис. 1.1 изображена оболочка Ubuntu по умолчанию - Unity.

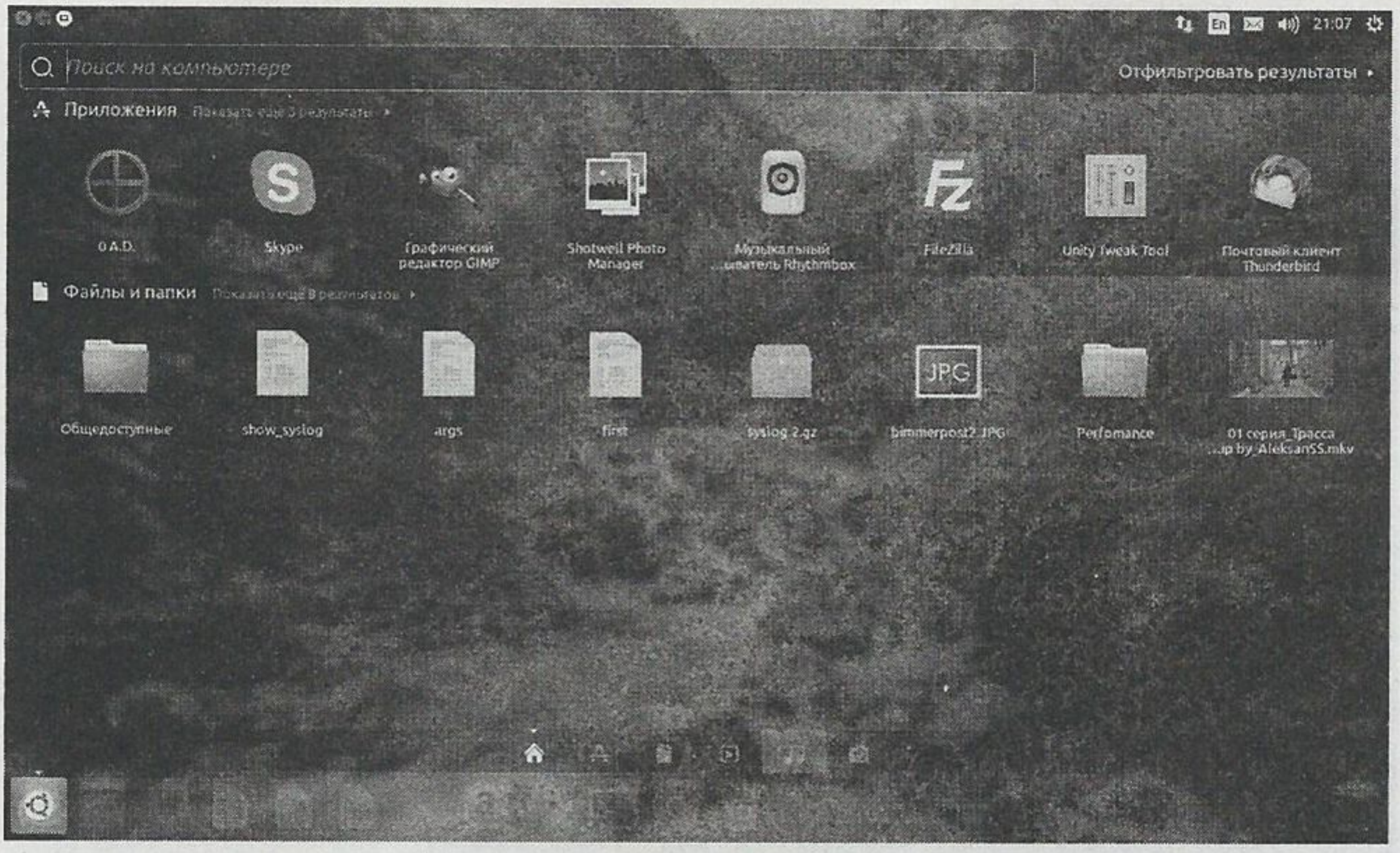


Рис. 1.1. Unity

1.4.1. Графическая среда GNOME

Очень популярная оболочка, используемая в различных дистрибутивах Linux по умолчанию. Если вы привыкли к GNOME (GNU Network Object Model Environment) или просто вам не нравится Unity (мне, например, она не очень нравится), то можно установить GNOME. Для этого нужно ввести в терминале следующую команду:

```
sudo apt-get install gnome-shell
```

> **Примечание.** Понимаю, что пока вы не понимаете, куда нужно вводить эту команду. Не беспокойтесь! В главе 3 будет рассмотрена командная строка, а в главе 5 мы поговорим об установке программ.

Преимущество GNOME - в том, что это стандарт де-факто и если вы привыкнете к нему, вы будете чувствовать себя как рыба в воде в любом дистрибутиве Linux. Если сравнивать с Unity, то последняя удобнее, чем вторая версия GNOME, а вот третья версия (GNOME 3) будет, на мой взгляд, удобнее, чем Unity.

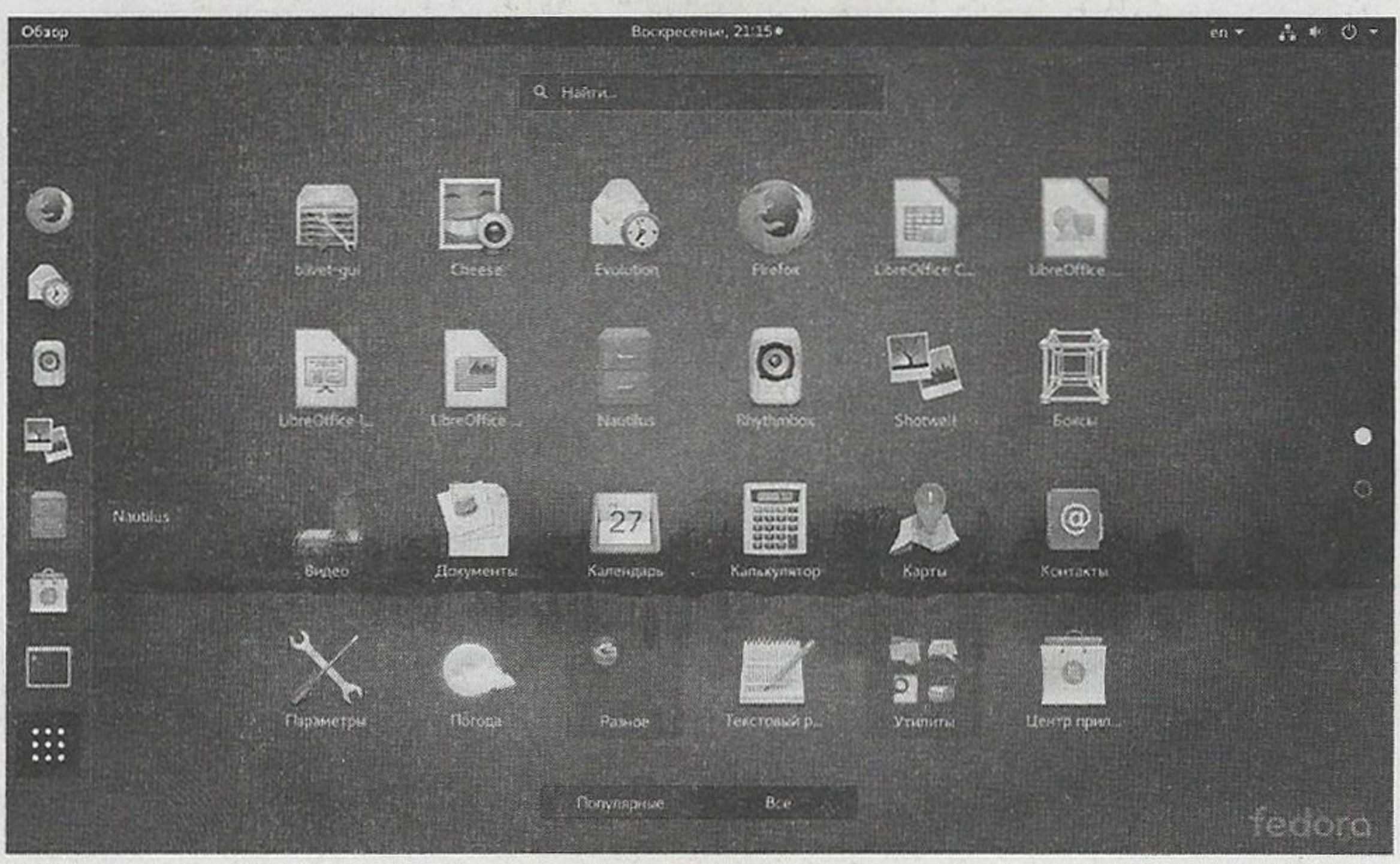


Рис. 1.2. GNOME (Fedora 26)

1.4.2. Графическая среда KDE

KDE (K Desktop Environment) - главный конкурент GNOME. Нужно отметить, что KDE появилась раньше, чем GNOME - в 1998 году, а GNOME - в 1999-ом. Фанаты GNOME и KDE могут спорить целыми днями напролет о том, какая графическая оболочка лучше. Чтобы решить, что все же лучше - KDE или GNOME, нужно попробовать поработать в обеих оболочках (рис. 1.3).

Для установки KDE в Ubuntu введите команду:

```
sudo apt-get install --no-install-recommends kubuntu-desktop
```

Я, думаю, вы будете довольны, как KDE, так и GNOME - просто решите, что удобнее непосредственно для вас.

Оболочки KDE, GNOME и Unity нельзя назвать «легкими». Это эдакие монстрообразные графические подсистемы (да именно так). А что, если у вас слабый компьютер и вы хотите более «легкий» графический менеджер? Чтобы работал быстрее, да и места на диске меньше занимал?

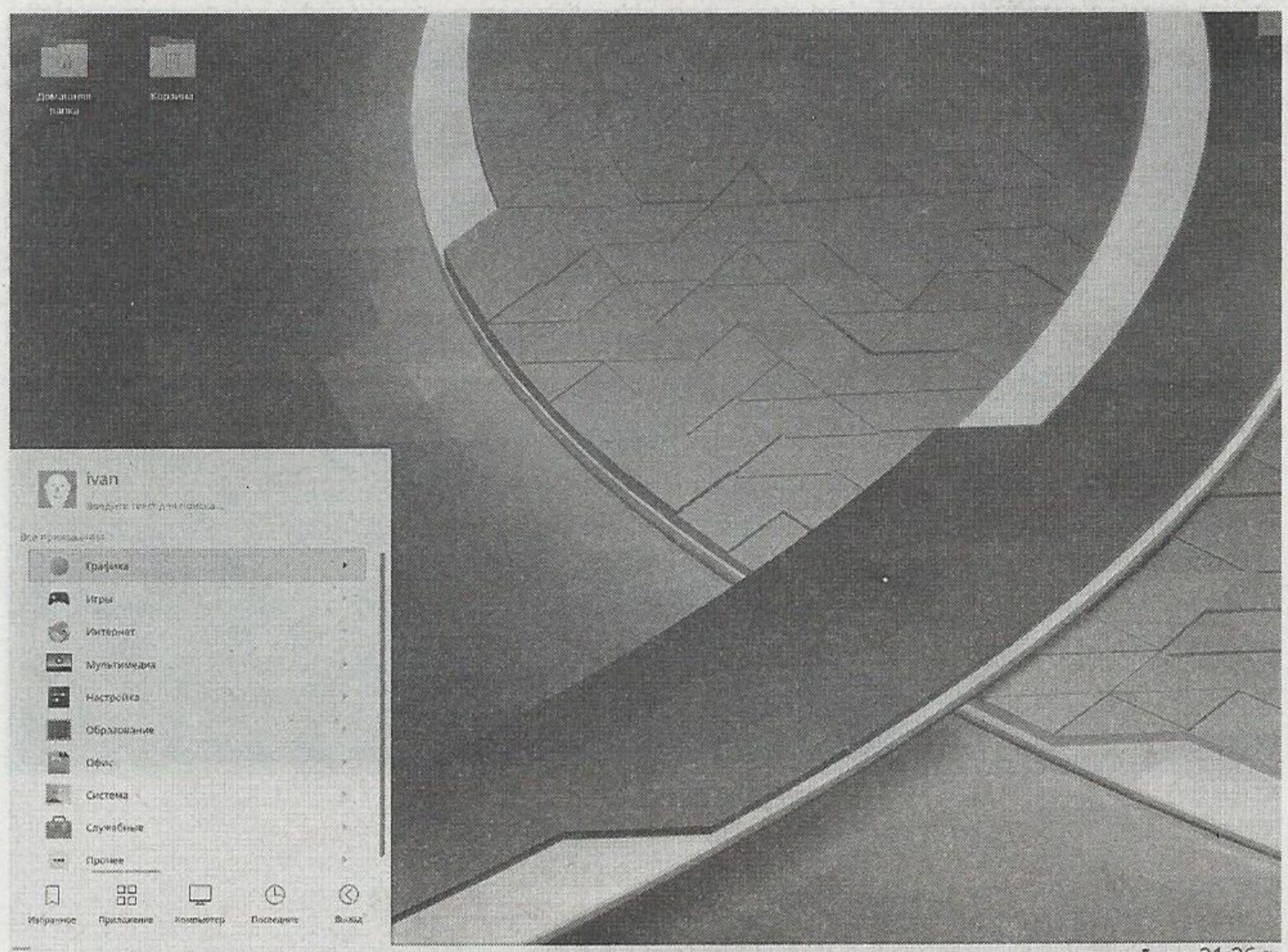


Рис. 1.3. KDE (Plasma) в Ubuntu 17.04

В этом случае нужно обратить внимание на следующие графические менеджеры.

1.4.3. Оболочка LXDE

Быстрая и полноценная среда рабочего стола, учитывая, что она потребляет очень мало системных ресурсов, идеальна для установки на слабых компьютерах. В качестве оконного менеджера используется Openbox, что обеспечивает довольно симпатичный вид окнам. Да и сама среда выглядит довольно таки ничего, а если представить, что еще и системных ресурсов практически не потребляет, прямо таки идеал для нетребовательных пользователей.

Для установки LXDE нужно ввести команду:

```
sudo apt-get install lxde
```

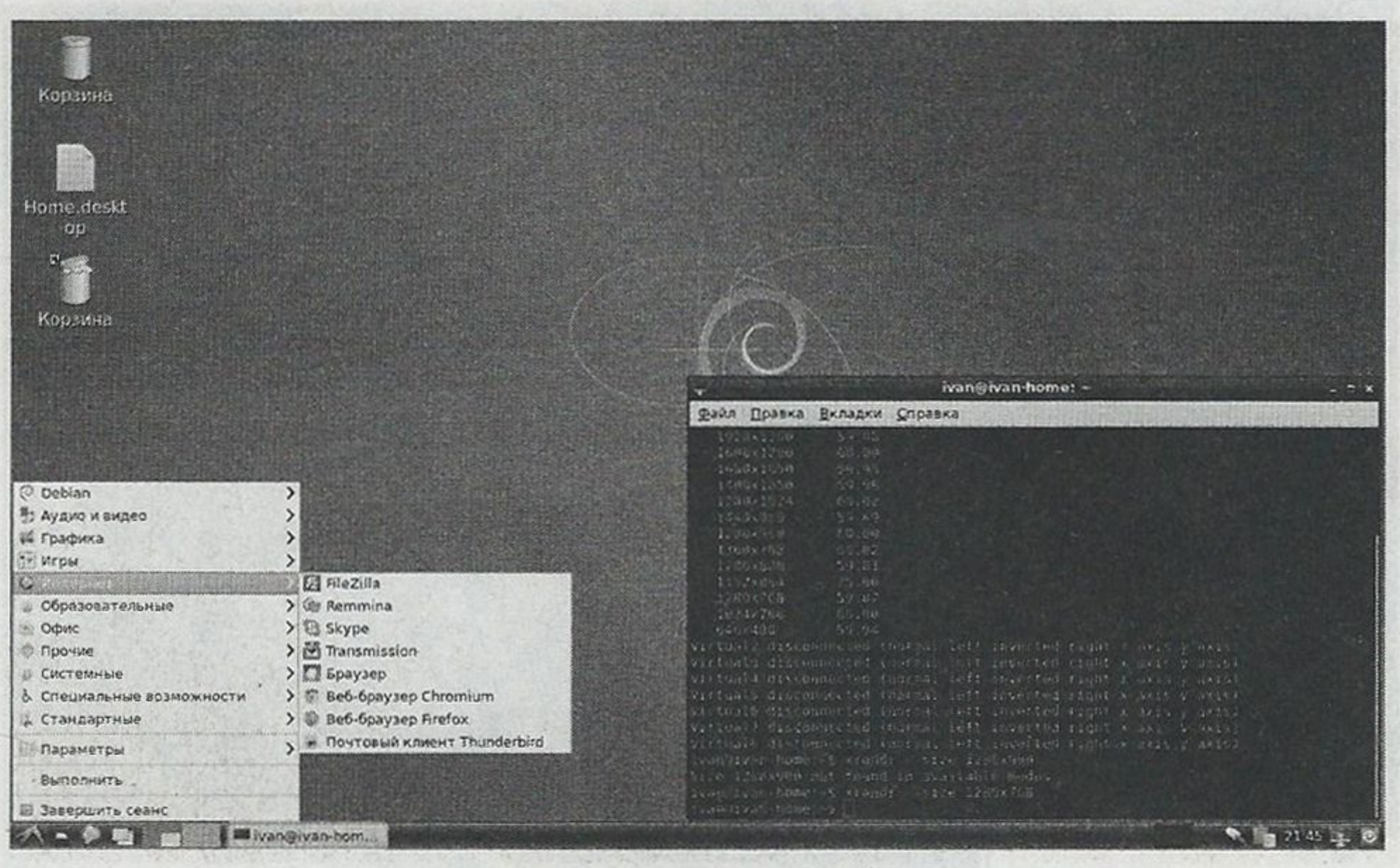


Рис. 1.4. Среда LXDE

1.4.4. Оболочка Xfce

Еще одна простая и довольно экономная среда (хотя все же ресурсов она потребляет бо́льше, чем LXDE) - Xfce, Основана на базе GTK+2 (эта же библиотека лежит в основе GNOME 2) и использует собственный менеджер окон Xfwm. Она используется по умолчанию в дистрибутиве Xubuntu - один из форков Ubuntu.

Установить можно командой:

```
sudo apt-get install xfce4
```

Все описанные до этого оболочки входят в состав репозитария Ubuntu (впрочем, как и многих других дистрибутивов), поэтому их установка сводится к вводу одной команды. А вот дальше мы рассмотрим более экзотические менеджеры, для установки которых придется немного «поколдовать».

Рис. 1.5. Оболочка Xfce

1.4.5. Оболочка Cinnamon

Cinnamon - оболочка рабочего стола, ответвившаяся от GNOME Shell. Главная особенность Cinnamon заключается в том, что она использует различные классические компоненты GNOME (панель задач, главное меню), при этом стараясь их полностью интегрировать в рабочую среду, для улучшения производительности работы. Другими словами, она похожа на GNOME, но работает быстрее. Данная оболочка используется по умолчанию в дистрибутиве Mint, который представляет собой ответвление от Ubuntu.

Для установки Cinnamon введите команды:

```
sudo add-apt-repository ppa:gwendal-lebihan-dev/cinnamon-
stable
sudo apt-get update
sudo apt-get install cinnamon
```

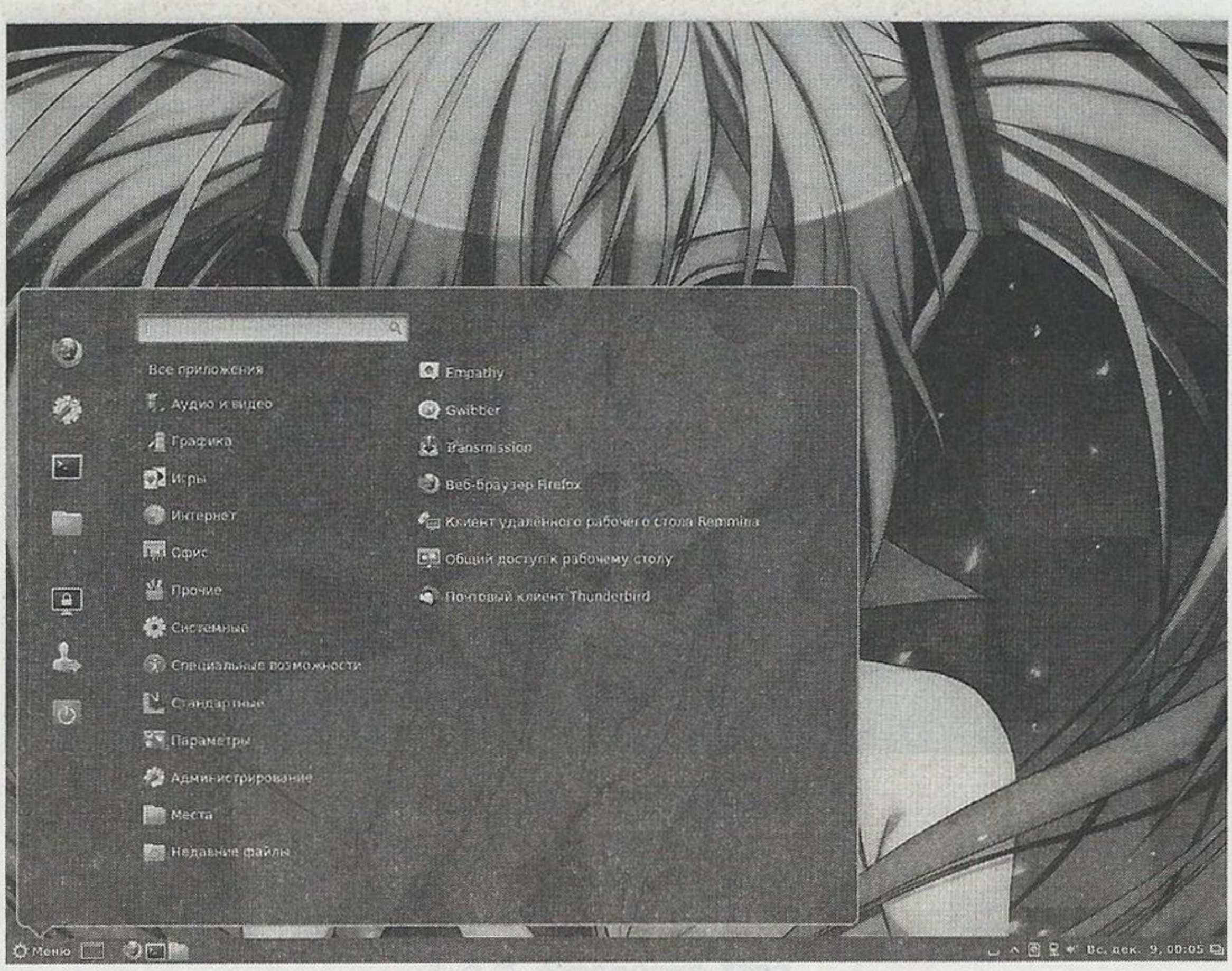


Рис. 1.6. Оболочка Cinnamon

1.4.6. Оболочка Mate

Как уже отмечалось ранее, современной версий GNOME на данный момент является GNOME 3.x. До этого же, вполне логично, была версия 2.x. Многим пользователям все еще больше нравится версия 2.x, но она уже не поддерживается. Что же делать? Выход есть - установить оболочку Mate. Ее главная особенность заключается в том, что она нормально устанавливается на систему с GNOME 3.x и при этом умудряется ничего не поломать:

```
sudo apt-get install mate-desktop-environment
```

При этом Mate выглядит так же, как и GNOME 2 - ностальгируйте на здоровье:

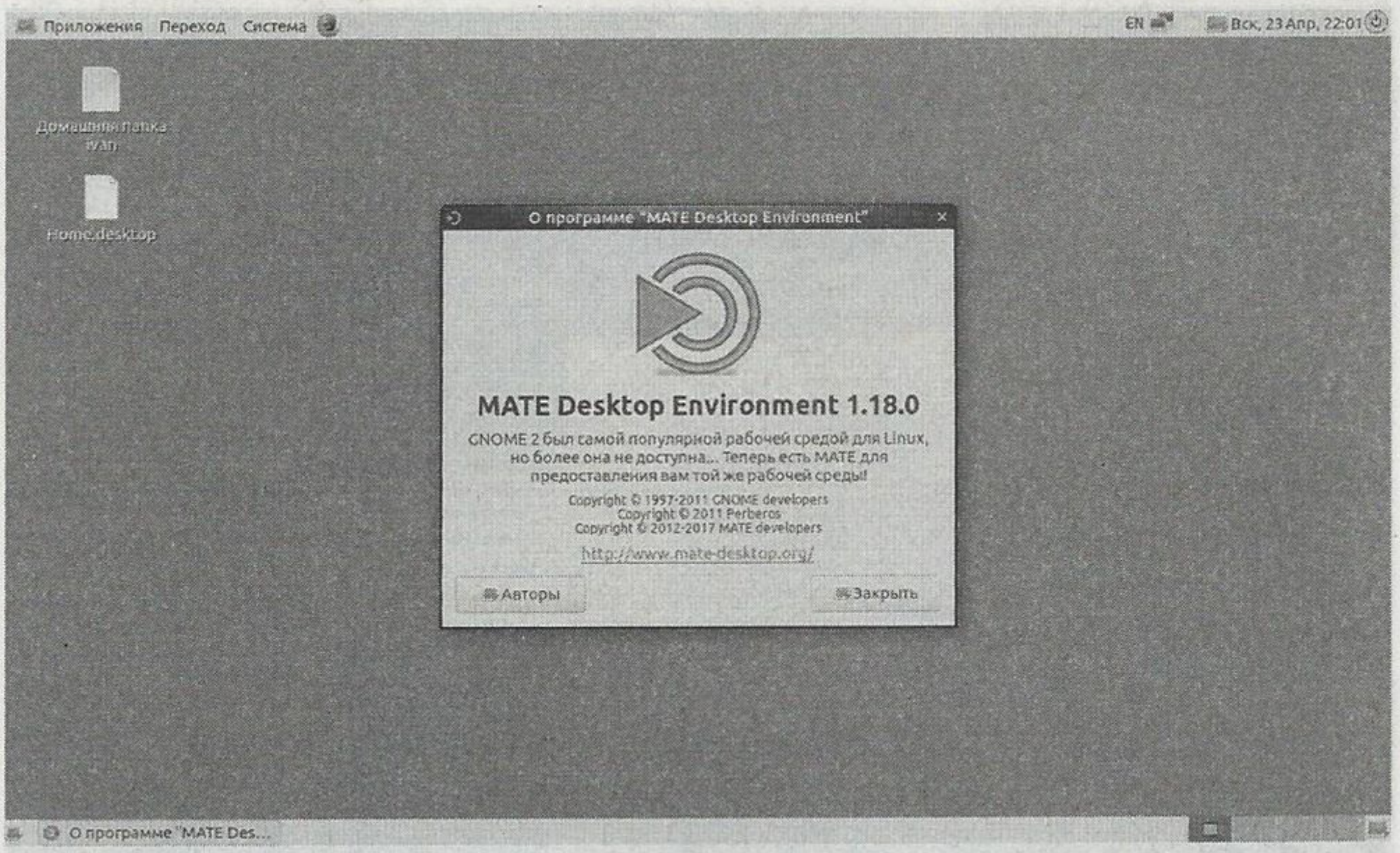


Рис. 1.7. Оболочка Mate

1.4.7. E17 (Enlightenment)

E17 - полноценная среда рабочего стола. По умолчанию выглядит внешне не очень, но все зависит от темы оформления. Небольшая размер и высокая производительность - вот характерные черты этой оболочки. Для ее установки нужно ввести следующие команды:

```
sudo apt-add-repository ppa:hannes-janetzek/enlightenment-
svn
sudo apt-get update
sudo apt-get install e17
```

Внешний вид E17 - на любителя, интерфейс кажется довольно «древним». Конечно, не таким древним, как WindowMaker (пакет wmaker).

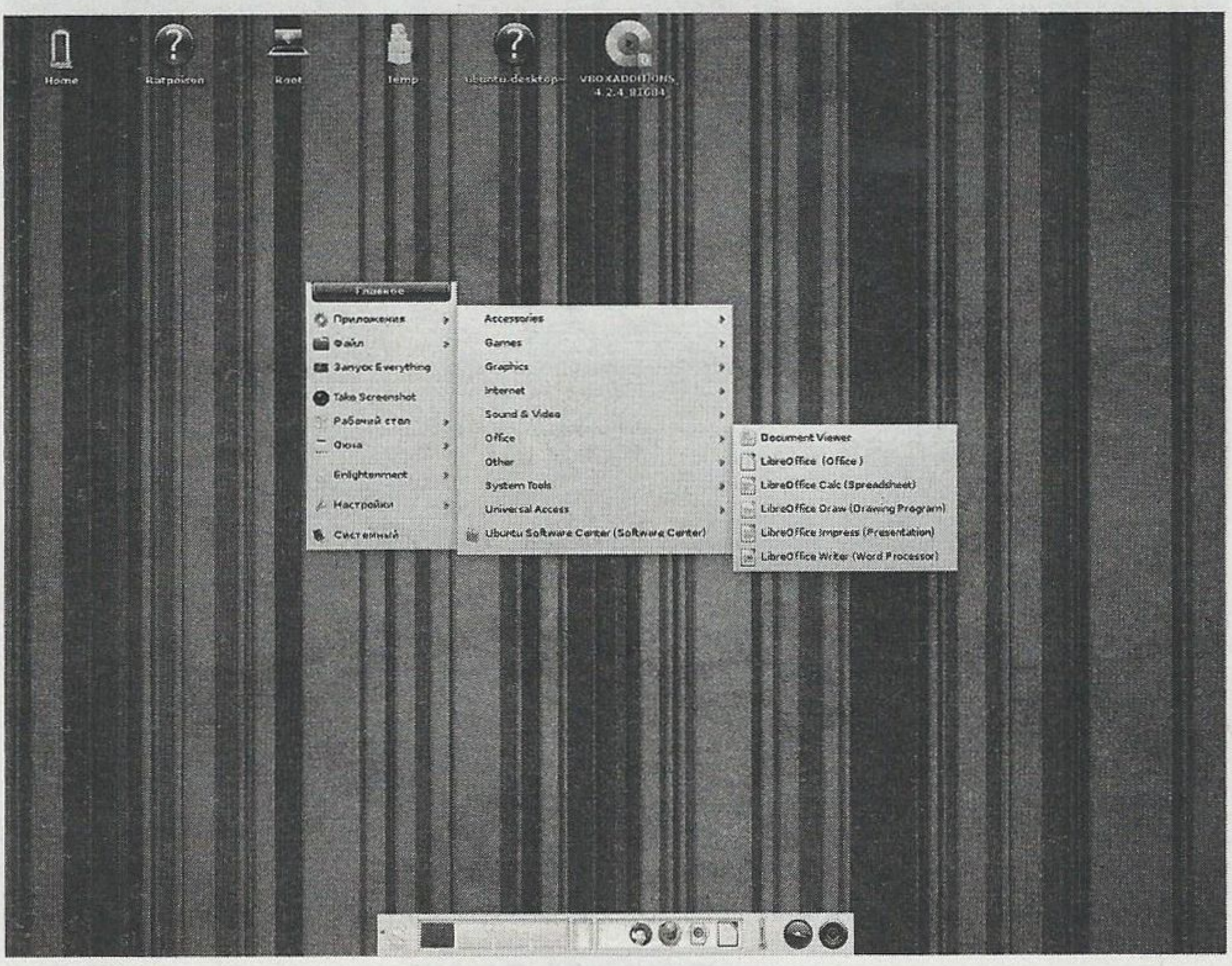


Рис. 1.8. Оболочка E17

1.4.8. Оболочка Razor-qt

Оболочка Razor-qt построена на базе той же библиотеки, что и KDE, - Qt. Именно поэтому оболочка может рассматриваться, как более легкая версия KDE и, благодаря использованию Qt, может использовать темы и другие компоненты такие же, как у KDE. А теперь представьте, что вам нужна какая-то программа, которая использует Qt. По идее, тогда бы вам пришлось установить KDE, но поскольку у вас слабый компьютер, все это будет тормозить. Выход из этого положения - оболочка Razor-qt, которая более «легкая», чем монстрообразный KDE.

Для установки оболочки введите команды:

```
sudo add-apt-repository ppa:razor-qt
sudo apt-get update
sudo apt-get install razorqt
```

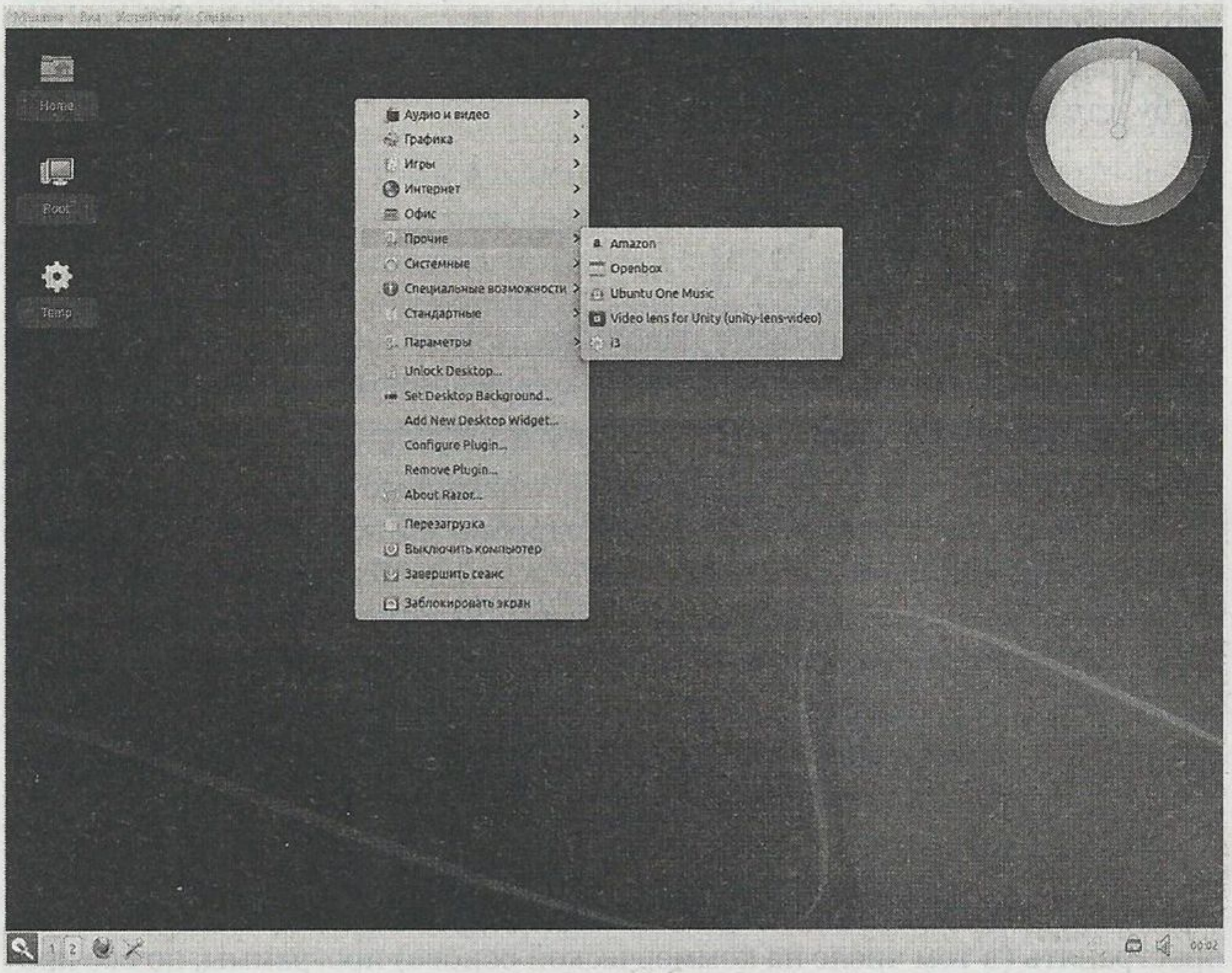


Рис. 1.9. Оболочка Razor-qt

Примечание. Данные команды нужно вводить в приложении Терминал. Подробнее о командной строке мы поговорим в главе 3.

На этом обзор оболочек заканчивается. На самом деле оболочек гораздо больше, чем было описано в этой главе. Но остальные оболочки вам вряд ли захочется устанавливать. Есть оболочки с довольно старым интерфейсом, например, уже упомянутый WindowMaker - вы будете чувствовать, что оказались в 1997 году... Да, эдакая машина времени. Есть оболочки с откровенно спартанским интерфейсом - открывается терминал для ввода команд и есть контекстное меню для запуска графических программ (и то, могу поспорить, что в нем будут далеко не все программы - зачастую меню нужно редактировать вручную). Пример таких оболочек - Fluxbox, Openbox, Blackbox, jwm, IceWM, fvwm. Есть оболочки с довольно странным интерфейсом, например, Ratpoison (http://www.tuxradar.com/content/how-turbo-charge-your-linux-desktop) - оболочка, не поддерживающая мышку, управление в ней осуществляется только с помощью клавиатуры. Еще при-

мер оболочки со странным интерфейсом - i3. Вы, конечно, можете попробовать в свободное время установить эти экзотические оболочки, но вряд ли вы будете в них работать.

1.5. Лицензия GPL

Наше знакомство с Linux будет неполным, если мы не поговорим о лицензии GPL, поскольку сама Linux, а также большинство программ для этой операционной системе распространяется именно по этой лицензии.

Стоит отметить, что в мире свободного программного обеспечения GPL - не единственная, но основная лицензия. Подробно мы не будем вникать в особенности той или иной лицензии - чтобы успешно использовать Linux, вам это знать не нужно. Но если возникнет желание или необходимость, вы без проблем найдете информацию о любой из лицензий в Интернете.

Итак, вернемся к GPL (GNU General Public License, Универсальная общественная лицензия GNU). Это лицензия на свободное программного обеспечение, созданная в 1988 году в рамках проекта GNU. Согласно GPL, автор передает программное обеспечение в общественную собственность.

Пользователю предоставляется право копировать, модифицировать и распространять (в том числе на коммерческой основе) программы, но все производные программы должны также распространяться по лицензии GPL. Принцип «наследования» прав называется «копилефт» (транслитерация с англ. copyleft) и был придуман Ричардом Столлманом. По контрасту с GPL, лицензии проприетарного (коммерческого) ПО практически никогда не предоставляют пользователю такие права и обычно, наоборот, стремятся их ограничить, например, запрещая восстановление исходного кода.

Если обобщить длинный текст лицензии, то GPL предоставляет пользователям следующие права:

- Право запуска программы с любой целью.
- Право изучать устройство программы.
- Право модифицировать исходный программы (лицензия GPL предусматривает свободный доступ к исходному коду программы).
- Свободу распространений исходного, так и исполнимого кода программы.
- Право улучшения программы.

Требование только одно - производная программа также должна распространяться по лицензии GPL. Это означает следующее: если вам нравится программа А, но вы ее хотите модифицировать и выпустить программу Б, то эта программа также должна распространяться по лицензии GPL и ее исходный код должен быть доступен. При этом лицензия GPL не запрещает продавать программу Б, но исходники ее должны быть при этом открыты. Однако в большинстве случаев программы, распространяемые по лицензии GPL, являются бесплатными. Конечно, есть и некоторые исключения. Например, некоторые дистрибутивы являются коммерческими (платными), но, если разобраться, то на самом деле дистрибутив оказывается бесплатным, а пользователь платит лишь за поддержку, в том числе за подписку на обновления. Пример такого дистрибутива - Red Hat Enterprise Linux. Основная его особенность - наличие коммерческой поддержки на протяжении 10 лет с возможностью продления до 13 лет. Стоит RHEL - не дешево. Так, один год стандартной подписки на Red Hat Enterprise Linux Server в 799 долларов, а 1 год премиум подписки - 1299 долларов. С одной стороны, это дорого. С другой стороны стоимость годовой подписки равна стоимости месячной зарплаты администратора такого сервера, так что на фоне всех остальных затрат - не так уж и много.

1.6. Роли операционной системы

1.6.1. Что такое сервер? Linux на сервере сети

Считается, что Linux лучше всего подходит для серверов. Поэтому начнем с серверной роли этой операционной системы.

Сервер или сервис? Программа или компьютер? У начинающих пользователей Linux часто происходит некоторая путаница с определением «сервер». Давайте попробуем разобраться, что это такое.

Сервер - это компьютер, предоставляющий определенные услуги, например, услуги веб-сервера, почтового сервера и т.д. Вроде бы понятно. Но что означает фраза «установить веб-сервер»? Нужно взять компьютер и куда-то его установить?

Данная фраза обозначает, что на компьютер нужно установить программу, которая будет обрабатывать HTTP-запросы и отправлять ответы пользователям (HTML-код веб-страничек). Программу также называют веб-сервером (существуют различные программы такого рода, например,

Apache, nginx). Так что же такое сервер - компьютер или программа? Правильнее программу все-таки называть сервисом, то есть веб-сервис (служба). Кстати, управление службами в Linux осуществляется командой service. А сервер - пусть это будет компьютер.

Один и тот же сервер может быть и веб-сервером, и FTP-сервером, и почтовым сервером. Это означает, что на один и тот же компьютер установлены три сервиса (службы): веб-сервис (например, Apache), FTP-сервис (например, proftpd) и почтовый сервис (postfix).

Сервер может быть физическим - большая коробка с мощным блоком питанием, несколькими процессорами, массивом жестких дисков и внушительным объемом оперативной памяти. А может быть и виртуальным - думаю, вы знаете, что такое виртуальная машина. Это когда внутри одной программы - гипервизора - запускается целая операционная система. При этом пользователь не может отличить, к какому серверу он обращается - к физическому или виртуальному. Да и пользователю, по большому счету, все равно - главное, что его запросы обслуживаются, а как это происходит - дело второстепенное.

Сегодня виртуальными серверами никого не удивишь. Виртуальные серверы позволяют эффективно управлять ресурсами физических компьютеров. Дело в том, что в наши дни аппаратные средства существенно опережают свое время. Для среднего сайта вполне достаточно такой конфигурации: один-два процессора (ядра), 1-2 Гб оперативной памяти, 10-25 Гб дискового пространства. Существующие «железные» серверы существенно мощнее - 16-ядерным процессором, объемом оперативной памяти в 32 Гб и дисковым массивом на 2 Тб сейчас никого не удивишь - это довольно скромная, на самом деле, конфигурация.

Так вот, если купить такой железный сервер и на нем развернуть этот самый средней руки сайт, то это будет настоящее расточительство системных ресурсов. Поэтому поступают следующим образом. На этот мощный физический компьютер устанавливают программу гипер-визор, которая сделает возможным запуск нескольких виртуальных машин, конфигурация каждой из них как раз и будет ограничена упомянутыми ранее параметрами. Часть этих виртуальных серверов резервируют для внутренних нужд компании, а часть - сдают в аренду. Таким образом, дорогое физическое оборудование начинает себя окупать. В противном случае оно просто устаревает и теряет в цене.

Почему же Linux идеально подходит для использования в качестве сервера сети? Да потому что (если отбросить такие ее качества как скорость работы и надежность) в состав любого дистрибутива входит программное обеспе-

чение для организации любого сервера. Хотите создать веб-сервер? Пожалуйста! Хотите гипер-визор - нет ничего сложного! Можно даже заменить первичный контроллер домена сети Windows, а рабочие станции даже не почувствуют подмены.

Теперь давайте рассмотрим другие роли операционной системы.

1.6.2. Рабочая станция или домашний компьютер. Настольные и портативные устройства

Почему в сознании многих IT-специалистов, да и обычных пользователей засел тот факт, что Linux больше всего подходит для серверов? Да потому что корни Linux уходят к UNIX, а она является сугубо серверной операционной системой. В 1991 году Линус Торвальдс (как отмечалось ранее) создал ядро, пригодное для запуска на обычных персональных компьютерах с процессором Intel 80386. Но что-то похожее на операционную систему, доступную широким массам энтузиастов, появилось в 1993 году (ближе к концу года). В то время на персональных компьютерах жила DOS с оболочкой Windows. Версия Windows 3.0 появилась в 1990 году, а 3.1 - в 1992-ом. Получается, что у DOS был графический интерфейс уже как три года и была какая-никакая, но многозадачность. Кто работал в Windows 3.11, тот знает, что особой многозадачности там и не было - можно было просто запускать несколько приложений (открывать несколько окон) и переключаться между ними. Но пользователей того времени это устраивало - ведь в один момент времени пользователь может работать только с одной программой.

В Linux же все было не так просто. На Unix-подобных операционных системах того времени использовалась оконная система X Window. Она обеспечивала стандартные инструменты и протоколы для построения графического интерфейса пользователя. Примечательно, но X Window появилась в 1984 году - задолго до появления первой версии оболочки Windows.

Примечание. Обратите внимание: в Linux многозадачность была заложена на уровне ядра, а в Windows - на фоне оболочки, то есть операционная система DOS сама по себе не была многозадачной. Это я уже молчу про то, что Linux была с самого начала многопользовательской операционной системой, то есть позволяла создавать несколько учетных записей пользователей и использовать их для входа в систему, в том числе одновременного входа.

Проблема только в том, что для компьютеров с процессором Intel 80386 эта система в виде проекта XFree86 появилась только в 1992 году. Появиться, то она появилась, но что представлял собой Linux в 1992 году? По сути - только ядро. Первые дистрибутивы, то есть ядро и набор программ, стали появляться в 1993 году, но XFree86 не был включен в их состав. Если мне не изменяет память, то первые дистрибутивы с XFree86 «на борту» появились в 1995 - 1996 гг. Могу, конечно, ошибаться, не хочется тратить время на поиск в Google. Проблема не в том, когда они появились, а в том, что до появления графических окружений KDE и GNOME, а это произошло в 1998 и 1999 гг. соответственно, графический интерфейс Linux выглядел очень убогим. Он был просто ужасен, ужаснее даже чем интерфейс Windows 3.1, особенно на фоне симпатичного интерфейса Windows 95, который уже как три года использовался - чуть ли не во всем мире. Вот отсюда, с 1995 года, и пошел миф о том, что Linux - сугубо серверная операционная система, поскольку у нее нет ни нормального графического интерфейса, ни набора пользовательских программ (вроде графических текстовых процессоров, браузеров и т.д.). Не хочется вас пугать скриншотами Linux тех времен, поверьте на слово - это был ужас. Самые любопытные могут найти скриншоты в Интернете, например, вот в этой замечательной статье https://habrahabr.ru/post/54469/, где приведены иллюстрации разных интерфейсов разных систем - Linux, Windows, OS/2, macOS и многих других.

Получается такая занимательная картина: по состоянию на 1997-ой год у Linux не было ни нормального графического интерфейса, ни нормальных графических пользовательских программ, а на персональных компьютерах Linux использовали только энтузиасты и программисты (ведь весь инструментарий, необходимый для разработки программного обеспечения, входил в состав Linux). Но, с другой стороны, Linux имела низкие системные требования и обладала всем необходимым для построения серверов. По сути, можно было взять любой компьютер с процессором Intel 80486 и на его базе сделать сервер сети, например, организовать совместный доступ к Интернету, развернуть небольшой веб-сервер и даже сделать почтовик. Это и предопределило серверную роль этой замечательной операционной системы.

Однако с тех пор все изменилось. Современные графические интерфейсы KDE и GNOME ничем не уступают интерфейсам современных версий Windows. Подтверждение тому было приведено в предыдущей главе, где были рассмотрены альтернативные графические оболочки для Linux - с графикой теперь в Linux все хорошо.

Многие современные программы были портированы на Linux - офисные пакеты, браузеры, системы CAD, систем для работы с графикой и т.д. Сейчас Linux - это полноценная операционная система, которую можно использо-

вать на рабочих местах пользователей и на домашних компьютерах. В крайнем случае, даже если нужной вам программы нет в Linux (часто такое происходит с играми), можно воспользоваться эмулятором wine для запуска Windows-программ в Linux.

1.6.3. Настольные и портативные устройства

Linux - очень гибкая операционная система. Ее ядро можно настроить для запуска практически на любом устройстве. Не верите? Операционная система Android основана на Linux - посмотрите, сколько устройств работает под ее управлением! Здесь и смартфоны, и планшеты, и небольшие компьютеры, и автомобильные магнитолы и т.д.

Сейчас на рынке наблюдается бум на небольшие устройства и гаджеты. Вот примеры таких устройств: мини-компьютеры, NAS (сетевое хранилище), маршрутизаторы, беспроводные точки доступа и т.д. Все это добро работает, как вы уже успели догадаться, под управлением Linux.

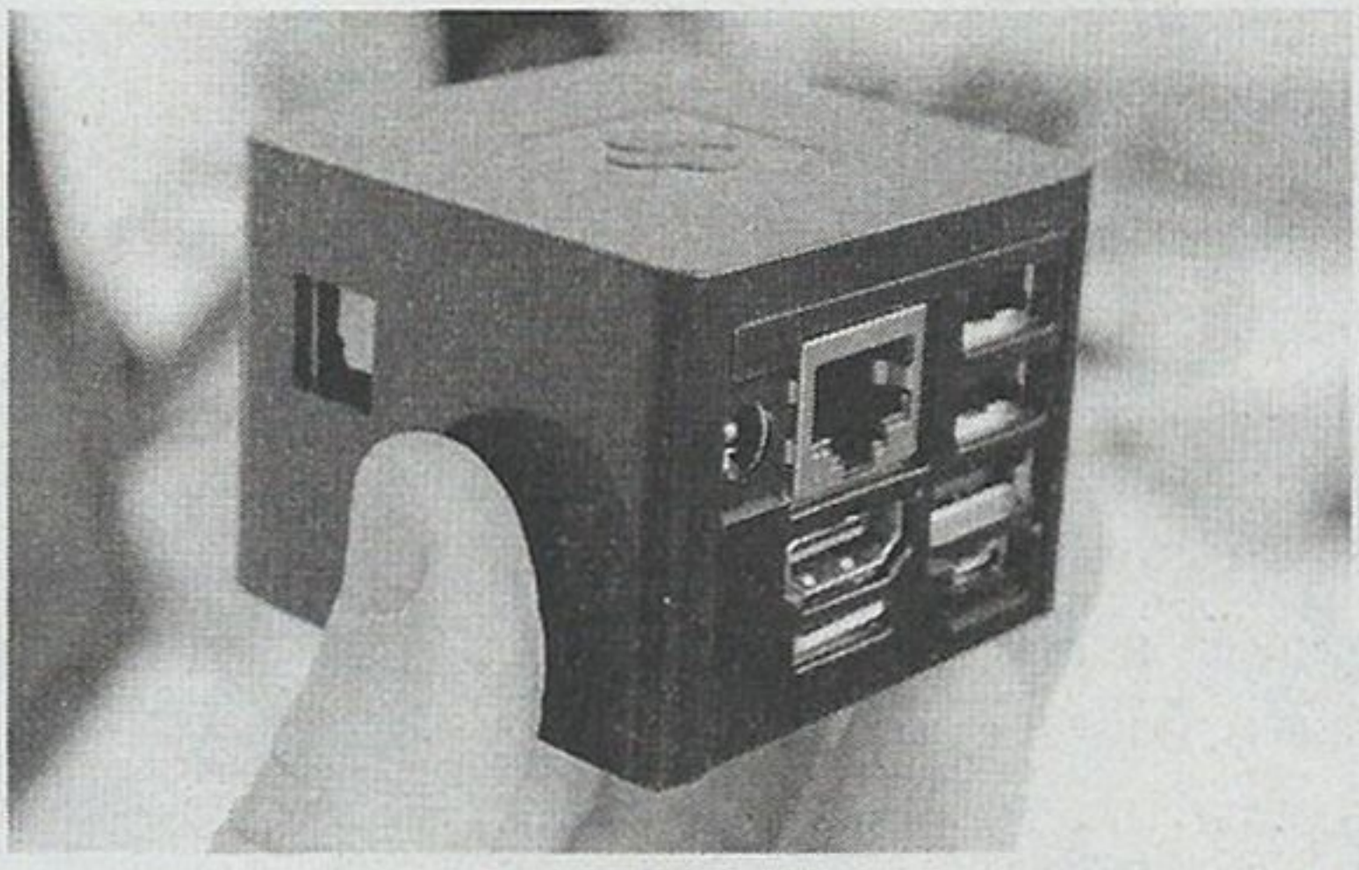


Рис. 1.10. Мини-компьютер

Заинтересовавшиеся читатели могут ознакомиться с обзором лучших мини-компьютеров, работающих под управлением Linux, по адресу

http://howtouselinux.net/best-linux-mini-pc/.

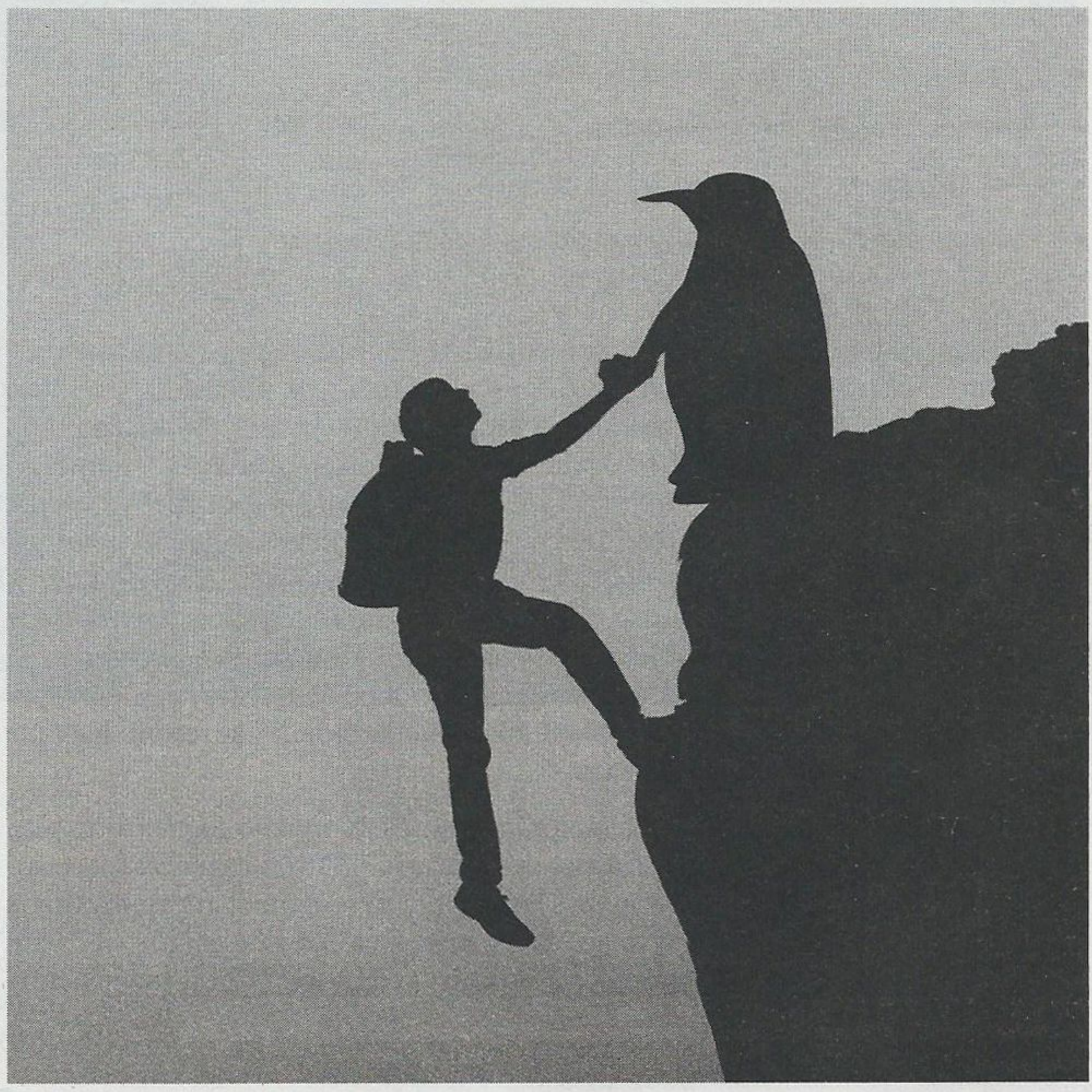
Пусть вас не смущают размеры «малышей» - они обладают довольно мощными процессорами частотой от 1 до 1.6 ГГц, внушительным объемом оперативной памяти 2 - 4 Гб, а также встроенной памятью (eMMC или SSD) размером от 16 до 64 Гб. Как правило, такие устройства обладают HDMI-

портом для подключения к телевизору или монитору, как минимум двумя USB-портами, кардридером и разъемами для подключения наушников и микрофона. Некоторые модели обладают даже разъемом RJ-45 - для подключения к обычной Ethernet-сети.

Стоимость таких устройств тоже невысока. Именно поэтому на них и используется свободная операционная система Linux - чтобы не платить еще и за программное обеспечение, стоимость которого может легко превысить стоимость «железа».

ГЛАВА 2.

Сразу после установки



В этой главе:

- *Вход в систему и завершение работы*
- *Вкратце о настройке сети*
- *Установка кодеков*
- *Тонкая настройка графического интерфейса*
- *Установка новых тем оформления в Ubuntu*
- *Очистка системы от ненужных файлов*
- *Расширение Nautilus для безопасного удаления файлов*

Установка современного дистрибутива Linux, как правило, не вызывает затруднений. Все, что нужно сделать - это загрузиться с инсталляционного носителя и следовать инструкциям инсталлятора. По сути, установка Linux сейчас мало чем отличается от установки обычной программы - запустил инсталлятор и нажимаешь кнопку **Далее**. Время от времени требуется реакция пользователя - например, нужно ввести пароль суперпользователя, добавить учетную запись обычного пользователя, выбрать часовой пояс и т.д. Все инструкции приводятся на русском языке (если пользователь его выбрал в самом начале установки), поэтому их выполнение, как правило, не вызывает никаких проблем. А в некоторых случаях, например, при покупке нового компьютера с предустановленной операционной системой, вообще Linux не приходится устанавливать.

Проблемы возникают сразу после установки. Ведь нужно выполнить минимальную настройку системы, а как это сделать? В этой главе мы рассмотрим 5 вещей, которые нужно сделать сразу после установки дистрибутивов Ubuntu и Fedora. Версии будут рассматриваться самые последние - 17.04 и 27 соответственно. Но сперва мы рассмотрим вход в систему и завершение работы.

2.1. Вход в систему и завершение работы

Все современные дистрибутивы, если они не предназначены для серверного использования, загружаются в графическом режиме. На рис. 2.1 показан экран входа в систему Ubuntu 17.04, но принципы везде одинаковые - есть выбор учетной записи (в нашем случае это ivan и Гостевой вход), после выбора которого можно ввести пароль. При желании можно нажать на пиктограмму логотипа графического окружения напротив имени пользователя и появится возможность выбрать альтернативное графическое окружение, если оно установлено (рис. 2.2).

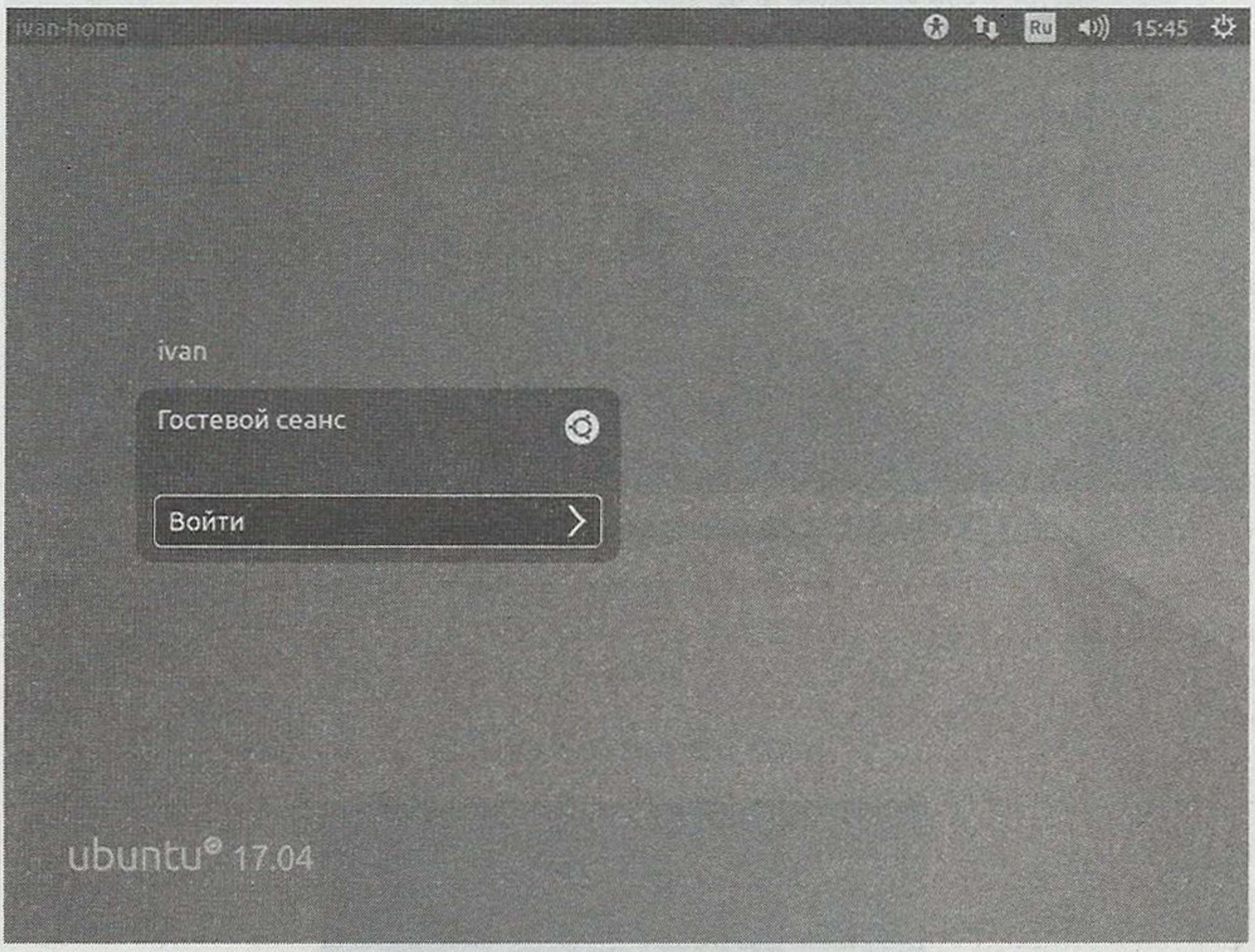


Рис. 2.1. Экран входа в систему

Для завершения работы найдите в меню графической оболочки соответствующую команду. В Ubuntu - это **Выключение** (рис. 2.3), после этого нужно выбрать вариант завершения работы - выключение или перезагрузка (рис. 2.4).

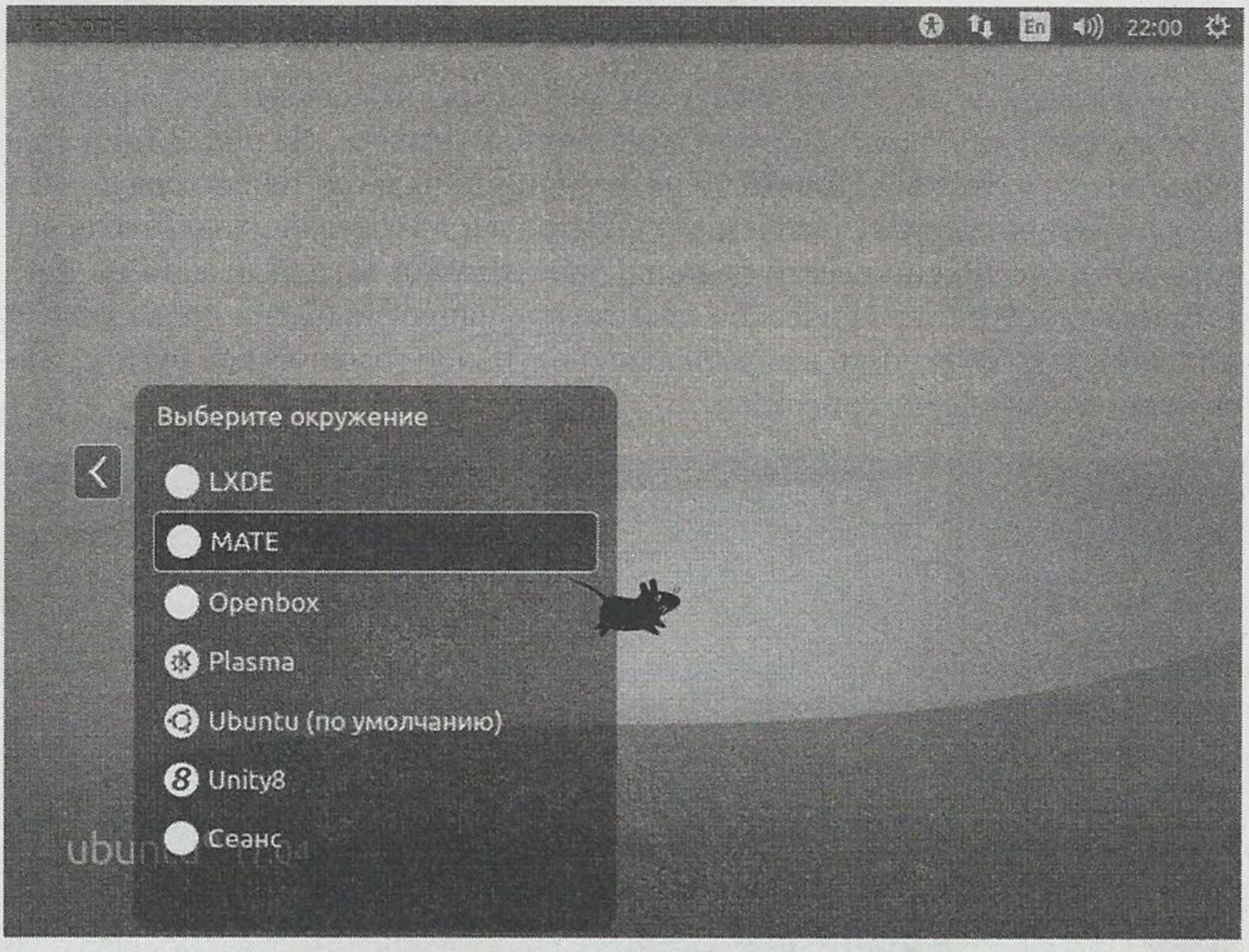


Рис. 2.2. Выбор графического окружения

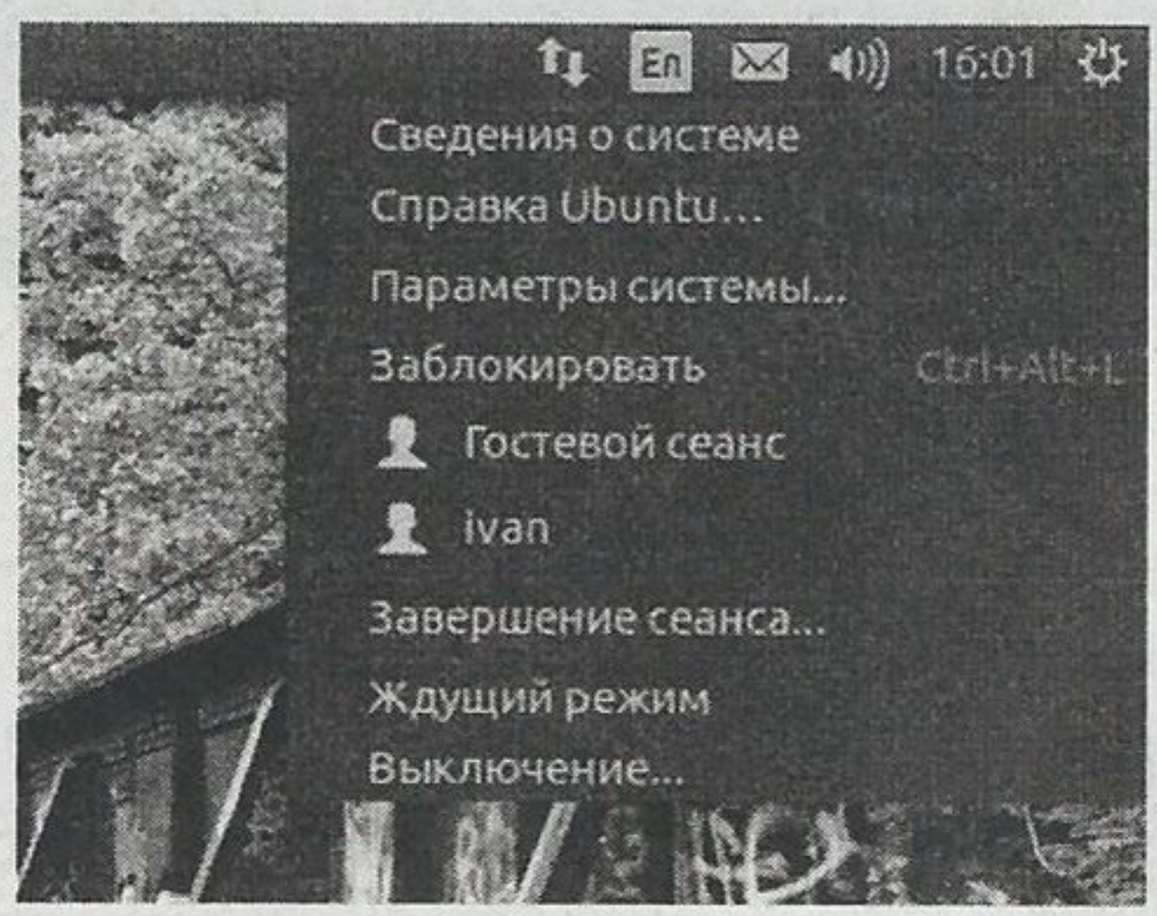


Рис. 2.3. Системное меню

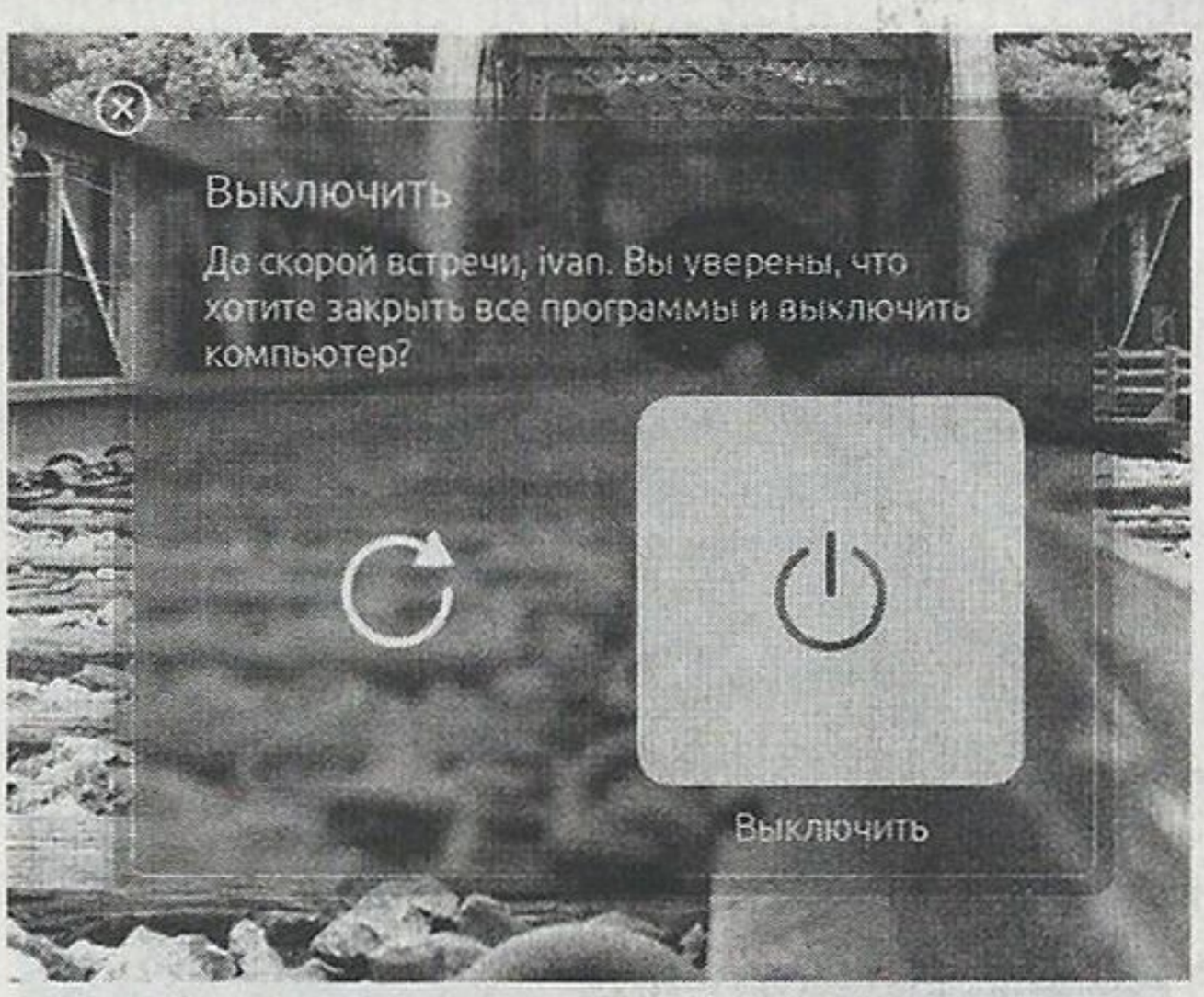


Рис. 2.4. Завершение работы или перезагрузка

2.2. Пять вещей, которые нужно сделать сразу после установки

2.2.1. Настройка сети

Если у вас обычное Ethernet-соединение, которое настраивается по DHCP, то настраивать вам ничего вообще не придется - просто подключите кабель в соответствующий разъем. Беспроводное соединение Wi-Fi тоже легко настраивается - щелкните по значку NetworkManager (для настройки сети во всех современных дистрибутивах используется именно он) в панели уведомлений и выберите одну из доступных Wi-Fi сетей, после чего введите пароль для подключения к ней (рис. 2.5).

Подробно о подключении к Интернету мы поговорим в главе 13.

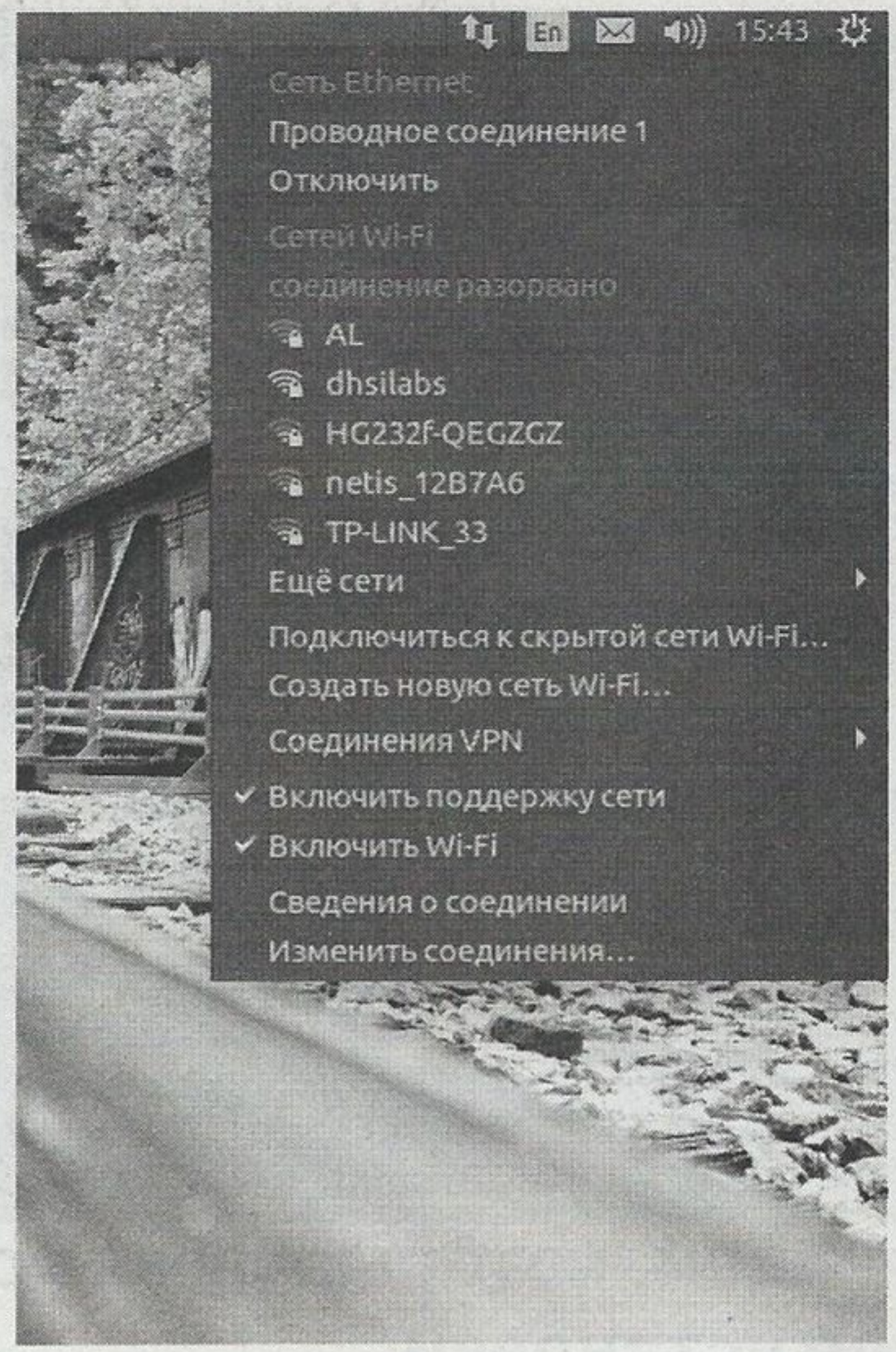


Рис. 2.5. Подключение к Интернету

2.2.2. Установка кодеков

Современные дистрибутивы содержат средства для воспроизведения видео и музыки, но чтобы научить эти средства понимать медиа-форматы, нужно установить кодеки (codec, coder/decoder).

По лицензионным соображениям кодеки не включены в состав дистрибутивов, но могут быть установлены совершенно бесплатно. Получается, что распространять кодеки нельзя, а вот использовать можно. Поэтому дистрибутив поставляется без кодеков, даже в некоторых случаях кодеки не включены в состав официальных репозитариев, нужно подключать сторонние репозитарии и устанавливать кодеки оттуда. Правда, ситуация немного улучшается, например, в Fedora 25 появилась поддержка кодеков MP3 «из коробки», а при установке Ubuntu инсталлятор предлагает скачать пропри-

етарные кодеки (нужно отметить флажок **Установить стороннее программное обеспечение для видеокарт и устройств Wi-Fi, а также Flash, MP3 и др.**).

Звучит устрашающе? На практике не все так страшно. Начнем с Ubuntu. Если вы при установке не установили кодеки, то откройте браузер и введите в него следующий URL:

```
apt://ubuntu-restricted-extras
```

Далее просто следуйте инструкциям инсталлятора. По сути, ничего делать не нужно - просто введите свой пароль и дождитесь установки пакетов.

В Fedora все сложнее. Первым делом нужно подключить репозитарий RPMFusion:

```
# rpm -ivh http://download1.rpmfusion.org/free/fedora/
rpmfusion-free-release-26.noarch.rpm
# rpm -ivh http://download1.rpmfusion.org/free/fedora/
rpmfusion-free-release-27.noarch.rpm
```

Первую команду нужно ввести в Fedora 26, вторую - если у вас Fedora 27. Данные команды подключают репозитарий RPMFusion, содержащий свободные пакеты. Если вы хотите установить, как свободные, так и не свободные (non-free) пакеты, то введите команду:

```
su -c 'dnf install http://download1.rpmfusion.org/free/fedora/
rpmfusion-free-release-$(rpm -E %fedora).noarch.rpm http://
download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-
release-$(rpm -E %fedora).noarch.rpm'
```

Данная команда универсальна - ее можно ввести в любой версии Fedora, начиная с версии 14. Более ранние репозитарием RPMFusion уже не поддерживаются.

Теперь установим необходимые пакеты:

```
su
# Если у вас GNOME
dnf install gstreamer{1,}-{ffmpeg,libav,plugins-
{good,ugly,bad{,-free,-nonfree}}} --setopt=strict=0
# Если у вас KDE
dnf install xine-lib* k3b-extras-freeworld
```

```
# Если планируете слушать Интернет радио онлайн через
rhythmbox:
dnf install gstreamer{1,}-{plugin-crystalhd,ffmpeg,plugins-
{good,ugly,bad{,-free,-nonfree,-freeworld,-extras}{,-extras}}}
libmpg123 lame-libs --setopt=strict=0
# Проигрыватель VLC
dnf install vlc
```

Разберемся, что есть что. Первым делом нужно ввести команду su, чтобы получить права root - без них вы не сможете установить пакеты. Далее установите gstreamer, если у вас GNOME или xine-lib, если у вас KDE. Если планируете слушать Интернет-радио, введите третью команду. Последняя команда устанавливает проигрыватель VLC, обладающий сам по себе поддержкой многих форматов.

2.2.3. Тонкая настройка графического интерфейса

Для Unity и GNOME есть утилиты тонкой настройки графического интерфейса (Tweak Tool). Данные утилиты позволяют настроить графический интерфейс так, как вам захочется. Мне, например, в Unity не нравится, когда ланчер находится слева. Мне было бы удобно, если бы он находился внизу.

Сделать это можно двумя способами - или использовать утилиту тонкой настройки (Unity Tweak Tool) или с помощью команды, введенной в терминале:

```
gsettings set com.canonical.Unity.Launcher launcher-position
Bottom
```

Для установки Unity Tweak Tool откройте Firefox и введите следующий URL:

```
apt://unity-tweak-tool
```

Далее следуйте инструкциям инсталлятора. На рис. 2.6 показано, как она выглядит.

Чтобы изменить положение ланчера перейдите в раздел Панель запуска и выберите позицию Bottom (рис. 2.7). Результат показан на рис. 2.8. Во-первых, так привычнее. Во-вторых, удобнее, поскольку мониторы сейчас, как правило, широкоформатные и при таком расположении панели на ней появляется больше свободного пространства.

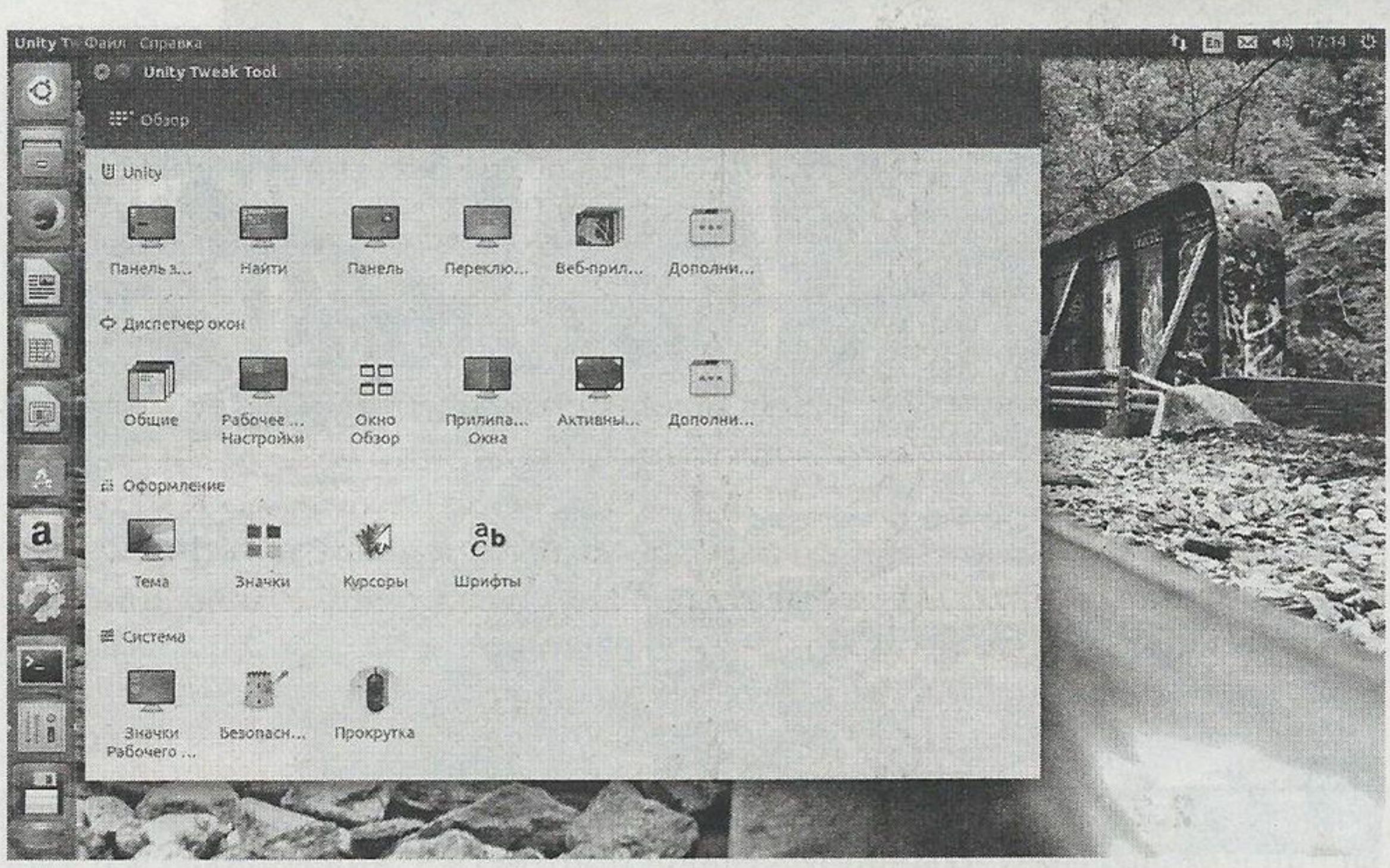


Рис. 2.6. Unity Tweak Tool

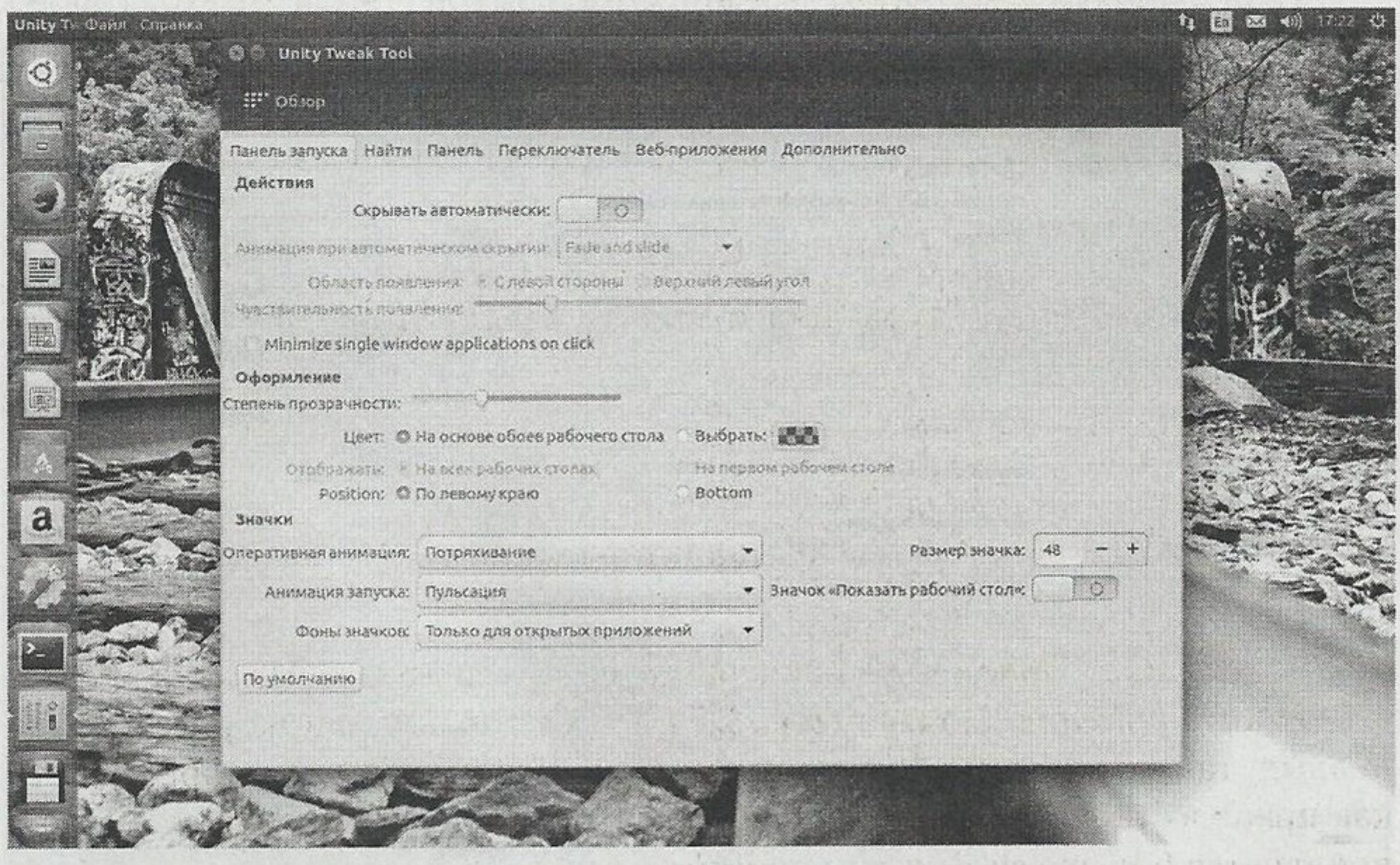


Рис. 2.7. Изменение параметров панели запуска

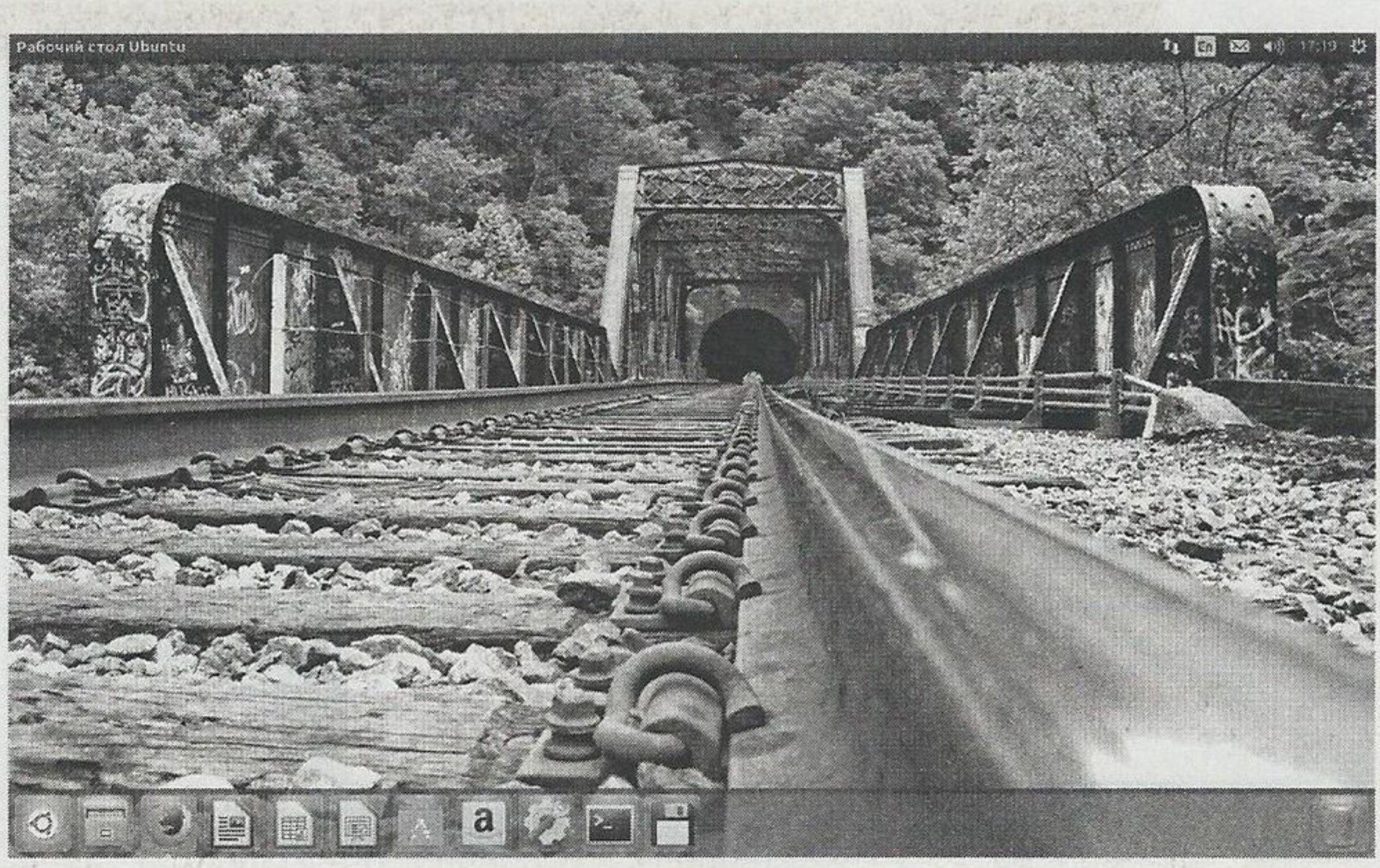


Рис. 2.8. Результат

Для GNOME также есть подобная утилита, которая называется GNOME Tweak Tool и вы без проблем сможете установить ее в Fedora:

https://apps.fedoraproject.org/packages/gnome-tweak-tool

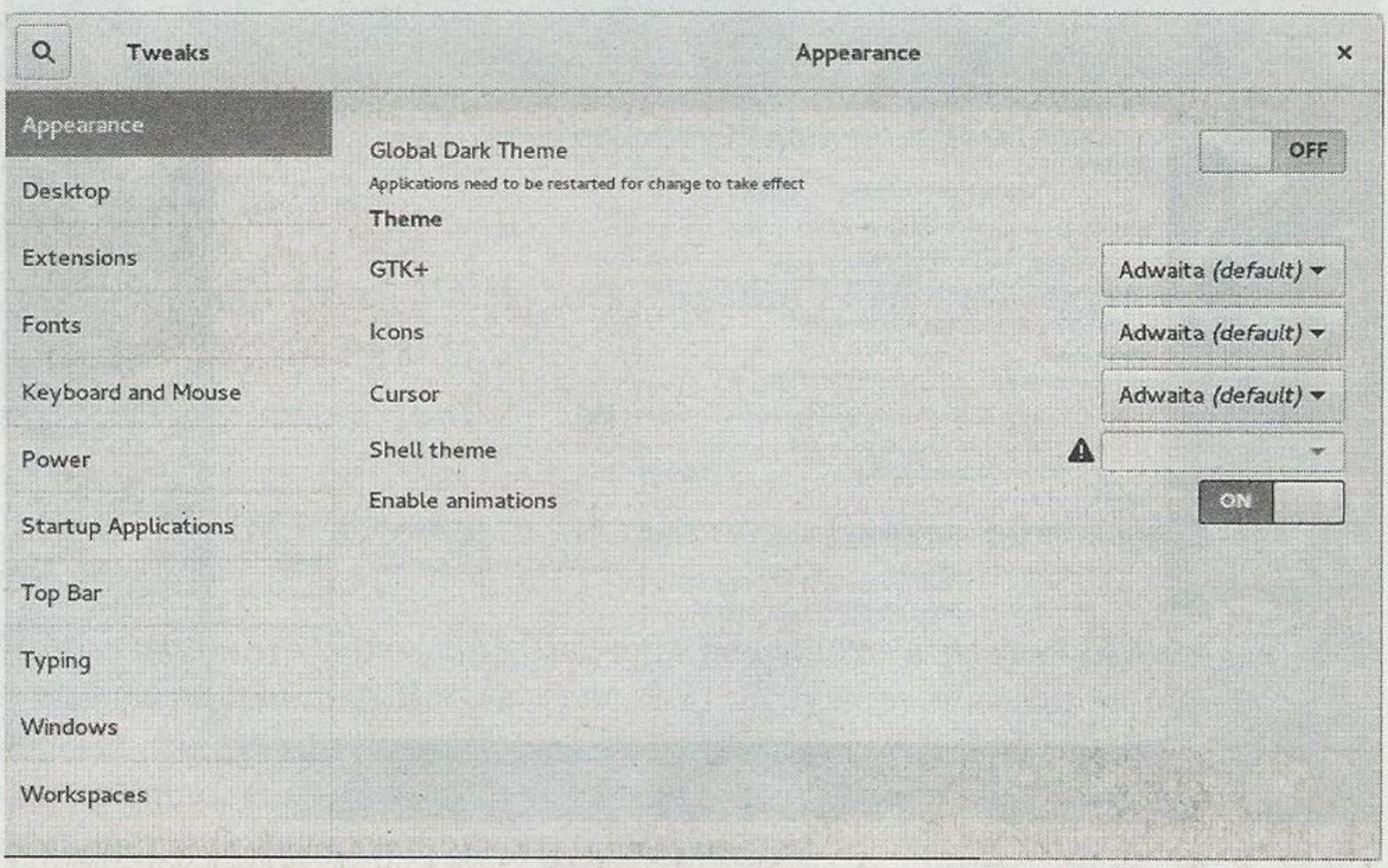


Рис. 2.9. GNOME Tweak Tool

2.2.4. Установка новых тем оформления графического интерфейса

Используя утилиты настройки, вы сможете изменить интерфейс до неузнаваемости. В качестве руководства к действию предлагаю прочитать статью, в которой рассказывается, как сделать Ubuntu/Fedora похожей на macOS:

http://www.omgubuntu.co.uk/2017/03/make-ubuntu-look-like-mac-5-steps

Хотя в статье речь идет об Ubuntu, в ней подразумевается использование рабочего стола GNOME и вся настройка производится над GNOME, а не над Ubuntu. Следовательно, превратить в macOS (разумеется, визуально) можно любой дистрибутив, в котором можно установить рабочий стол GNOME.

Вообще как для Unity, так и для GNOME разработано множество различных тем оформления. Рассмотрим процесс установки темы оформления на примере темы Arc. Для ее установки в терминале (см. гл. 3) введите следующие команды:

```
sudo sh -c "echo 'deb http://download.opensuse.org/
repositories/home:/Horst3180/xUbuntu_16.04/ /' >> /etc/apt/
sources.list.d/arc-theme.list"
```

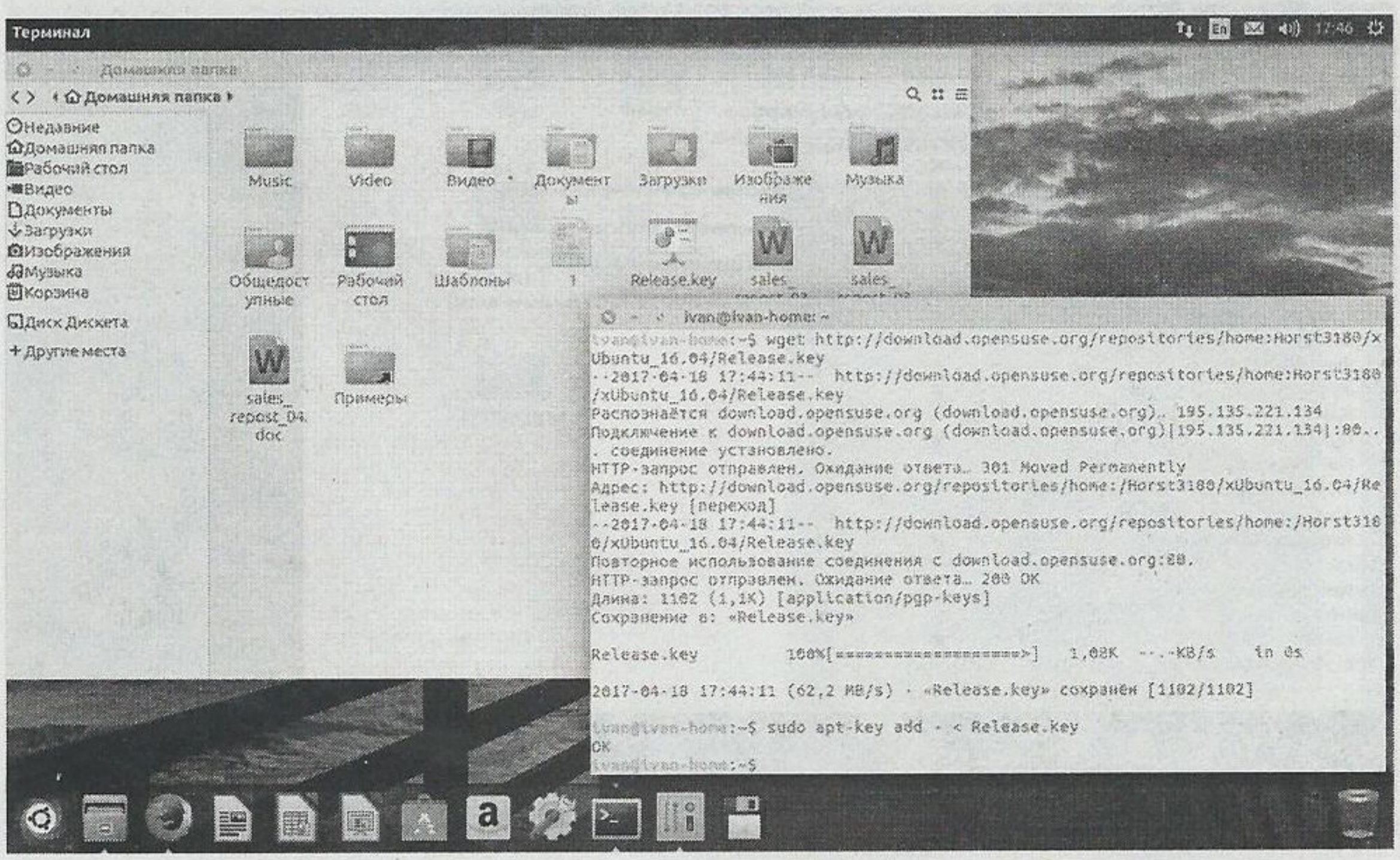


Рис. 2.10. Тема оформления Arc

```
wget http://download.opensuse.org/repositories/home:Horst3180/
xUbuntu_16.04/Release.key
sudo apt-key add - < Release.key
sudo apt-get update && sudo apt-get install arc-theme
```

Далее нужно открыть Unity Tweak Tool и перейти в раздел **Темы**, где и выбрать только что установленную тему. Результат показан на рис. 2.10.

2.2.5. Установка средства очистки системы

В процессе работы системы скапливается множество ненужных файлов. Чего только стоят файлы пакетов, которые загружаются на диск, но не удаляются после установки пакетов. Периодически систему нужно чистить. Одно из лучших средств очистки системы называется BleachBit. Для его установки нужно ввести команду (одну из двух):

```
sudo apt install bleachbit
su; dnf install bleachbit
```

Первая команда установит средство очистки в Ubuntu, вторая - в Fedora.

Утилита в работе изображена на рис. 2.11. Видно, что 225 Мб занимают пакеты установленных мною программ, 30 Мб - файлы Firefox (кэш и другие вспомогательные файлы). Различные ненужные системные файлы заняли

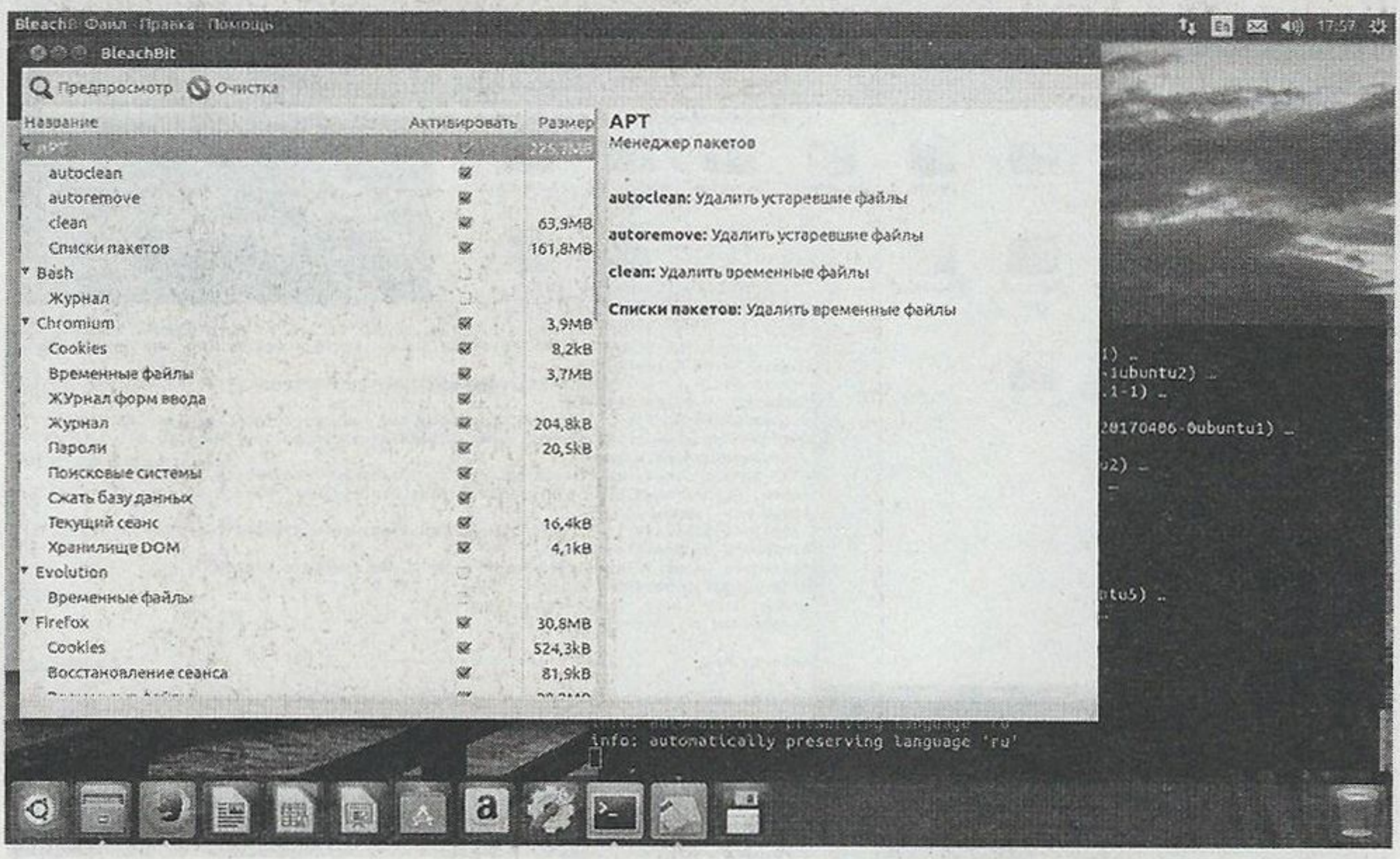


Рис. 2.11. Программа в действии

129 Мб (на рисунке этого не видно). Самое интересное, что все эти файлы появились за один рабочий день и какие-либо большие программы (вроде рабочего стола KDE или GNOME) я не устанавливал. Учитывая, что размер SSD-диска, на который установлена Linux составляет всего 32 Гб, то даже 1 Гб данных будет ощутим.

Итак, выделите то, что хотите очистить и нажмите кнопку **Предпросмотр**. Программа сообщит не только, сколько места занимают выбранные вами объекты, но и покажет, какие файлы собралась удалять. Важно помнить, что очистка Журнала и Паролей браузеров приведет к потере истории и паролей. Если вы преследуете цель - обеспечить конфиденциальность, тогда это приемлемо, в противном случае вы потеряете историю посещенных страниц и все сохраненные пароли. Для очистки, то есть удаления объектов, нажмите кнопку **Очистить**.

Примечание. Раз уж мы заговорили о конфиденциальности, то помните, что при удалении файлов, физически информация не удаляется с носителя и ее можно восстановить (до того, как место, занимаемое файлом не будет перезаписано другим файлом). У всей этой ситуации есть два

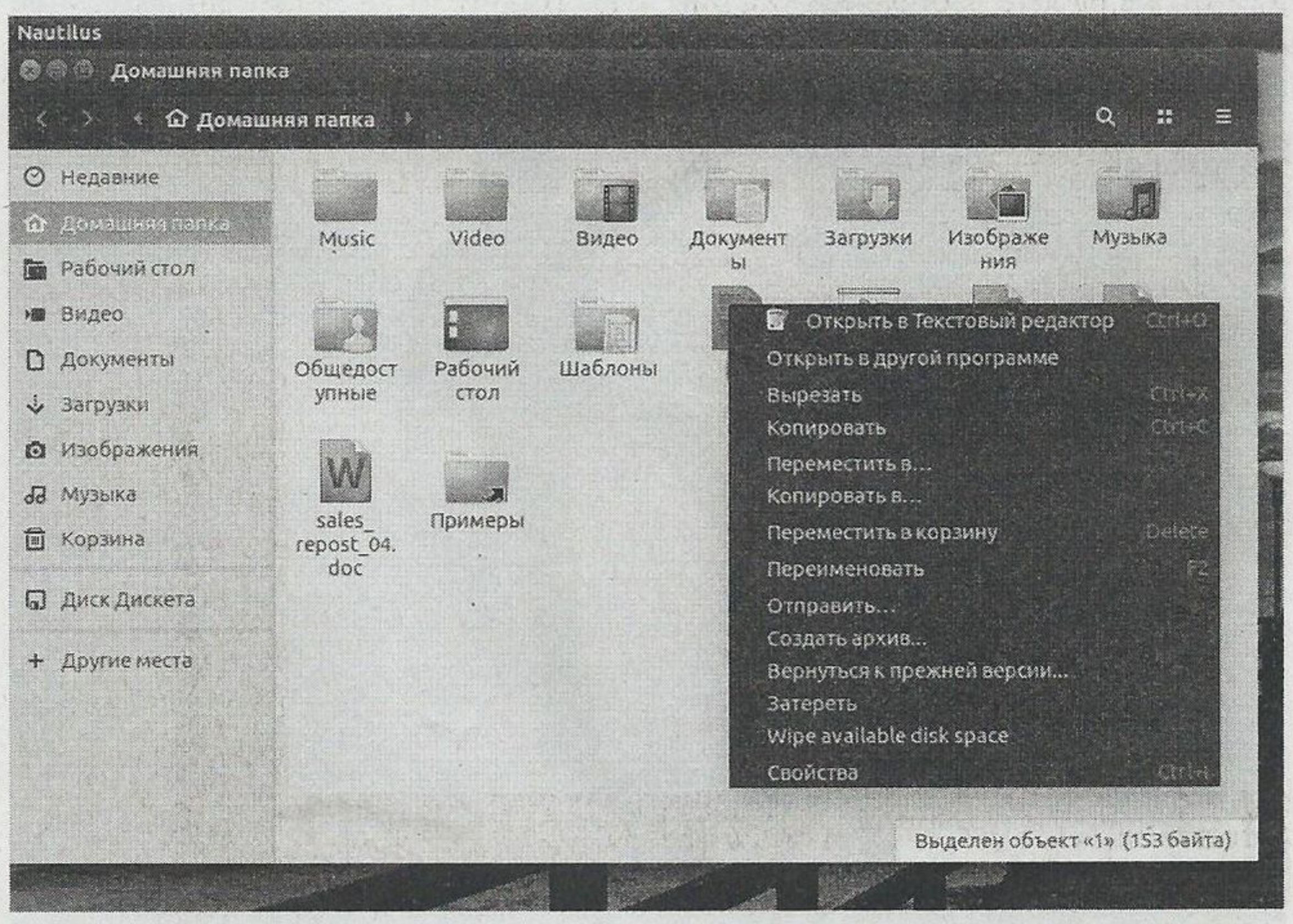
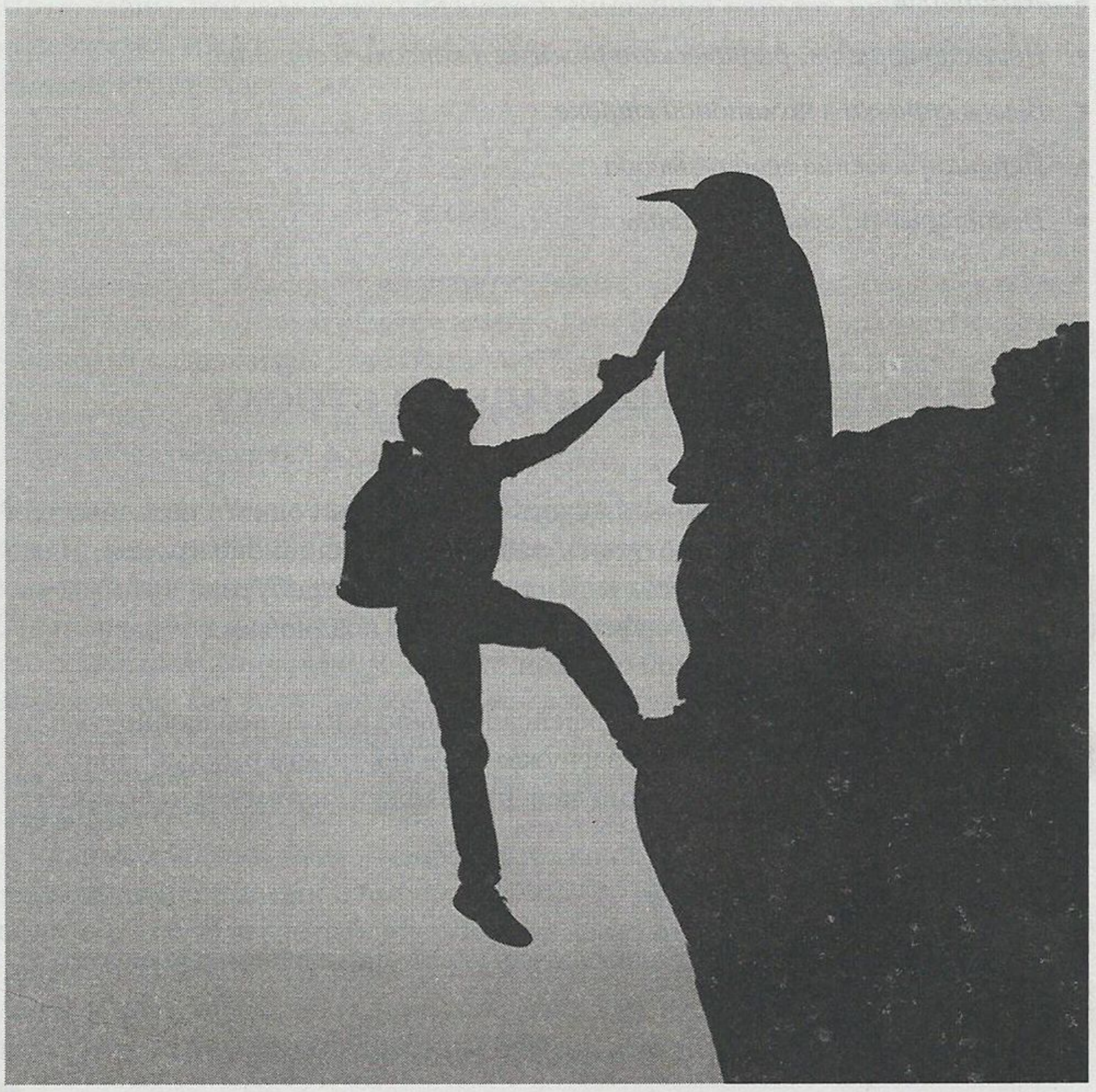


Рис. 2.12. Расширение Nautilus для безопасного удаления данных добавило в контекстное меню две новых команды

побочных эффекта: вы можете легко восстановить случайно удаленный файл и это хорошо. Второй побочный эффект - этот же файл может восстановить кто-то другой и он может содержать конфиденциальную информацию, разглашения которой вы бы не хотели. Для безопасного удаления файлов нужно использовать утилиты вроде nwipe (пакет называется тоже nwipe) - они не только удаляют файл, но и физически перезаписывают занимаемое им место. Можно также установить плагин безопасного удаления файлов для файлового менеджера Nautilus. Пакет называется nautilus-wipe. После его установки выйдите из системы и снова войдите. В контекстном меню файла или каталога появятся две новых команды - Затереть и Wipe available disk space (рис. 2.12). Первая безопасно удаляет выделенный файл, а вторая затирает все доступное дисковое пространство - на случай если вы до этого удаляли конфиденциальные данные обычной командой **Переместить в Корзину** или использовали команду rm.

ГЛАВА 3.

Командная строка



В этой главе:

- *Как получить доступ к командной строке*
- *Терминалы*
- *Переключение из графического режима в консоль и обратно*
- *Вызов справки в командной строке*
- *Перенаправление ввода/вывода*
- *Некоторые полезные команды*

3.1. Введение в командную строку

Командная строка - неотъемлемая часть Linux. Нет ни одного пользователя Linux, которому бы не приходилось работать с командной строкой. Даже если вы установите последнюю версию любого дистрибутива с самым современным графическим интерфейсом, рано или поздно вам все равно придется столкнуться с командной строкой.

Раньше, когда графического интерфейса не было или за ненадобностью он часто не использовался, система загружалась в текстовом режиме - никакой графики. Пользователь видел приглашение входа в систему. Что-то вроде:

```
Login:
```

После ввода логина (имени пользователя) система запрашивала пароль:

```
Password:
```

Если логин и пароль правильные, система запускала командный интерпретатор (оболочку), которая принимала команды от пользователя и выполняла их. Приглашение облочки выглядит или $ (для обычного пользователя) или # (для пользователя с административными правами - root):

```
$
```

Далее пользователь должен ввести команду. Раньше работа с системой осуществлялась только путем ввода команд. Понятное дело, что пользователю нужно было знать сотни различных команд для успешной работы с системой. Сейчас же развит графический интерфейс и многие пользователи совершенно не представляют, что можно ввести в командную строку и для чего используется та или иная команда.

Представим, что вам нужно перейти в домашний каталог, просмотреть его содержимое и скопировать файл sales_repost_03.doc в sales_repost_04.doc. Раньше это делалось так:

```
cd ~
ls
cp sales_repost_03.doc sales_repost_04.doc
```

Первая команда изменяет текущий каталог (~ обозначает домашний каталог пользователя, который обычно имеет путь /home/<имя_пользователя>), ls - выводит содержимое этого каталога, а cp - выполняет копирование.

Сейчас же пользователь использует графический файловый менеджер и даже не задумывается о каких-то командах.

Однако во многих случаях использование командной строки более эффективно, чем графического интерфейса. Приведу конкретный пример. Рассмотрим тот же графический файловый менеджер. Посмотрите на рис. 3.1. Для перехода в домашний каталог вообще не нужно что-либо делать - просто запустите файловый менеджер и он сам отобразит его содержимое. В крайнем случае, если файловый менеджер уже запущен и вы находитесь в каком-то другом месте файловой системы, для перехода в домашний каталог выберите команду **Домашняя папка** на панели слева - один щелчок мыши, что может быть проще? Снова, содержимое каталога отобразится без ввода всякой команды. А теперь попробуйте скопировать файл. Щелкните на нем правой кнопкой мыши и выберите команду **Копировать в**. Затем откроется окно, в котором вы не найдете поля для ввода нового имени, можно выбрать только папку. Если вы выберите домашнюю папку, то файл будет скопирован как sales_repost_03 (копия).doc (рис. 3.2). То же самое произойдет, если вы будете использовать команды буфера обмена - **Копировать** (Ctrl + C)

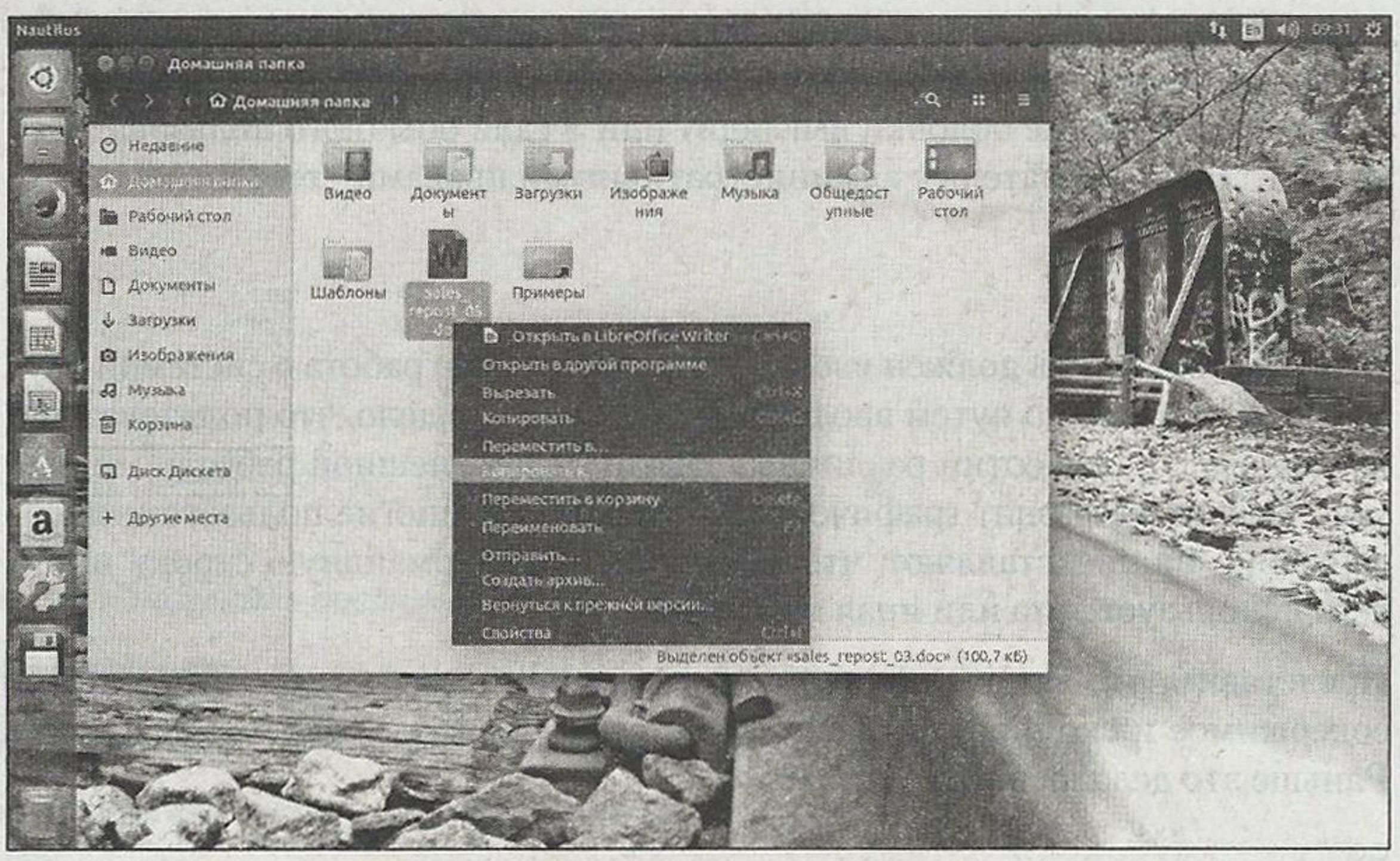


Рис. 3.1. Файловый менеджер

Рис. 3.2. Файл скопирован

и **Вставить** (Ctrl + V). Затем, чтобы получить нужное имя sales_repost_04.doc, вам нужно щелкнуть на файле правой кнопкой мыши, выбрать команду **Переименовать** и ввести новое имя.

Вам не кажется, что вы делаете слишком много лишних действий? Ведь в командной строке вам нужно было бы ввести всего одну команду:

```
cd ~; cp sales_repost_03.doc sales_repost_04.doc
```

Посмотрите на рис. 3.3. Мы ввели одну команду и сразу же получили желаемый результат. Вторая команда (ls) не нужна и мы ее использовали только для вывода содержимого файла, чтобы показать, что файл таки скопировался.

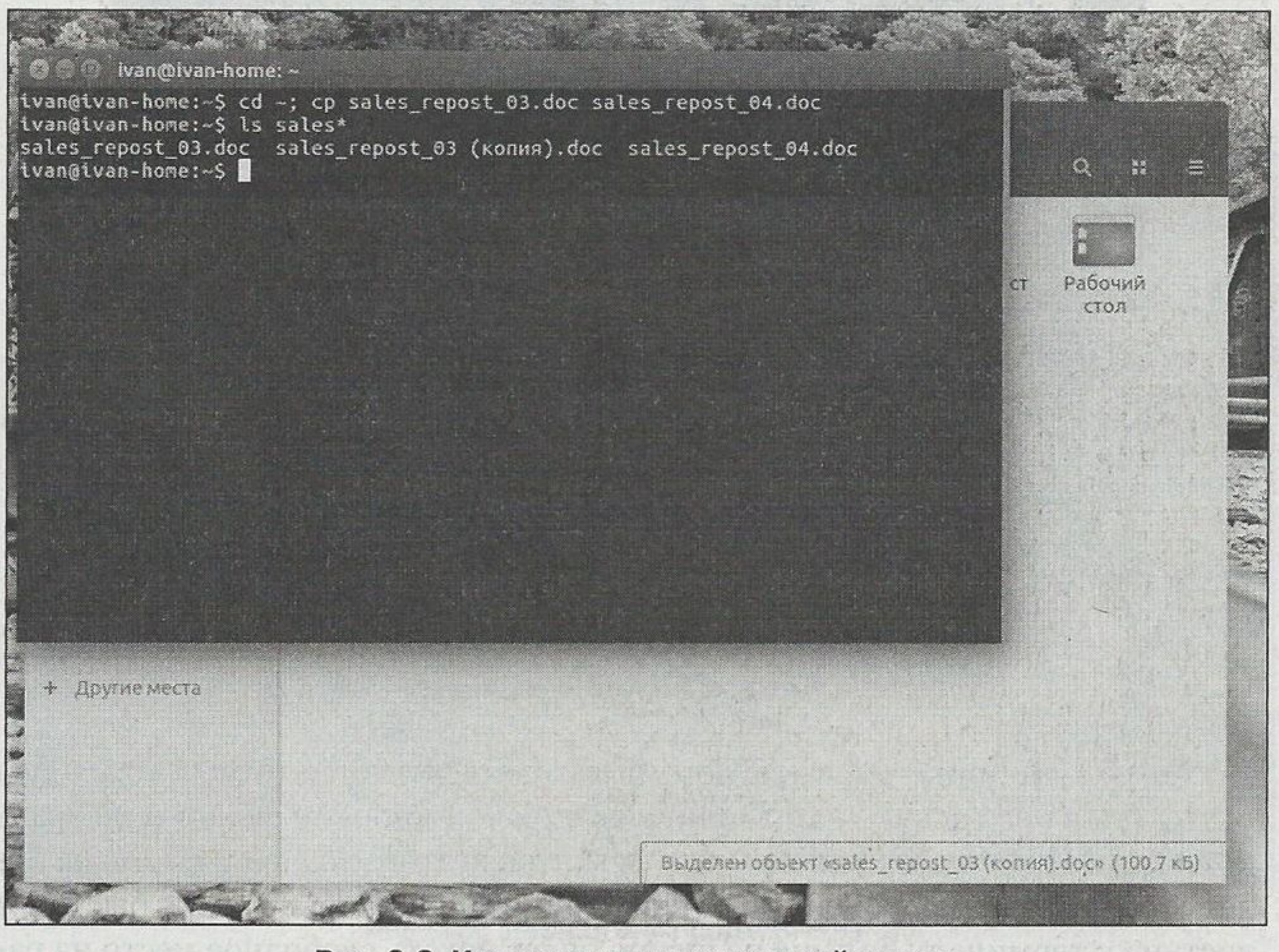


Рис. 3.3. Использование командной строки

Конечно, эффективность использования командной строки зависит от количества команд, которые вы знаете. Рассмотрение всех существующих команд выходит за рамки этой книги, но есть другие книги, посвященные только командам Linux. Далее в этой главе будет показано, как эффективно использовать командную строку.

3.2. Доступ к командной строке

Проще всего получить доступ к командной строке, не выходя из графического режима - запустить приложение Терминал. Вы без проблем найдете это приложение в меню вашего графического интерфейса. В Ubuntu можно также щелкнуть правой кнопкой мыши на рабочем столе и выбрать команду **Открыть терминал** (рис. 3.4).

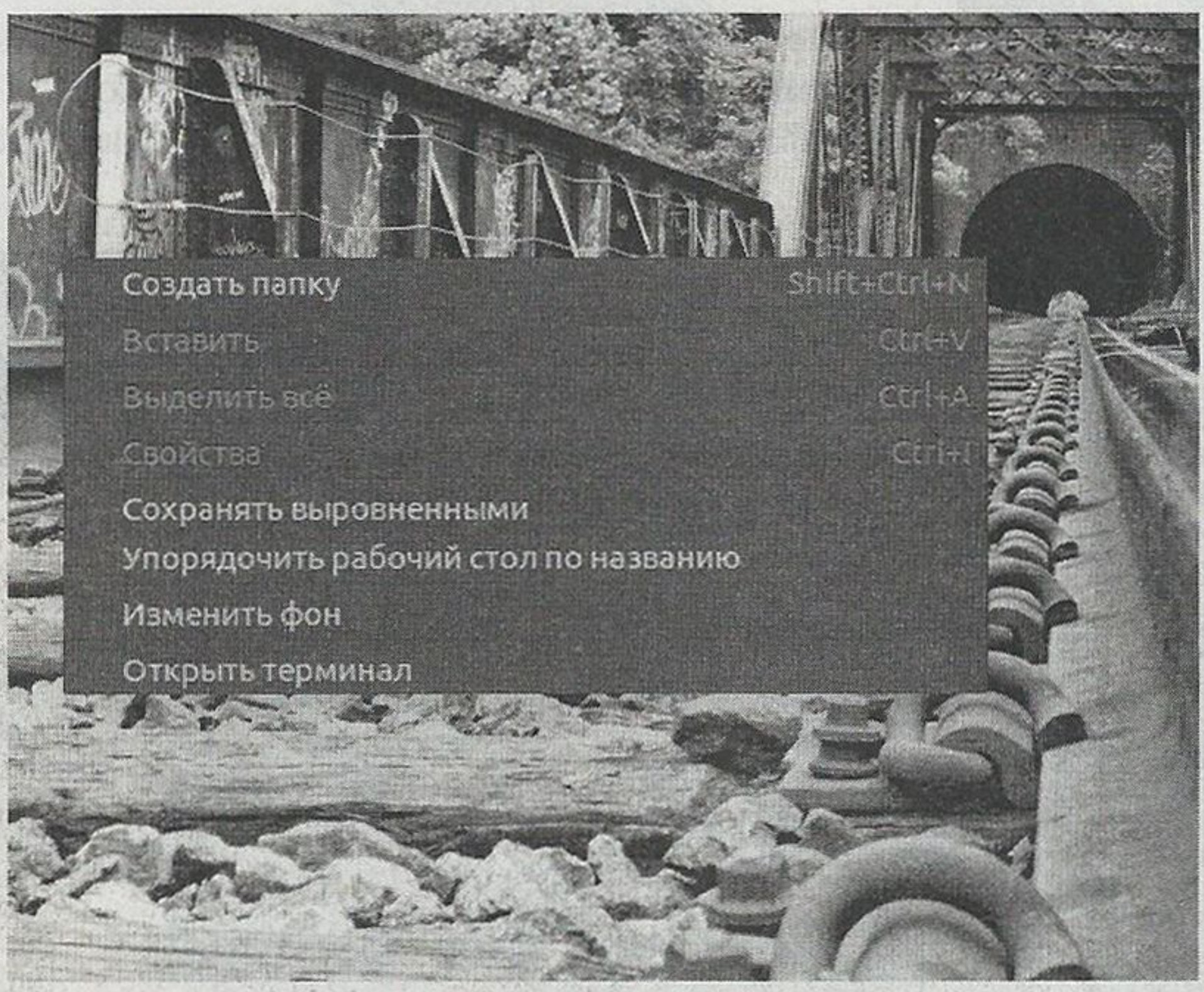


Рис. 3.4. Запуск терминала

Поскольку терминалом вы будете пользоваться часто, то я рекомендую щелкнуть на его значке на панели слева правой кнопкой мыши и выбрать команду **Закрепить на панели** (рис. 3.5). После этого значок терминала будет отображаться на панели, даже если приложение закрыто, а для быстрого доступа к терминалу больше не нужно будет искать свободное место на рабочем столе, чтобы щелкнуть на нем правой кнопкой мыши.

Если же вам хочется «хардкора», то есть вы хотите почувствовать себя настоящим линуксоидом, нажмите комбинацию клавиш Ctrl + Alt + F1. Вы переключитесь в консоль. Никакой графики. Просто текст и все (рис. 3.6).

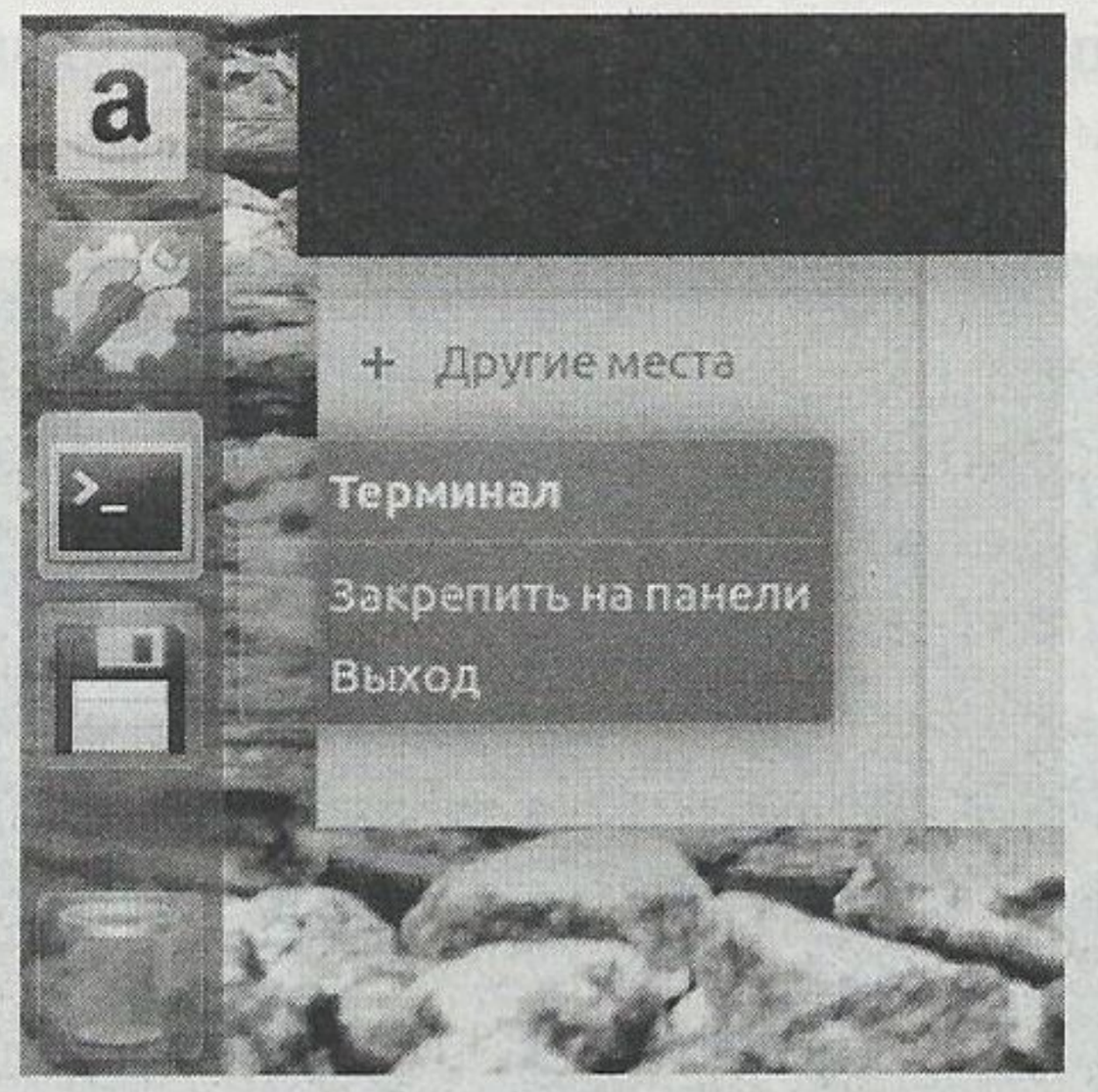


Рис. 3.5. Закрепление значка приложения *Терминал*

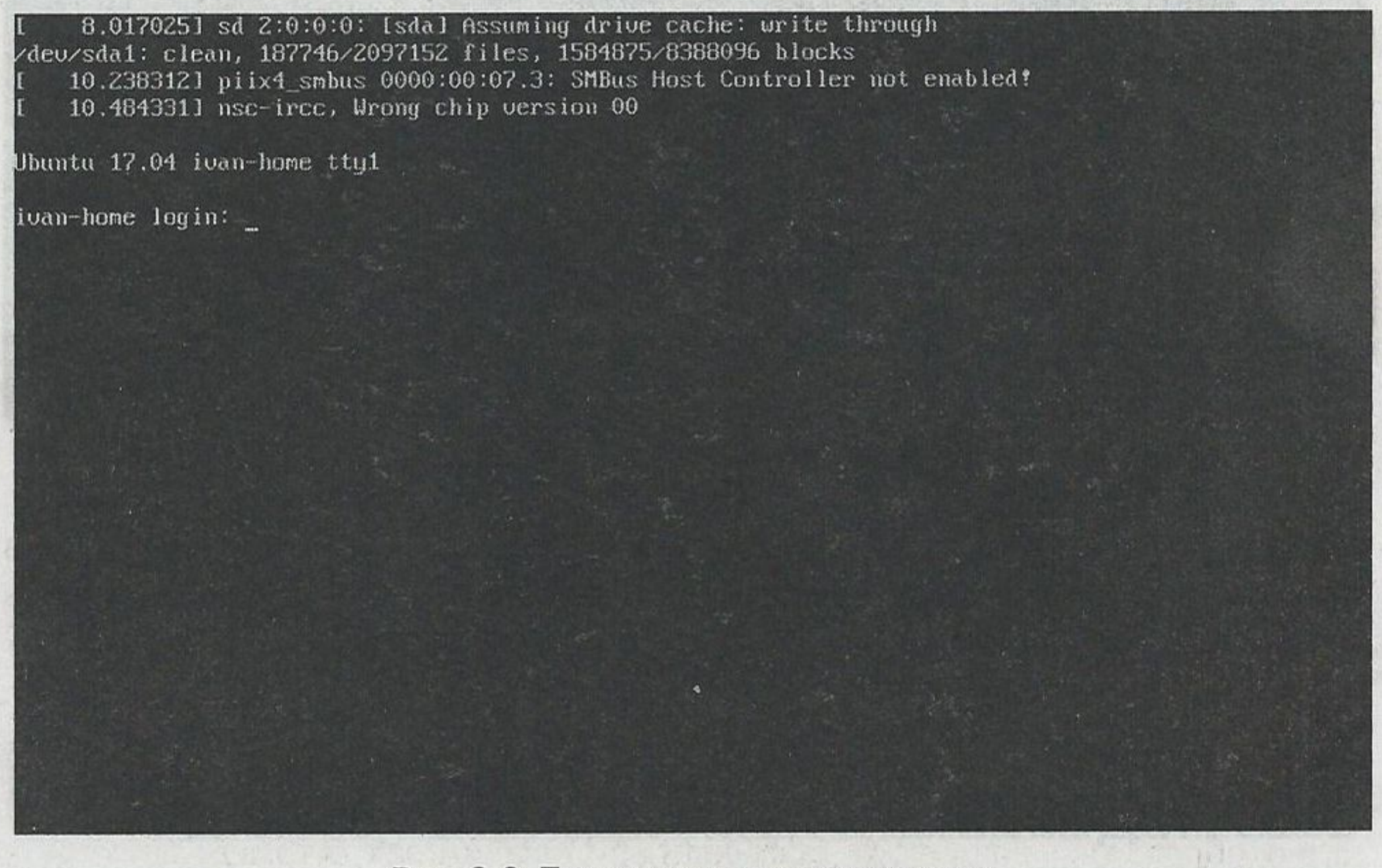


Рис. 3.6. Первая виртуальная консоль

Вам снова нужно будет ввести имя пользователя и пароль. Но вы сможете себя почувствовать настоящим линуксоидом - консоль такая же, как и 20 лет тому назад.

Всего виртуальных консолей 6 - они пронумерованы от 1 до 6. Для переключения между ними используются комбинации клавиш Alt + Fn, где n

- это номер консоли. Например, на рис. 3.7 показано, что пользователь находится на консоли 2 (tty2).

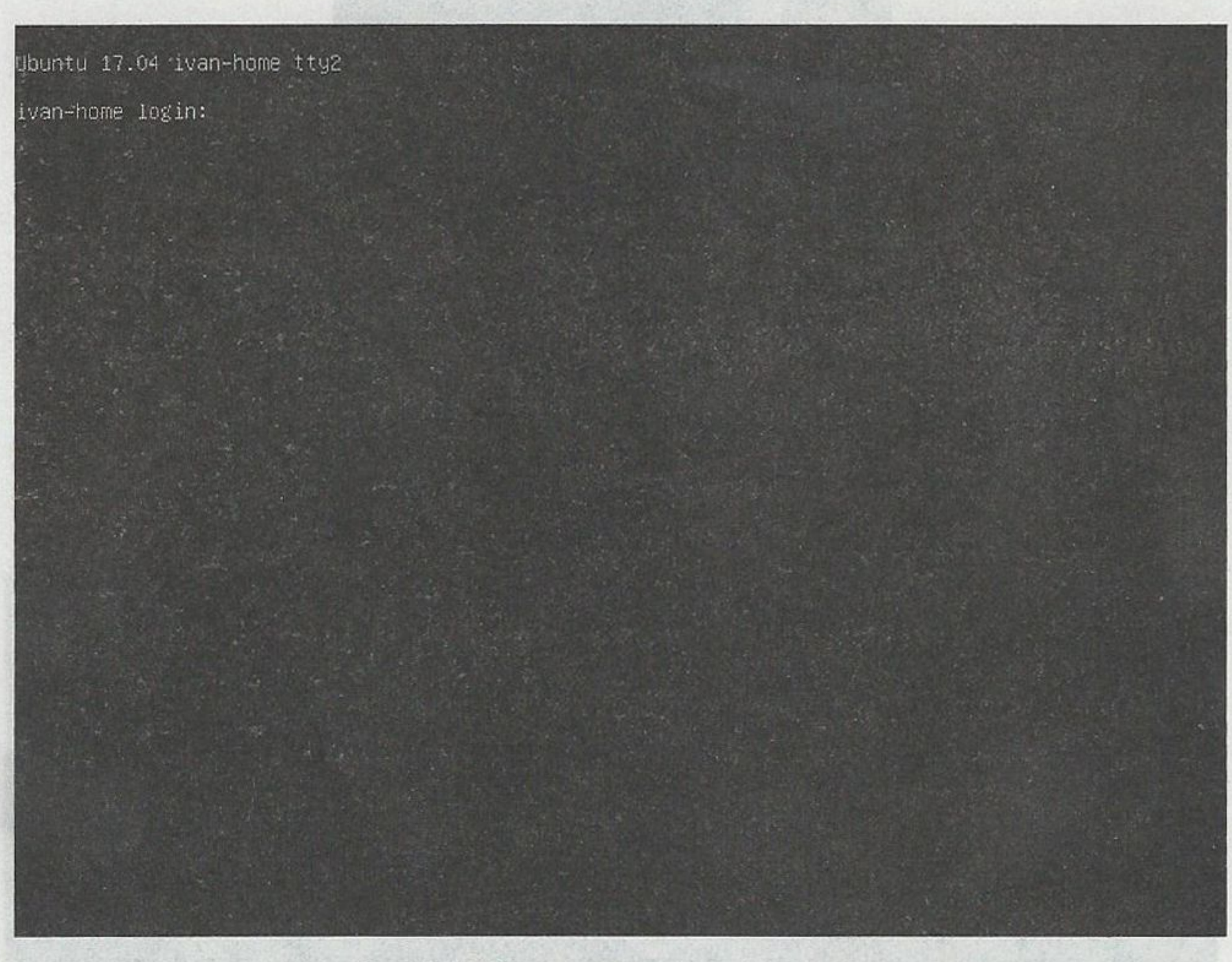


Рис. 3.7. Вторая виртуальная консоль

Вернуться обратно очень просто - нажмите Alt + F7. Вы переключитесь в графический режим. Честно говоря, я не думаю, что вы будете переключаться в консольный режим, тем не менее, я не мог не рассказать вам об этом.

3.3. Подсказка в командной строке и вызов справки

В командной строке поддерживается функция автодополнения команды. Она помогает не только более быстро вводить команды, но и подскажет название команды, если вы его забыли. Нужно знать только начальные символы команды.

Посмотрите на рис. 3.8. Я забыл имя команды, но помню начальные символы - system. Я ввел system и нажал Tab.

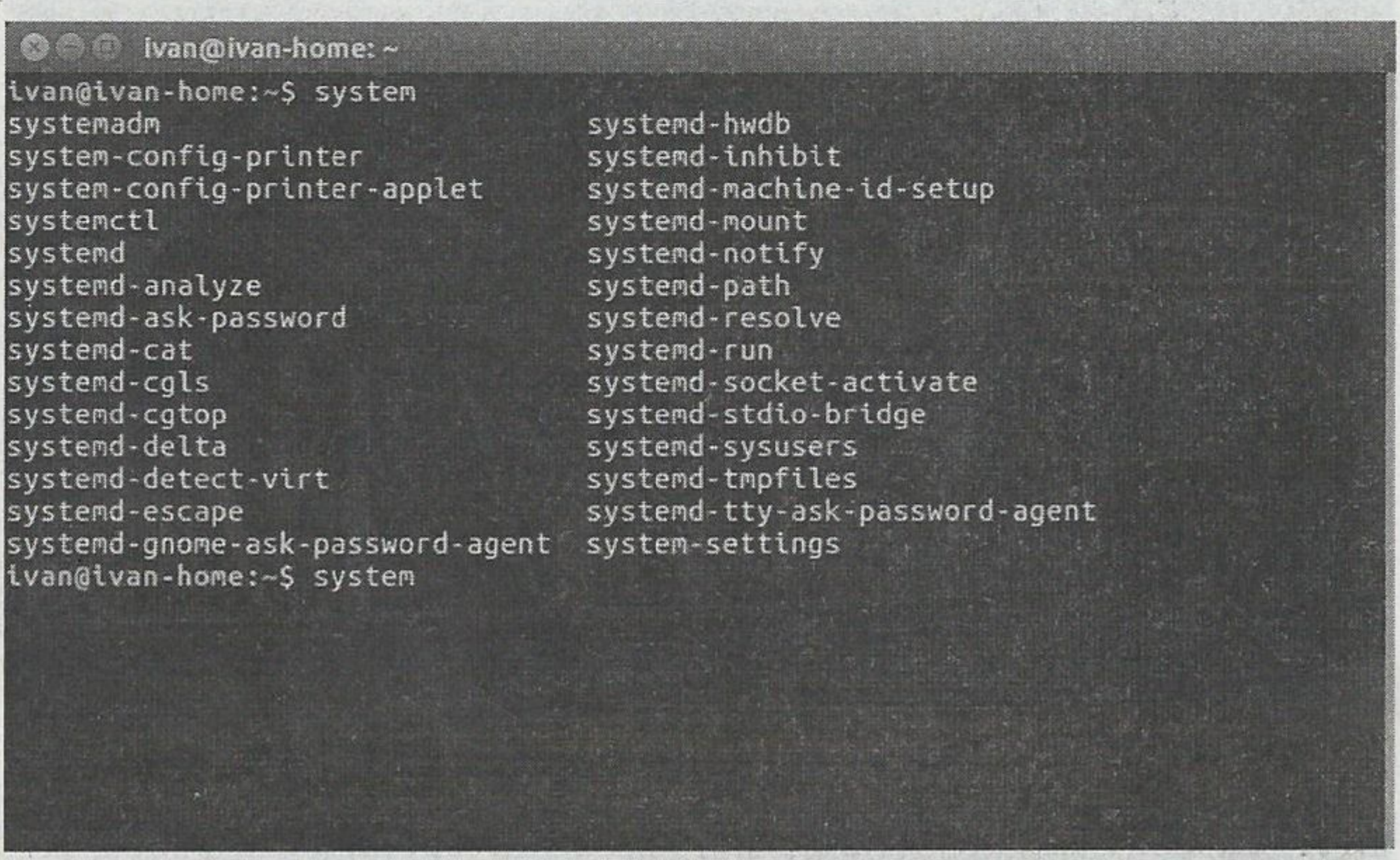


Рис. 3.8. Автодополнение и подсказка

Автодополнение работает так: если после нажатия Tab есть только один вариант команды, то командная строка будет автоматически дополнена. Например, если вы введете systemd-tty и нажмете Tab, то команда будет автоматически дополнена до systemd-tty-ask-password-agent. Вам не придется вводить столь длинную команду.

Если же после нажатия Tab будет обнаружено несколько вариантов, то все они будут отображены, что и показано на рис. 3.8.

У каждой команды может быть очень много параметров. Понятно, что все запомнить пользователь не в состоянии. Именно поэтому каждая команда в Linux тщательно документирована. Для просмотра справки можно или использовать параметр --help (рис. 3.9) или обратиться к справочной системе man:

```
man <имя_команды>
```

К сожалению, есть два неприятных момента и о них я не могу не рассказать:

- Параметр --help поддерживается не всеми программами.

```
ivan@ivan-home: ~
ср: пропущен операнд, задающий файл
По команде «ср --help» можно получить дополнительную информацию.
ivan@ivan-home:~$ cp --help
Использование: ср [КЛЮЧ]… [-T] ИСТОЧНИК НАЗНАЧЕНИЕ
      или:    ср [КЛЮЧ]… ИСТОЧНИК… КАТАЛОГ
      или:    ср [КЛЮЧ]… -t КАТАЛОГ ИСТОЧНИК…
Копирует ИСТОЧНИК в НАЗНАЧЕНИЕ, или несколько ИСТОЧНИКОВ в КАТАЛОГ.

Аргументы, обязательные для длинных ключей, обязательны и для коротких.
  -a, --archive                то же, что и -dPR --preserve=all
      --attributes-only        не копировать данные файла, только атрибуты
      --backup[=КОГДА]         создавать резервную копию каждого целевого
                               файла
  -b                           то же, что и --backup, но без аргумента
      --copy-contents          копировать содержимое специальных файлов при
                               рекурсивной обработке
  -d                           то же, что и --no-dereference --preserve=link
  -f, --force                  если невозможно открыть существующий файл, то
                               удалить его и попробовать ещё раз (этот параметр
                               игнорируется, если также указан ключ -n)
  -i, --interactive            спрашивать перед тем как переписывать (отменяет
                               ранее указанный ключ -n)
  -H                           следовать по символьным ссылкам в ИСТОЧНИКЕ
  -l, --link                   создавать жёсткие ссылки вместо копирования
```

Рис. 3.9. Использование параметра --help

- При использовании параметра --help подсказка отображается на системном языке (в нашем случае - на русском), чего не скажешь о страницах руководства - man. Если программа не выводит подсказку на русском языке, то придется обращаться к man, а страницы руководства man далеко не всегда переведены на русский язык. Раньше в дистрибутивах был пакет manpages-ru, но в последних версиях той же Ubuntu его замечено не было.

Для выхода из справочной системы man нажмите q на клавиатуре.

3.4. Перенаправление ввода/вывода

Вывод некоторых команд бывает очень длинным. Чтобы его было удобнее читать или анализировать (если, например, вывод команды нужно передать другому специалисту), нужно использовать перенаправление ввода вывода:

```
<команда> | <программа>
<команда> > <файл>
<команда> >> <файл>
```

В первом случае вывод команды перенаправляется на стандартный ввод другой программы. Это удобно, если вывод слишком длинный, а вам нужно либо найти что-то в этом выводе, либо просто организовать его постраничный вывод.

Рассмотрим практический пример. При загрузке Linux ведет достаточно подробный журнал всех сообщений ядра. Размер этого журнала примерно 10 Кб. Понятно, что просмотреть за один раз все сообщения ядра вы вряд ли сможете. Поэтому можем использовать следующие команды:

```
dmesg | less
dmesg | grep Intel
```

Первая команда перенаправит вывод dmesg программе less, которая организует удобный просмотр любого текста, отправленного ей. Вторая команда выведет только те строки, которые содержат слово «Intel», то есть с ее помощью я хочу получить информацию обо всем оборудовании компании Intel, которое установлено в системе. В данном случае видно, что процессор, чипсет и сетевой адаптер произведены компанией Intel (рис. 3.10).

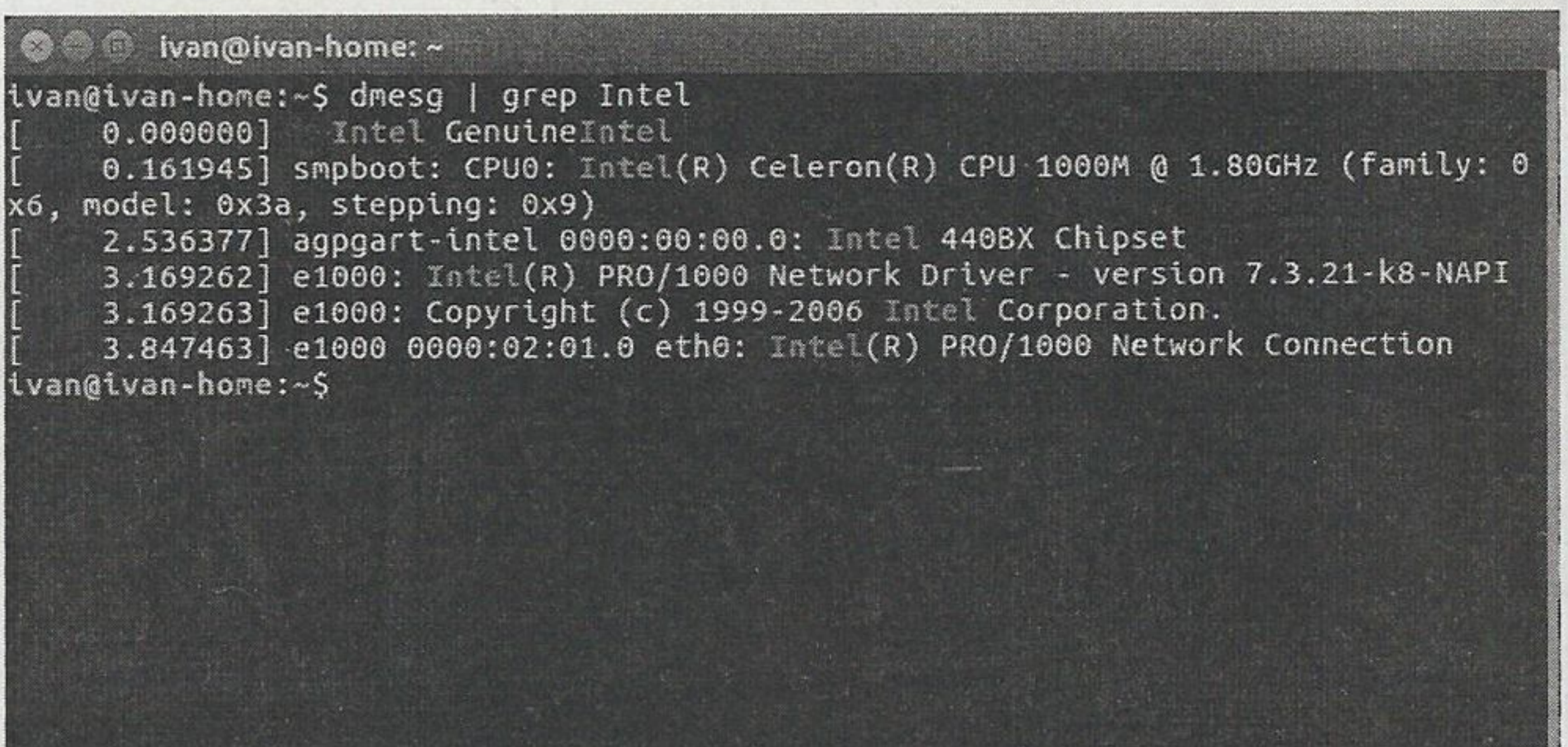


Рис. 3.10. Перенаправление ввода/вывода в действии

Символ > используется для перенаправления вывода в файл, например:

```
dmesg > dmesg.txt
```

В данном случае вывод команды dmesg будет помещен в файл dmesg.txt. Если файл существует, то он будет перезаписан. Если файл не существует, он будет создан.

Если нужно не перезаписывать файл, а лишь добавить в него информацию, тогда используют два символа >:

```
dmesg >> dmesg.txt
```

3.5. Некоторые полезные команды

Как уже было отмечено, данная книга - не справочник по командам Linux. Но все же с некоторыми командами должен быть знаком каждый Linux-пользователь, поэтому было решено сделать небольшую шпаргалку по командам Linux. Подробную информацию о каждой из команд вы сможете найти или в Интернете или на страницах руководства (man). В таблице 3.1 не будут представлены команды для работы с файлами, поскольку они будут рассмотрены в следующей главе.

Таблица 3.1. Полезные команды Linux

Команда	Описание
Завершение работы	
poweroff	Завершить работу системы прямо сейчас и выключить питание
reboot	Перезагрузить систему прямо сейчас
shutdown -r now	Перезагрузить систему прямо сейчас (now), аналогично reboot
shutdown -h now	Выключить компьютер прямо сейчас (now), аналогично poweroff
shutdown -h 20:00	Выключить компьютер в 20:00
shutdown -c	Отмена запланированного выключения/перезагрузки

Получение информации о системе	
arch	Выводит информацию об архитектуре процессора
uname -r	Выводит версию ядра
cat /proc/cpuinfo	Выводит информацию о процессоре
free -h	Выводит информацию об использовании памяти, параметр -h позволяет вывести информацию в мегабайтах, что удобнее для восприятия
df -ḥ	Информация об использовании дискового пространства, в мегабайтах (-h)
cat /proc/net/dev	Показать список сетевых интерфейсов и статистику по ним
date	Выводит системную дату
cal 2017	Выводит таблицу-календарь 2017 года
lspci -tv	Выводит PCI-устройства в виде дерева
lsusb -tv	Выводит USB-устройства в виде дерева
clear	Очищает консоль
Сетевые интерфейсы	
ifconfig ens33	Показать конфигурацию сетевого интерфейса ens33
ifup ens33	Активировать интерфейс ens33
ifdown eth0	Деактивировать интерфейс ens33
ifconfig ens33 192.168.1.1 netmask 255.255.255.0	Установить IP-адрес и сетевую маску интерфейса ens33
dhclient ens33	Активировать интерфейс ens33 в dhcp-режиме (для его автоматической настройки)

route -n	Вывести локальную таблицу маршрутизации
route add -net 0/0 gw IP-адрес	Задать IP-адрес шлюза по умолчанию (default gateway)
route add -net 192.168.0.0 netmask 255.255.0.0 gw 192.168.1.1	Добавить статический маршрут в сеть 192.168.0.0/16 через шлюз с IP-адресом 192.168.1.1
route del 0/0 gw IP-адрес	Удалить IP-адрес шлюза по умолчанию (default gateway)
echo «1» > /proc/sys/net/ipv4/ip_forward	Разрешить пересылку пакетов (forwarding)
hostname	Отобразить имя компьютера
host www . mail.ru	Разрешить имя www.mail.ru хоста в IP-адрес и наоборот (если указан IP-адрес, а не имя узла)
ip link show	Отобразить состояние всех интерфейсов
mii-tool ens33	Показать статус и тип соединения для интерфейса ens33
ethtool ens33	Отображает статистику интерфейса ens33 с выводом такой информации, как поддерживаемые и текущие режимы соединения
tcpdump tcp port 80	Показывает весь трафик на TCP-порт 80 (обычно - HTTP)
iwlist scan	Просканировать эфир на предмет, доступности беспроводных точек доступа
iwconfig eth1	Показать конфигурацию беспроводного сетевого интерфейса eth1
traceroute linux.org	Трассировка маршрута до указанного хоста (linux.org), аналог tracert в Windows. В некоторых дистрибутивах не установлена и нужно установить вручную (с помощью менеджера пакетов).
ping example.com	Позволяет определить доступность (пропинговать) узла example.com

```
ivan@ivan-home: ~
ivan@ivan-home:~$ arch
x86_64
ivan@ivan-home:~$ uname -r
4.10.0-19-generic
ivan@ivan-home:~$ free -h
              всего        занято       свободно      общая  буф./врем.   досту
пно
Память:       1,9G        560M        963M         10M        464M        1,2G
Подкачка:     1,5G          0B        1,5G
ivan@ivan-home:~$ df -h
Файл.система   Размер Использовано  Дост Использовано% Смонтировано в
udev              973M            0  973M             0% /dev
tmpfs             199M          6,9M  193M             4% /run
/dev/sda1          32G          5,5G   25G            19% /
tmpfs             995M           12K  995M             1% /dev/shm
tmpfs             5,0M          4,0K  5,0M             1% /run/lock
tmpfs             995M             0  995M             0% /sys/fs/cgroup
tmpfs             199M          136K  199M             1% /run/user/1000
ivan@ivan-home:~$ cat /proc/net/dev
Inter-|   Receive                                                |  Transmit
 face |bytes    packets errs drop fifo frame compressed multicast|bytes    packe
ts errs drop fifo colls carrier compressed
 ens33:   21868     199    0    0    0     0          0         0    15610     1
48    0    0    0     0       0          0
    lo:   24802     338    0    0    0     0          0         0    24802     3
38    0    0    0     0       0          0
ivan@ivan-home:~$
```

Рис. 3.11. Некоторая системная информация

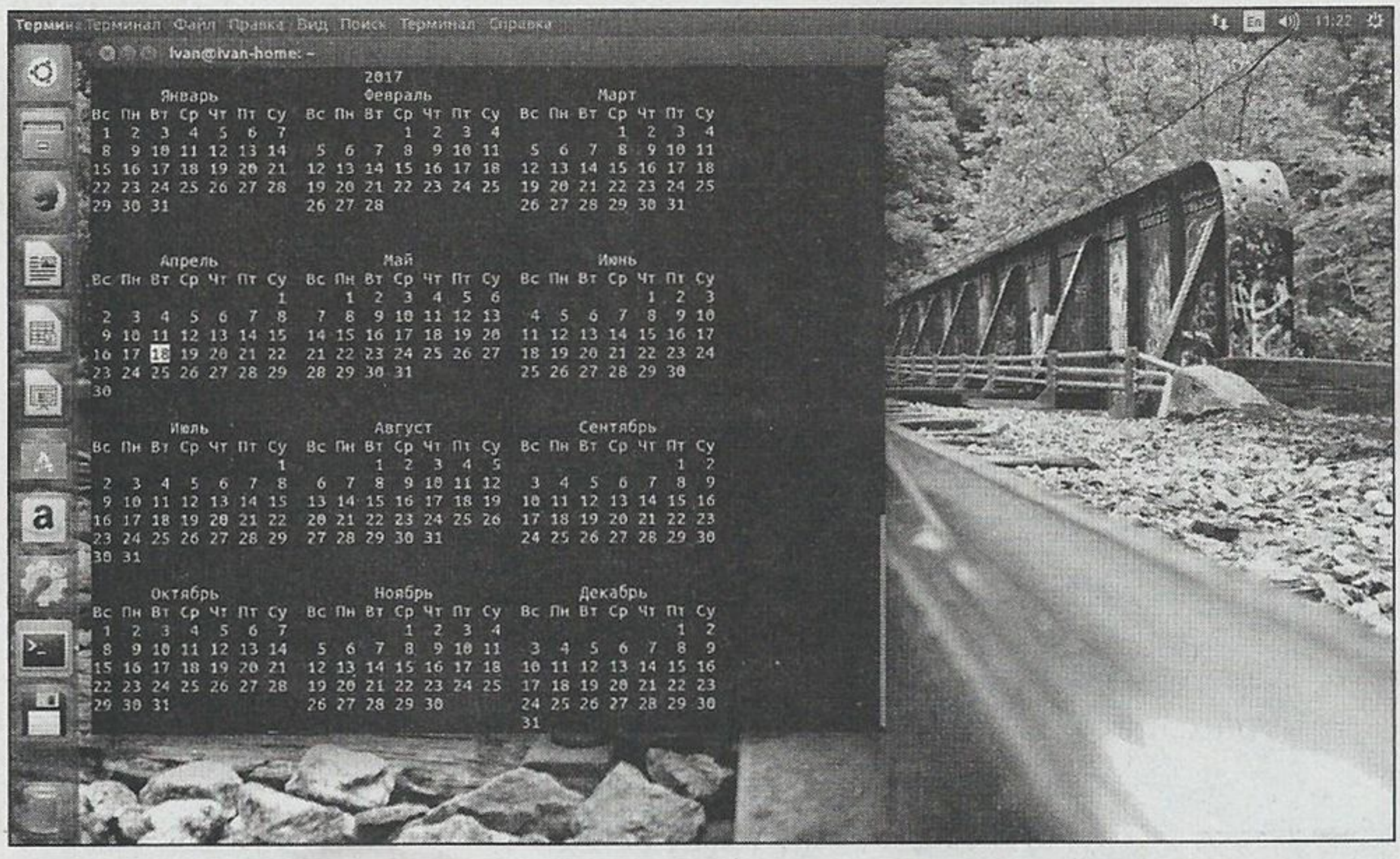
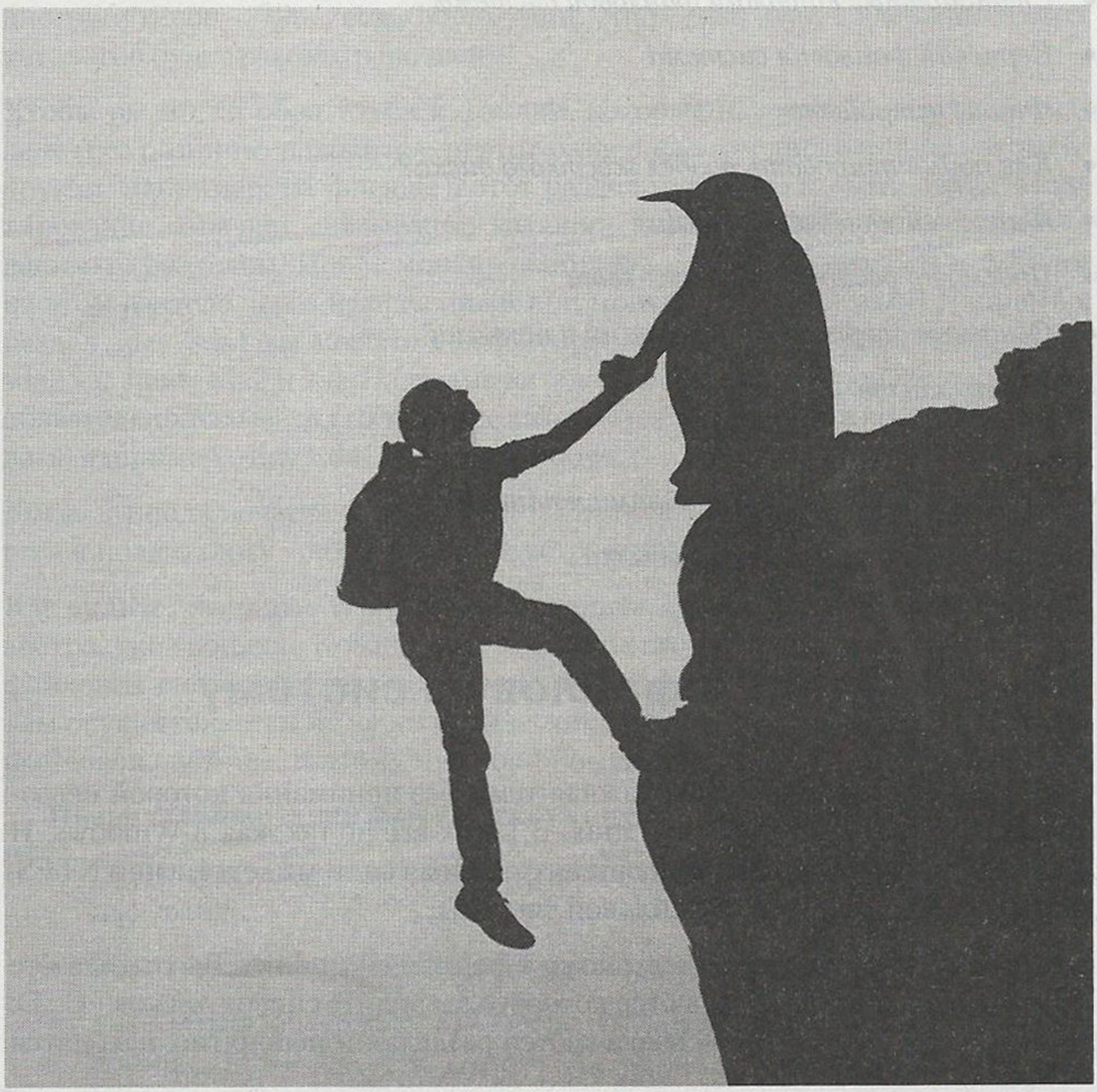


Рис. 3.12. Таблица-календарь 2017 года

Теперь вы точно знаете, что можно делать в командной строке. Надеюсь, данные команды вам пригодятся - пусть не прямо сейчас, а когда-то в будущем.

ГЛАВА 4.

Работа с файлами и каталогами



В этой главе:

- *Особенности файловой системы Linux*
- *Стандартные каталоги файловой системы*
- *Корневая файловая система*
- *Файлы устройств*
- *Как подмонтировать раздел жесткого диска?*
- *Файловый менеджер Nautilus*
- *Просмотр ресурсов Windows-сети*
- *Основные операции над файлами и папками*
- *Поиск файлов*
- *Архивация*
- *Команды для работы с файлами и папками*
- *Жесткие и символические ссылки*

4.1. Введение в файловую систему

Файловая система - еще одна важная тема, без понимания которой невозможна продуктивная работа с Linux. В Linux все не так, как в Windows. И дело не в том, что используется другая файловая система - ext4, а не в NTFS. Дело в самой организации файловой системы.

Вспомним, как осуществляется доступ к файлам в Windows. Вы открываете Проводник, в разделе Компьютер которого вы видите список дисков - C:, D: и т.д. Интересно, что буквы назначаются разделам и непонятно, находятся

ли они на одном физическом диске или на разных. Согласитесь, у вас может быть два жестких диска, на каждом из которых может быть по одному разделу. Аналогично, у вас может быть только один диск с двумя разделами. В обеих ситуациях будет логических диска - C: и D:. Узнать точное расположение каждого можно с помощью оснастки Управление компьютером.

В Linux есть понятие корневой файловой системы. Корневая файловая система создается на разделе диска, который был выбран при установке системы. Представим, что в системе есть только один диск. Тогда имя у него будет /dev/sda. Путь на диске созданы несколько разделов (не важно, какая файловая система используется) - /dev/sda1, ... /dev/sda7. При установке Linux система был выбран раздел /dev/sda5. Тогда на этом разделе будет создана корневая файловая система, которая обозначается как /. Она содержит базовый набор каталогов, который практически одинаковый для всех дистрибутивов - /bin, /etc, /home, /mnt, /tmp, /usr, /var. Содержимое этих каталогов будет рассмотрено далее.

Чтобы вы могли обратиться к файлам, которые находятся на другом разделе (без разницы, находится ли этот раздел на этом диске или на каком-то другом, например, на флешке), этот раздел нужно подмонтировать. Монтирование означает связывание каталога корневой файловой системы с каким-то разделом. После монтирования вы можете обратиться к файлам на этом разделе через назначенный при монтировании каталог. Например, раздел /dev/sda1 вы можете подмонтировать к каталогу /mnt/sda1, /dev/sda2 - к /mnt/sda2 и т.д. Название каталога может быть произвольным - не обязательно называть его так, как называется раздел. Можно подмонтировать, например, /dev/sda1 к /mnt/windows.

Когда вы подключаете флешку или USB-диск, он автоматически монтируется к подкаталогу /media/<UUID-устройства>.

Как видите, разница с Windows кардинальна. Если в Windows есть какое-то абстрактное понятие Компьютер, то в Linux его можно сравнить с корневой файловой системой (хотя сравнение очень «натянутое»). Доступ к разделам осуществляется не через логические диски, а через каталоги корневой файловой системы. К этому нужно привыкнуть.

Правила именования файлов и каталогов:

- Linux чувствительна к регистру символов: file.txt и FILE.txt - два разных файла.
- Максимальная длина имени (с расширением) - 254 символа.
- Имя может содержать любые символы (в том числе и кириллицу), кроме / \ ? < > * «|.

- Для разделения пути вместо \ используется символ /

4.2. Файловые системы Linux

В качестве корневой файловой системы может быть использована не только ext4, но и много других разных файловых систем - ext3, XFS, JFS, Reiser, ZFS, Btrfs и некоторые другие.

Выбор файловой системы зависит от предпочтений разработчиков дистрибутивов и от поставленных задач. По умолчанию в настольных версиях дистрибутивов используется, как правило, файловая система ext4, поскольку ext3 устарела, а все остальные довольно специфические. В серверных версиях могут быть использованы другие файловые системы. Так, в Fedora Server используется файловая система XFS. Она поддерживает высокую производительность - до 7 Гбайт/с. На настольных компьютерах, учитывая производительность дисковой подсистемы, ее использовать нет смысла - даже если у вас SATA III, то максимальная пропускная способность составит всего 600 Мбайт/с

Отдельного внимания заслуживает файловая система Reiser. Она позволяет хранить в одном блоке несколько файлов. Другие файловые системы могут хранить в одном блоке максимум один файл. Если размер блока равен 4 Кб, а файл занимает 2 Кб, то оставшиеся 2 Кб используются впустую. Reiser позволяет устранить этот недостаток и использовать дисковое пространство более эффективно. Если у вас много мелких файлов, то целесообразно использовать файловую систему Reiser. Из личного опыта могу добавить, что эта файловая система хорошо себя зарекомендовала на веб-сервере, где была необходимость хранения множества мелких файлов - файлов сессий. Было столько мелких файлов, что заканчивалось не дисковое пространство, а доступные иноды, в результате чего сервер зависал, поскольку невозможно было произвести запись на диск. Переход на Reiser позволил решить эту проблему.

4.3. Стандартные каталоги корневой файловой системы

В таблице 4.1 приводится набор стандартных каталогов файловой системы Linux. В зависимости от дистрибутива, могут быть небольшие отличия, но в основном набор каталогов будет таким, как показано в таблице.

Таблица 4.1. Стандартные каталоги корневой файловой системы

Каталог	Описание
/	Корневой каталог
/bin	Здесь «живут» стандартные программы Linux вроде cat, ls, cp, mv и т.д.
/boot	Каталог загрузчика GRUB2, содержит конфигурацию загрузчика и другие файлы, относящиеся к загрузке операционной системы. Будет рассмотрен в главе 14.
/dev	Содержит файлы устройств
/etc	Содержит конфигурационные файлы системы. Аналог реестра в Windows.
/home	Содержит домашние каталоги пользователей
/lib	Здесь находятся библиотеки и модули
/lib64	64-разрядные модули и библиотеки
/lost+found	Сюда помещаются восстановленные после некорректного размонтирования файловой системы файлы
/media	Содержит точки монтирования сменных носителей (дисков DVD, USB-дисков)
/mnt	Традиционно этот каталог используется для монтирования других разделов
/opt	Опциональный каталог, некоторое программное обеспечение (обычно то, которое отсутствует в репозитории) устанавливается в этот каталог. В вашей системе может отсутствовать.
/proc	Каталог псевдо-файловой системы /proc. Служебный каталог, обычному пользователю здесь делать нечего.
/root	Домашний каталог пользователя root

/run	Файлы, информация в которых соответствует действительности только до очередной перезагрузки системы. Например, файл /run/utmp содержит информацию о пользователях, подключенных к системе в данный момент. В некоторых дистрибутивах этот каталог находится не в корневой файловой системе, а в каталоге /var.
/sbin	Содержит системные утилиты. Запускать утилиты из этого каталога может только пользователь root
/tmp	Каталог с временными файлами
/usr	Сюда устанавливаются пользовательские программы, документация, исходные коды программ.
/var	Постоянно изменяющиеся данные - журналы системы, почтовые ящики, базы данных, файлы веб-сервера, очереди печати.

Примечание. Обратите внимание, что в Linux для разделения пути используется символ слэша /, а не обратного слэша \, как в Windows. Пример пути к файлу /home/ivan/film.avi.

4.4. Файлы устройств

Примечательно, но все устройства в Linux представлены в виде файлов. У каждого устройства есть соответствующий ему файл. Все файлы устройств находятся в каталоге /dev. Так, вашему приводу DVD соответствует имя /dev/dvd, консоли - /dev/console и т.д.

Все это очень занимательно, но нас сейчас интересуют файлы устройств, относящиеся к дисковым накопителям.

Раньше была некоторая путаница, связанная с наличием трех интерфейсов - IDE, SATA, SCSI. Сейчас же интерфейс SATA вытеснил другие интерфейсы, поэтому все диски называются однотипно - /dev/sdX, где X - буква от a до z, соответствующая порядку подключения физического диска к контроллеру. Если диск подключен к контроллеру первым, то его имя будет /dev/sda, вторым - /dev/sdb и т.д. Если у вас только один диск, то его имя будет /dev/sda. Когда вы подключаете флешку или USB-диск, то ему будет присвоено имя /dev/sdb.

На каждом диске может быть несколько разделов. Им присваиваются номера, начиная с 1 - /dev/sda1, /dev/sda2 и т.д.

4.5. Монтирование

Как уже отмечалось, для получения доступа к файлам и каталогам, находящимся на другом разделе, этот раздел нужно подмонтировать. Монтирование сменных носителей осуществляется автоматически. Обычно сменные носители монтируются к каталогу /media/<UUID-устройства>, UUID - это уникальный идентификатор устройства.

Разделы, находящиеся на том же диске, что и корневой раздел, и на других дисках, установленных в компьютере, могут не монтироваться по умолчанию. В этом случае их нужно подмонтировать вручную. Монтирование осуществляется с помощью команды mount:

```
sudo mount <имя_устройства> <точка_монтирования>
```

Точка монтирования — это каталог, через который будет осуществляться доступ к монтируемой файловой системе. Обычно монтирование осуществляется к одному из подкаталогов каталога /mnt, но это не обязательно. Вы можете выбрать для монтирования другой каталог.

Монтирование требует прав root. В Ubuntu поэтому нужно запускать команду монтирования через команду sudo. В Fedora нужно сначала получить права root командой su:

```
su
mount /dev/sdb1 /mnt/files
```

Также обратите внимание, что на момент выполнения операции монтирования точка монтирования (то есть каталоге /mnt/files в предыдущем примере) должна существовать.

Для размонтирования файловой системы используется команда umount:

```
sudo umount <устройство или точка монтирования>
```

Скорее всего, у вас возникнет необходимость монтирования NTFS-раздела, если на компьютере установлена также Windows. Для монтирования NTFS-раздела нужно указать опцию -t ntfs-3g, то есть явно указать тип файловой системы. В некоторых случаях, если раздел NTFS не был размонтирован

правильно (например, был отключен свет), подмонтировать его можно только с помощью опции force:

```
sudo mount -t ntfs-3g /dev/sdb1 /mnt/files -o force
```

Чтобы при каждой загрузке не вводить команды монтирования, точки монтирования можно «прописать» в файле /etc/fstab. Формат записей следующий:

```
устройство точка_монтирования тип опции РК ФП
```

Здесь: тип — это тип файловой системы, а РК — флаг резервного копирования. Если он установлен (1), то программа dump заархивирует данную файловую систему при создании резервной копии. Если не установлен (0), то резервная копия этой файловой системы создаваться не будет. ФП устанавливает, будет ли данная файловая система проверяться на наличие ошибок программой fsck.

Поле опции может содержать различные параметры файловой системы. Некоторые из них:

- auto - должна монтироваться автоматически при загрузке, используется по умолчанию, поэтому ее можно не указывать.
- noauto - монтироваться при загрузке не должна, удобно использовать, если вы не хотите, чтобы тот или иной раздел монтировался автоматически, но для монтирования вручную нужно будет указать только имя устройства - больше не нужно указывать ни точку монтирования, ни опции.
- defaults - используется стандартный набор опций, которые установлены по умолчанию.
- exec - разрешает запуск выполняемых файлов на данной файловой системе.
- noexec - запрещает запуск выполняемых файлов на данной файловой системе, полезно установить эту опцию для файловых систем Windows.
- ro - монтирование в режиме « только чтение».
- rw - монтирование в режиме «чтение/запись».
- user - данную файловую систему разрешено монтировать и размонтировать обычному пользователю.
- nouser - файловую систему разрешено монтировать только пользователю root, используется по умолчанию.

- utf8 - нужен для корректного отображения символов в именах файлов в современных системах.

Загляните в ваш файл /etc/fstab. В нем вы найдете строку вида:

```
UUID=0a23abd6-d721-4def-a56c-c7a7457160ab / ext4
errors=remount-ro 0  1
```

Давайте разберемся, что она означает. Устройство можно задать в виде имени файла устройства, например, /dev/sda1. А можно использовать UUID-устройства. Преимущество использования UUID очевидно. Идет привязка именно к конкретному устройству, а не к его имени.

Представим, что у вас есть стационарный компьютер, у него есть жесткий диск с UUID 0a23abd6-d721-4def-a56c-c7a7457160ab, который был подключен как первый диск к SATA-контроллеру, то есть имя у него было /dev/sda. Вы отключили диск, например, чтобы подключить к компьютеру друга для обмена объемными данными. Затем вы ошибочно подключили диск как второй, ему было присвоено имя /dev/sdb. Если бы в fstab вы использовали имена вида /dev/sda1, то при загрузке система бы не смогла найти устройство /dev/sda1, поскольку вместо него теперь устройство /dev/sdb1. Использование UUID исключает такую неприятную ситуацию.

Просмотреть привязку простых имен вида /dev/sdXn к UUID можно с помощью команды:

```
sudo blkid
```

Результат выполнения этой команды приведен на рис. 4.1.

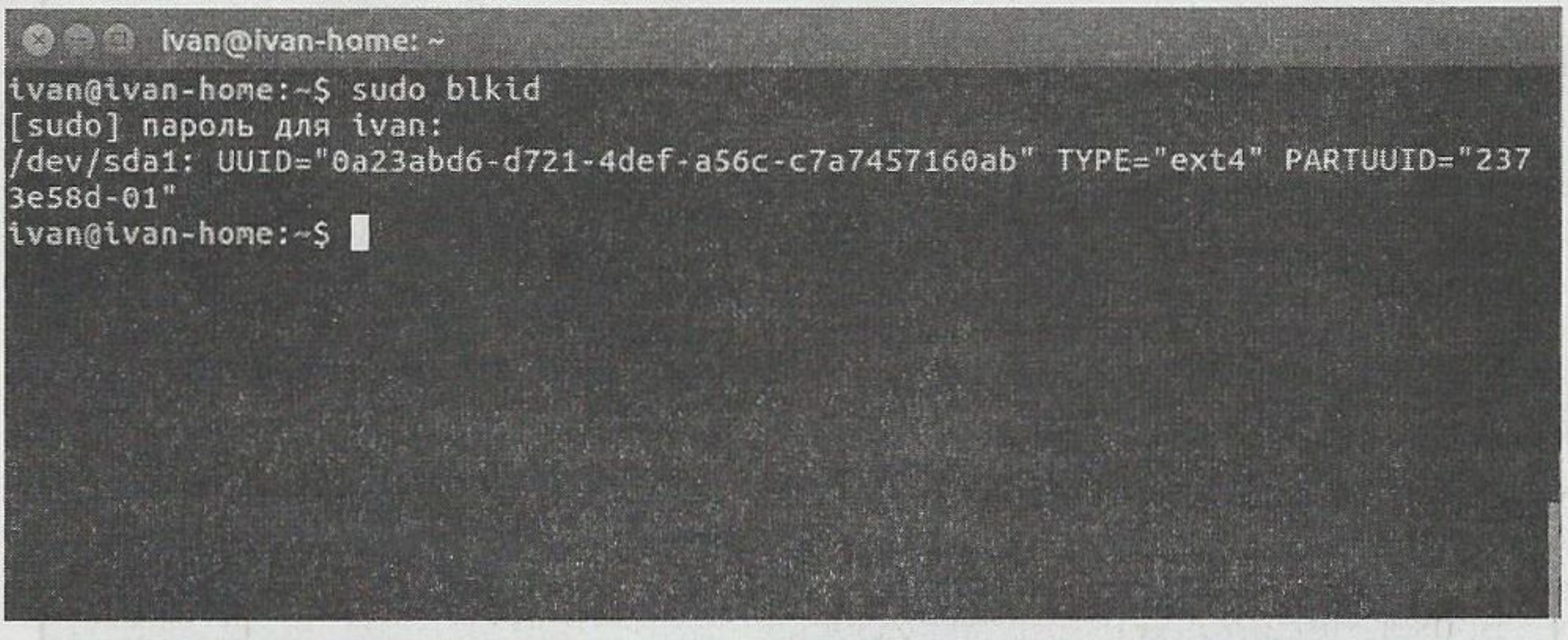


Рис. 4.1. Результат выполнения команды sudo blkid

4.6. Файловый менеджер Nautilus

После рассмотрения столь сложных материй пора приступить к обычной работе с файлами и каталогами. Работать с файлами и каталогами можно или с помощью графического менеджера вроде Nautilus или с помощью команд операционной системы. Команды будут рассмотрены далее, а пока рассмотрим графический файловый менеджер Nautilus.

4.6.1. Элементы окна Nautilus

При запуске файловый менеджер Nautilus отображает содержимое домашнего каталога пользователя (рис. 4.2). В верхней части окна находятся кнопки Назад и Вперед, которые удобно использовать при навигации по файловой системе, кнопка, отображающая название текущей папки, кнопка поиска файлов, кнопка сортировки и управления отображением, кнопка (последняя на панели) дополнительных операций.

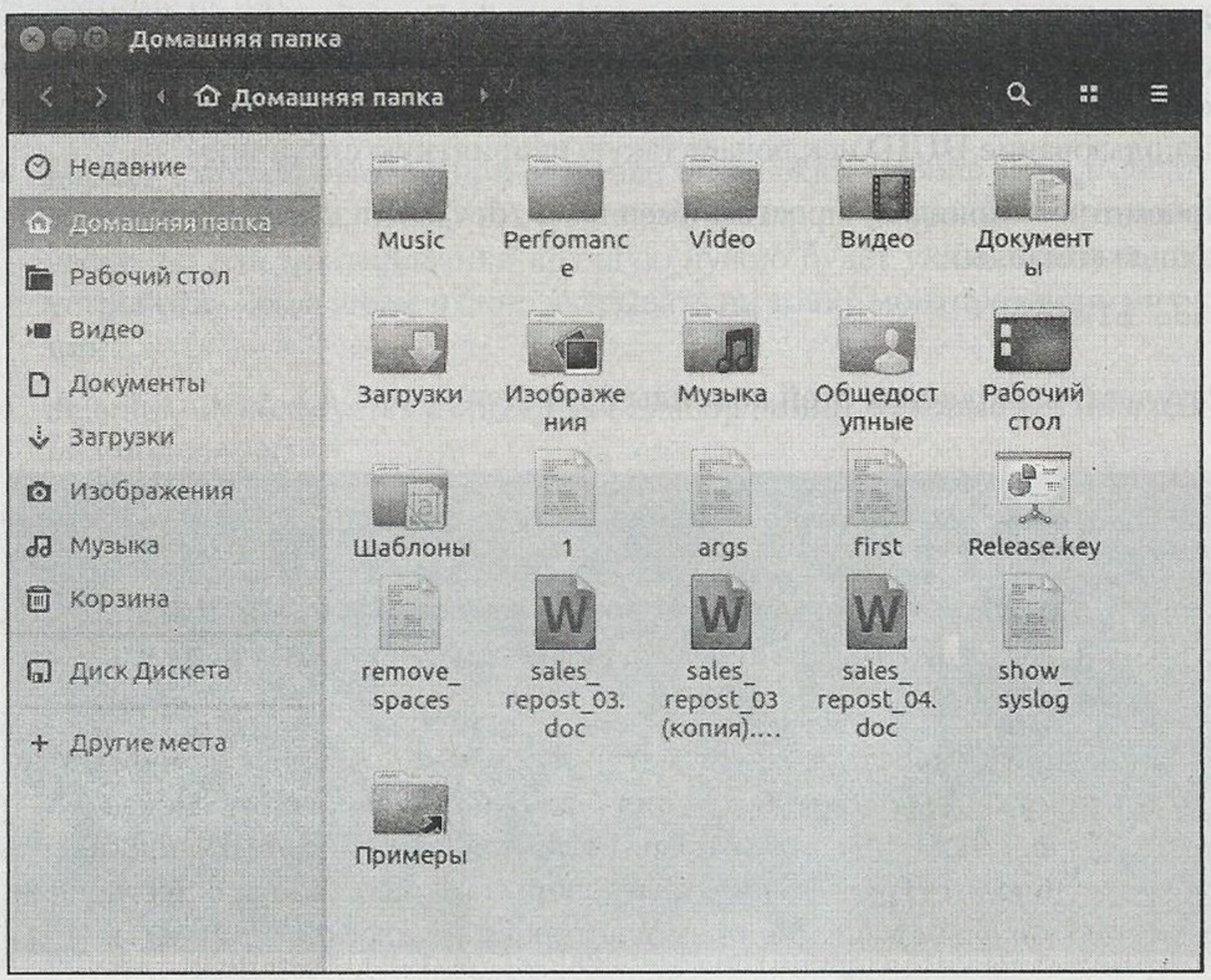


Рис. 4.2. Основное окно Nautilus

Слева находится боковая панель навигации по файловой системе. Она содержит кнопки быстрого доступа к часто используемым местам файловой системы. Кроме содержимого файловой системы Nautilus поддерживает и отображение общих ресурсов сети Windows. Для этого нужно перейти в раздел **Другие места** - в нем вы найдете список компьютеров сети Windows.

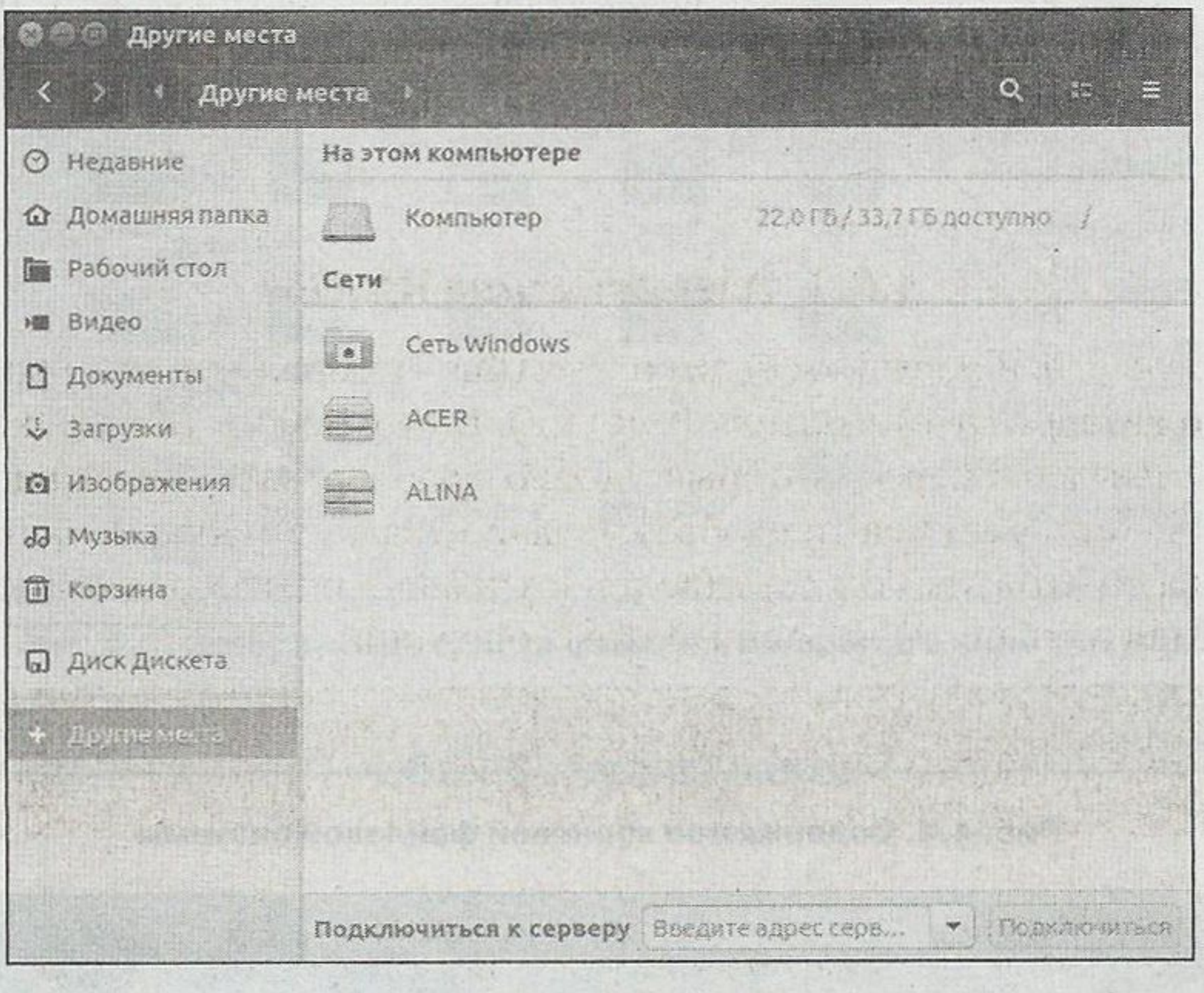


Рис. 4.3. Сеть Windows

Элемент **Компьютер** (см. рис. 4.3) позволяет получить доступ к корневой файловой системе. При двойном щелчке на нем откроется содержимое корневой файловой системы (рис. 4.4).

Кнопка управления отображением и сортировкой позволяет изменить способ сортировки, способ отображения, включить и выключить показ скрытых файлов. Также в меню, открываемой при нажатии этой кнопки, есть команда **Обновить** (рис. 4.5). Обратите внимание, что в Linux скрытым считается любой файл или каталог, имя которого начинается с точки (.).

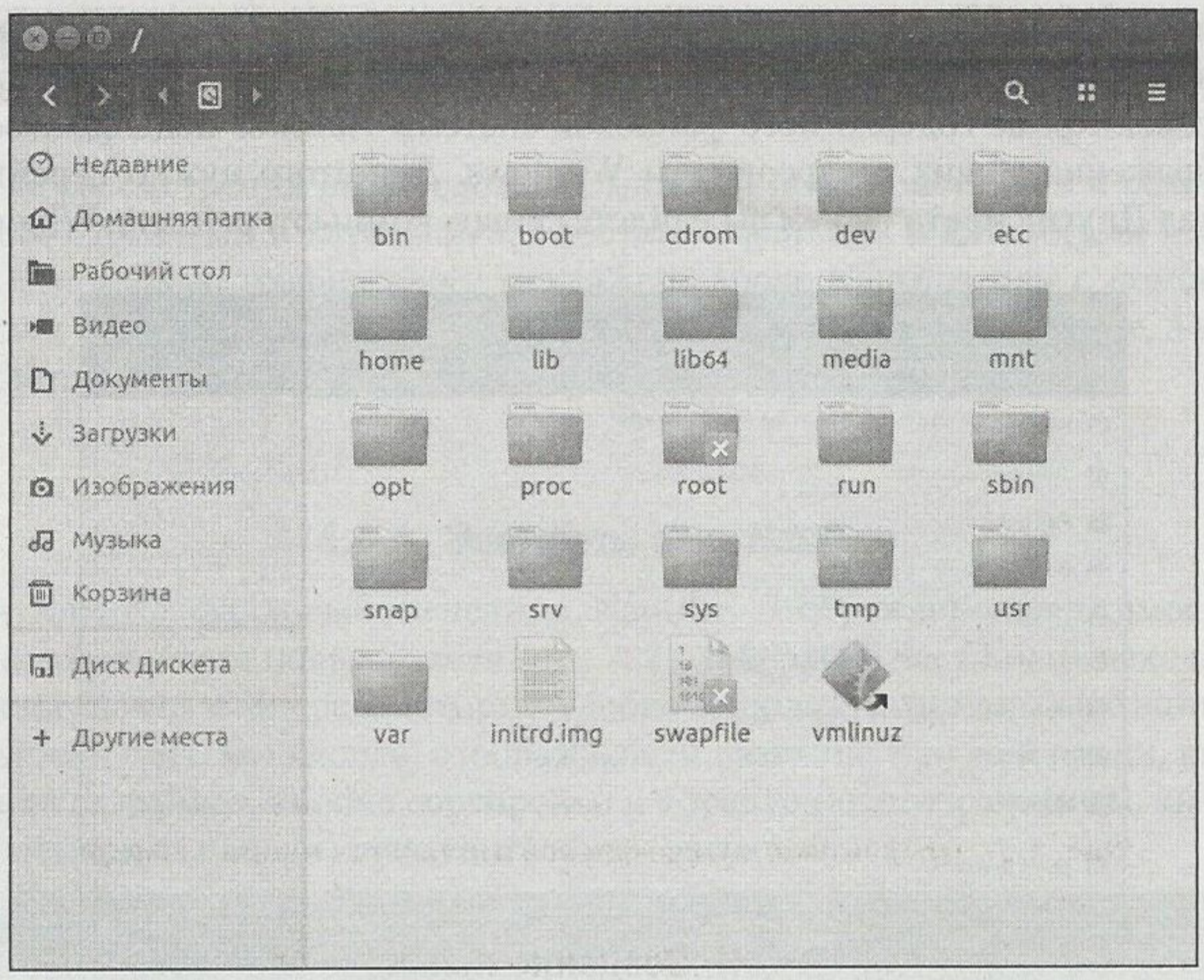


Рис. 4.4. Содержимое корневой файловой системы

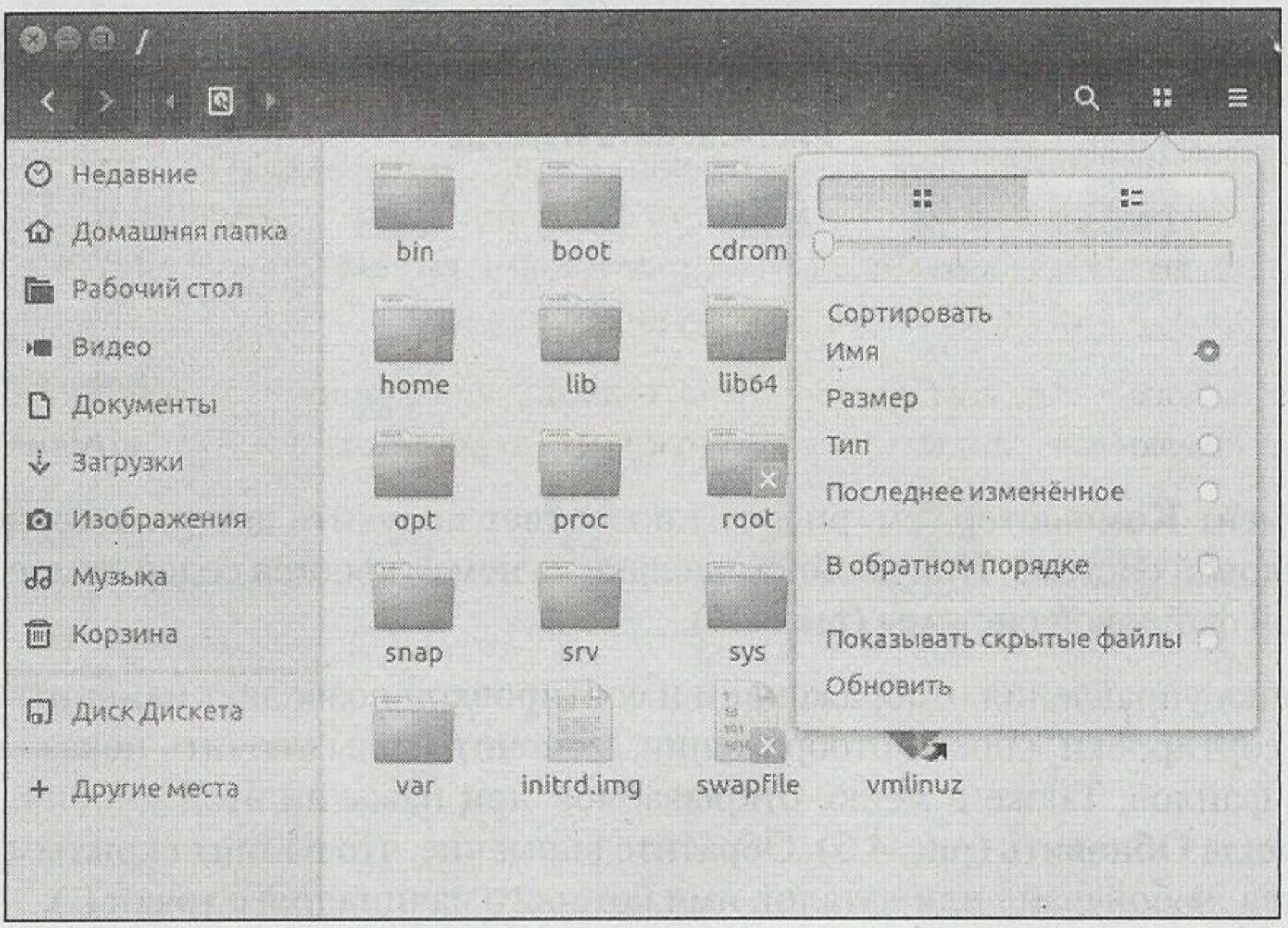


Рис. 4.5. Кнопка управления отображением и сортировкой

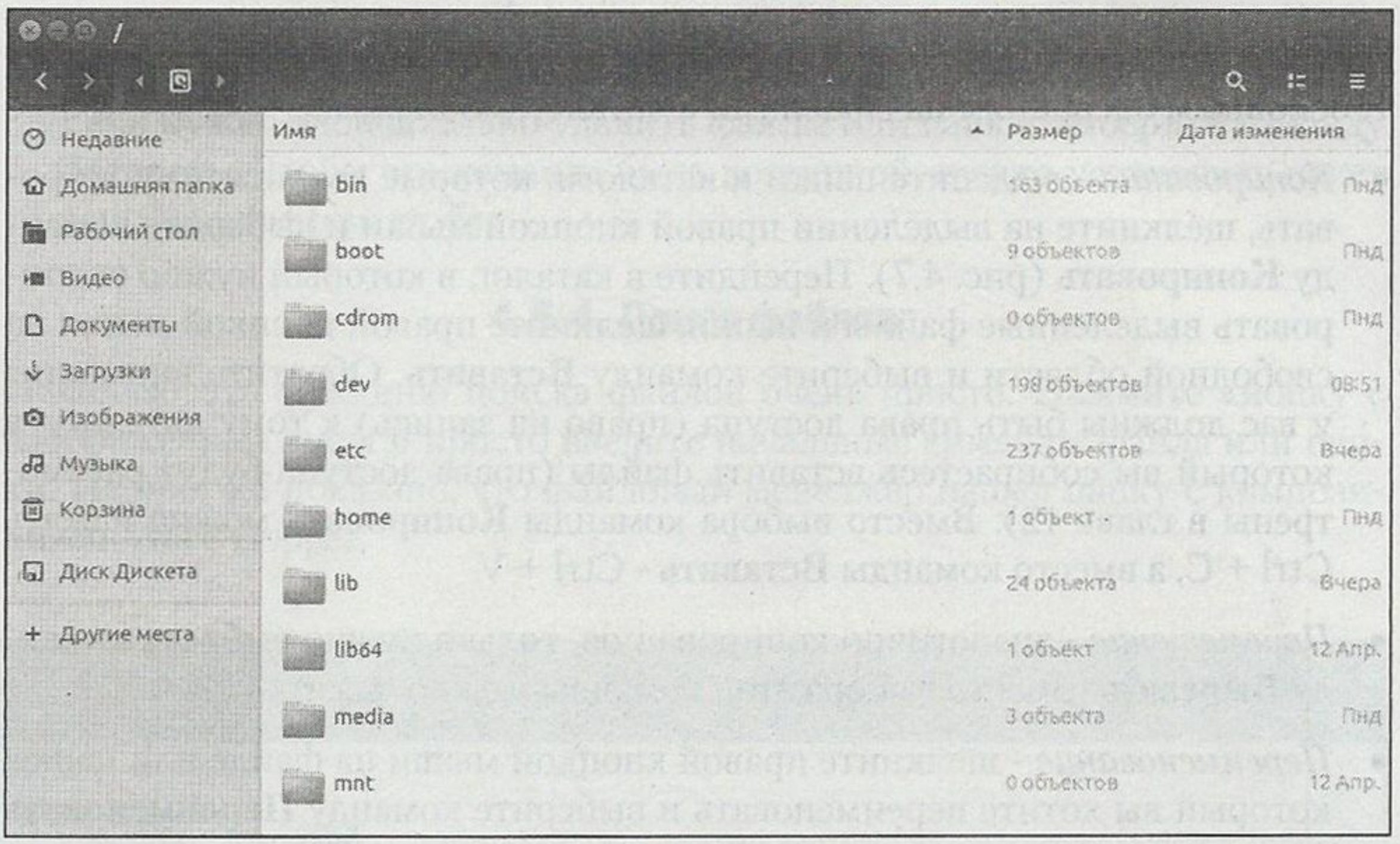


Рис. 4.6. Отображение списка файлов и каталогов в виде таблицы

4.6.2. Создание папки

Для создания папки нажмите кнопку дополнительных действий (последняя кнопка на панели Nautilus) и выберите команду **Создать папку**.

Обратите внимание: для создания папки у вас должны быть соответствующие права доступа. Как правило, при создании папок в домашнем каталоге проблем не будет, а вот создать папку в корневой файловой системе у вас без прав суперпользователя не получится.

Если же все-таки, вам нужно создать папку в корневой файловой системе или в каком-то ее подкаталоге, то откройте терминал и введите команду:

```
sudo mkdir <имя_папки>
```

Например:

```
sudo mkdir /opt
```

4.6.3. Основные операции над файлами и папками

К основным операциям над файлами и каталогами относят:

- *Копирование* - выделите папки и каталоги, которые вы хотите скопировать, щелкните на выделении правой кнопкой мыши и выберите команду **Копировать** (рис. 4.7). Перейдите в каталог, в который нужно скопировать выделенные файлы и папки, щелкните правой кнопкой мыши на свободной области и выберите команду **Вставить**. Обратите внимание: у вас должны быть права доступа (право на запись) к тому каталогу, в который вы собираетесь вставить файлы (права доступа будут рассмотрены в главе 12). Вместо выбора команды **Копировать** можно нажать Ctrl + C, а вместо команды **Вставить** - Ctrl + V.
- *Перемещение* - аналогично копированию, только нужно выбрать команду **Вырезать**. Вместо выбора этой команды можно нажать Ctrl + X.
- *Переименование* - щелкните правой кнопкой мыши на файле или папке, который вы хотите переименовать и выберите команду **Переименовать** (можно просто щелкнуть на файле и нажать кнопку F2). Введите новое имя.
- *Удаление* - выделите папки и каталоги, которые вы хотите удалить, щелкните правой кнопкой мыши на выделении и выберите команду **Переме-**

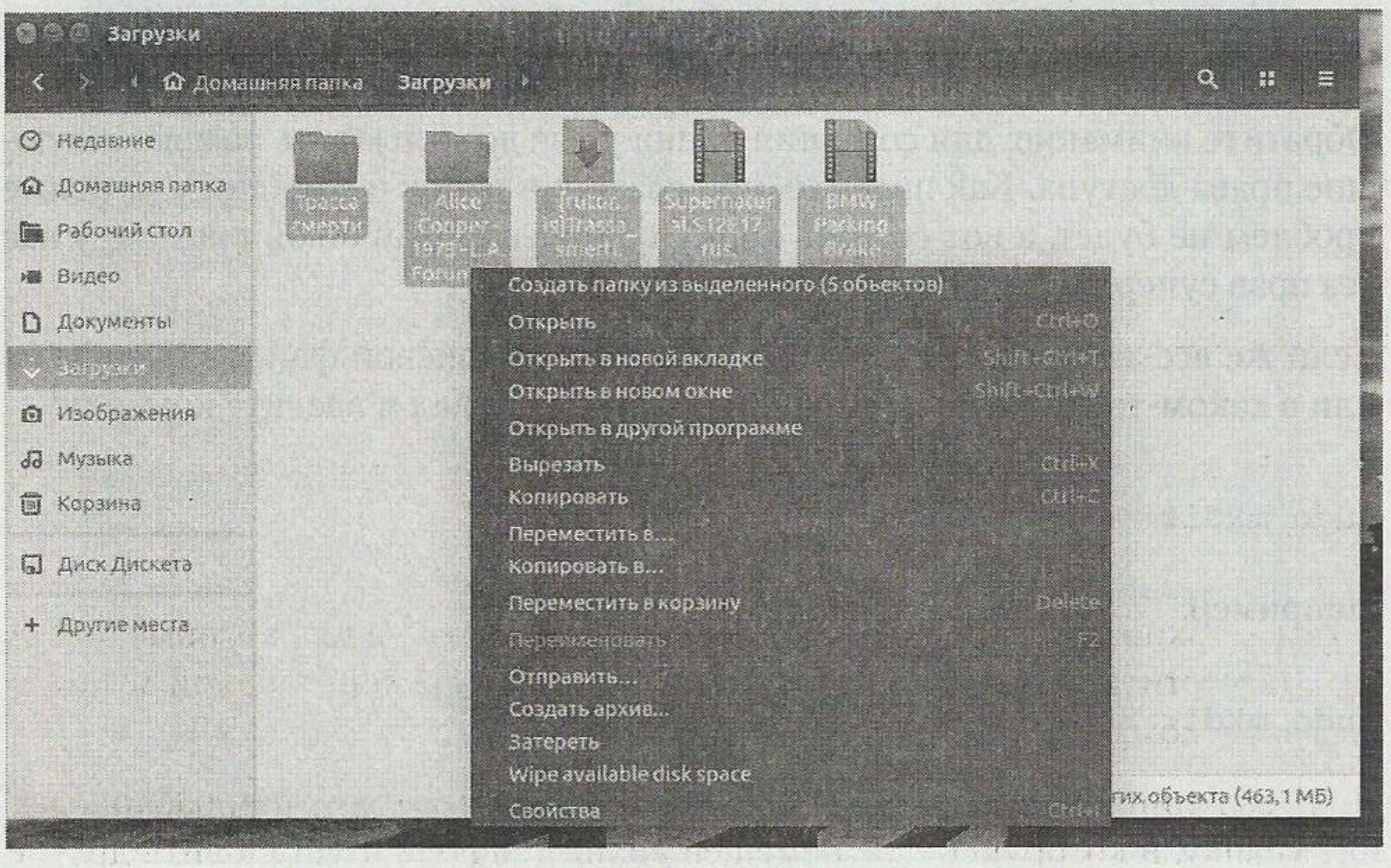


Рис. 4.7. Операции над файлами и каталогами

стить в корзину. Выделенные файлы и папки будут перемещены в Корзину, доступ к которой есть на боковой панели файлового менеджера. Если нужно окончательно удалить файлы или папки, выберите команду **Затереть**. Чтобы эта команда была доступной, нужно установить пакет nautilus-wipe (см. гл. 5).

4.6.4. Поиск файлов

Использовать функцию поиска файлов очень просто. Нажмите кнопку с изображением лупы и просто введите начальные символы файла или папки. На рис. 4.8 показано, что файловый менеджер нашел папку с композициями Alice Copper.

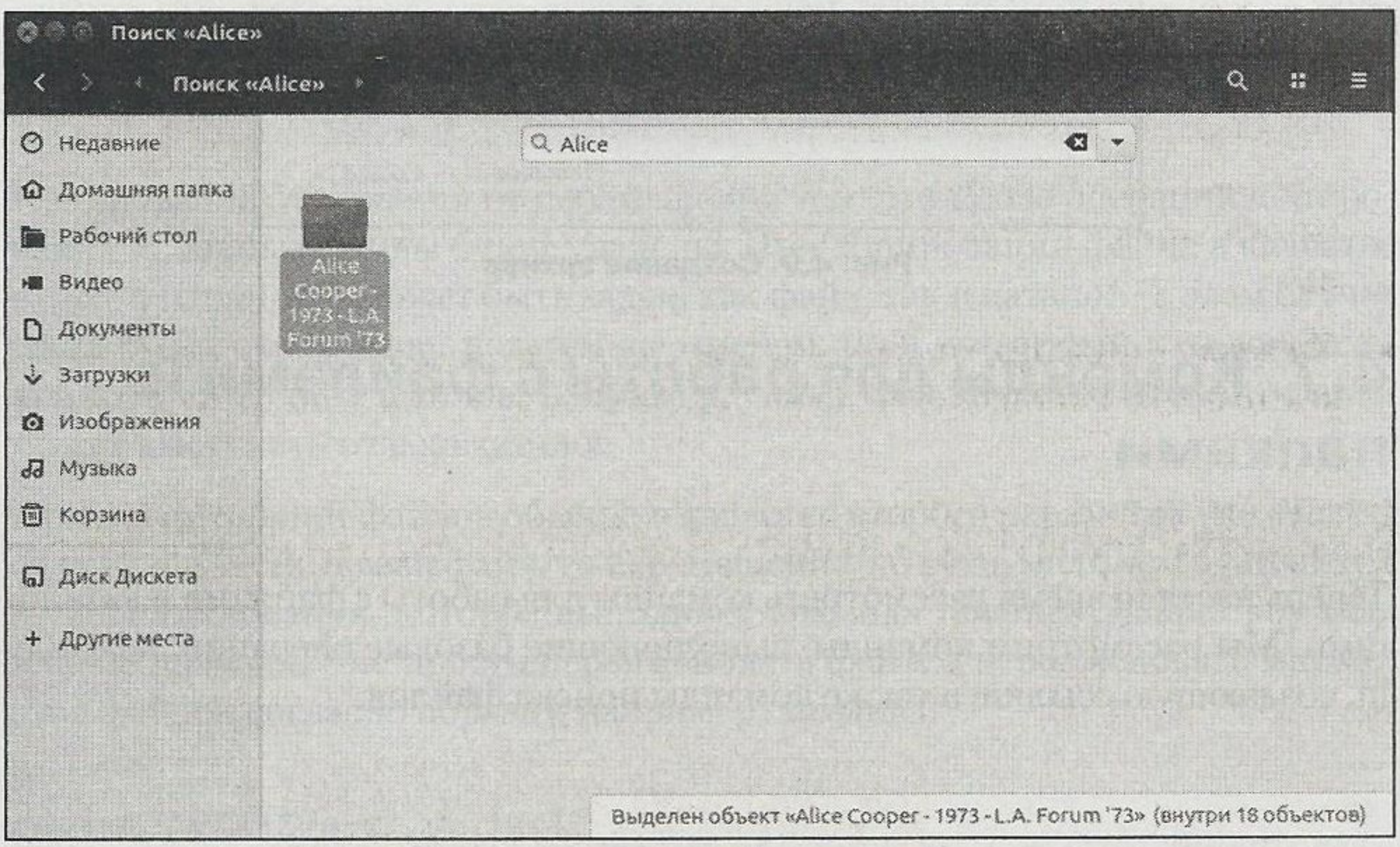


Рис. 4.8. Поиск в действии

4.6.5. Создание архивов

Создать архив достаточно просто. Выделите файлы и папки, которые вы хотите поместить в архив, щелкните на выделении правой кнопкой мыши и выберите команду **Создать архив**.

При создании архива (см. рис. 4.9) можно выбрать его формат, расположение (папку, в которую будет помещен архив), пароль и установить другие опции.

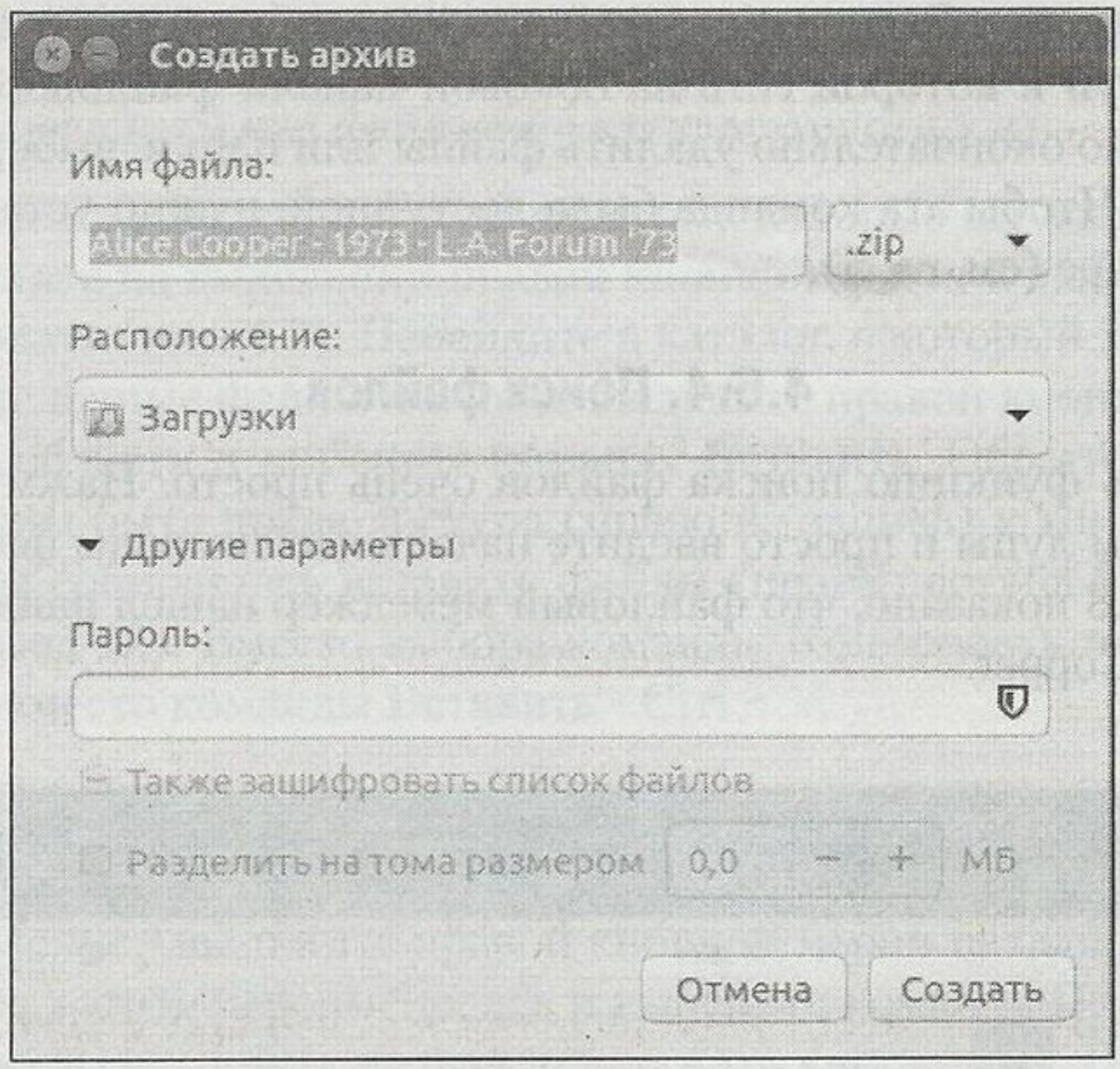


Рис. 4.9. Создание архива

4.7. Команды для работы с файлами и папками

Теперь настало время рассмотреть команды для работы с файлами и каталогами. Мы рассмотрим команды, выполняющие базовые операции, команду ln, создающую ссылки, а также команды поиска файлов.

4.7.1. Команды для работы с файлами

Начнем с команды touch. Эта команда изменяет время модификации файла, а если файл не существует, она его создает, поэтому ее удобно использовать для создания пустого файла, например:

```
touch file.txt
```

Команды cat и tac используются для вывода на консоль содержимого файла. Команда cat выводит файл в прямом порядке, tac - в обратном. Обычно эти команды используются в паре с командой less или grep - текстовые

файлы могут содержать очень много информации, которая не поместится на экране. Команда less организует удобный просмотр такой информации, а команда grep представляет собой фильтр, например:

```
cat file.txt | less
cat file.txt | grep Linux
```

Первая команда просто выведет на экран содержимое file.txt, сделав это в удобном для пользователя порядке. А вторая команда выведет только те строки, которые содержат Linux.

Команды cp и mv используются, соответственно, для копирования и перемещения файлов:

```
cp <откуда> <куда>
mv <откуда> <куда>
```

Копирование отличается от перемещения тем, что после операции копирования исходные файлы будут удалены. При копировании файла в качестве второго параметра может быть задан, как файл, так и каталог - в этом случае файл будет скопирован под тем же именем, но в указанный каталог. Если копируется группа файлов, например, *.doc, то в качестве второго файла может выступать только каталог.

При перемещении файлов обычно в качестве второго параметра указывается каталог. Если указан файл, то перемещение, по сути, мало чем отличается от переименования, которое выполняет команда rename. Однако команда rename позволяет не просто переименовать файл, а переименовать множество файлов согласно образцу, например, команда:

```
rename 'y/A-Z/a-z' *
```

позволит переименовать все файлы так, что все символы в имени файла в верхнем регистре будут заменены на нижний регистр (команда работает только для латиницы). Рассмотрим еще один пример:

```
rename 's/\.html$/\.php/' *.html
```

В данном случае все *.html файлы будут переименованы в .php (то есть будет изменено расширение). Смена расширения с 'html' на 'php' обеспечивается регулярным выражением (первый параметр), а второй параметр задает файлы, к которым будет применено переименование.

Дополнительную информацию вы сможете получить на странице руководства (man rename), а мы пока вернемся к копированию и перемещению. Рассмотрим несколько примеров:

```
cp file.txt file2.txt
cp *.txt docs
```

Первая команда копирует file.txt в file2.txt. Вторая команда копирует все файлы с расширением .txt в каталог docs.

Еще раз прошу обратить внимание: если у вас не будет прав на копирование файла, то вы получите соответствующее сообщение системы. На рис. 4.10 показано, что обычный пользователь пытался копировать файл file.txt в каталог /var. У него ничего не вышло - система отказала в доступе. Если же файл действительно нужно скопировать в каталог /var, то нужно использовать команду sudo cp file.txt /var/

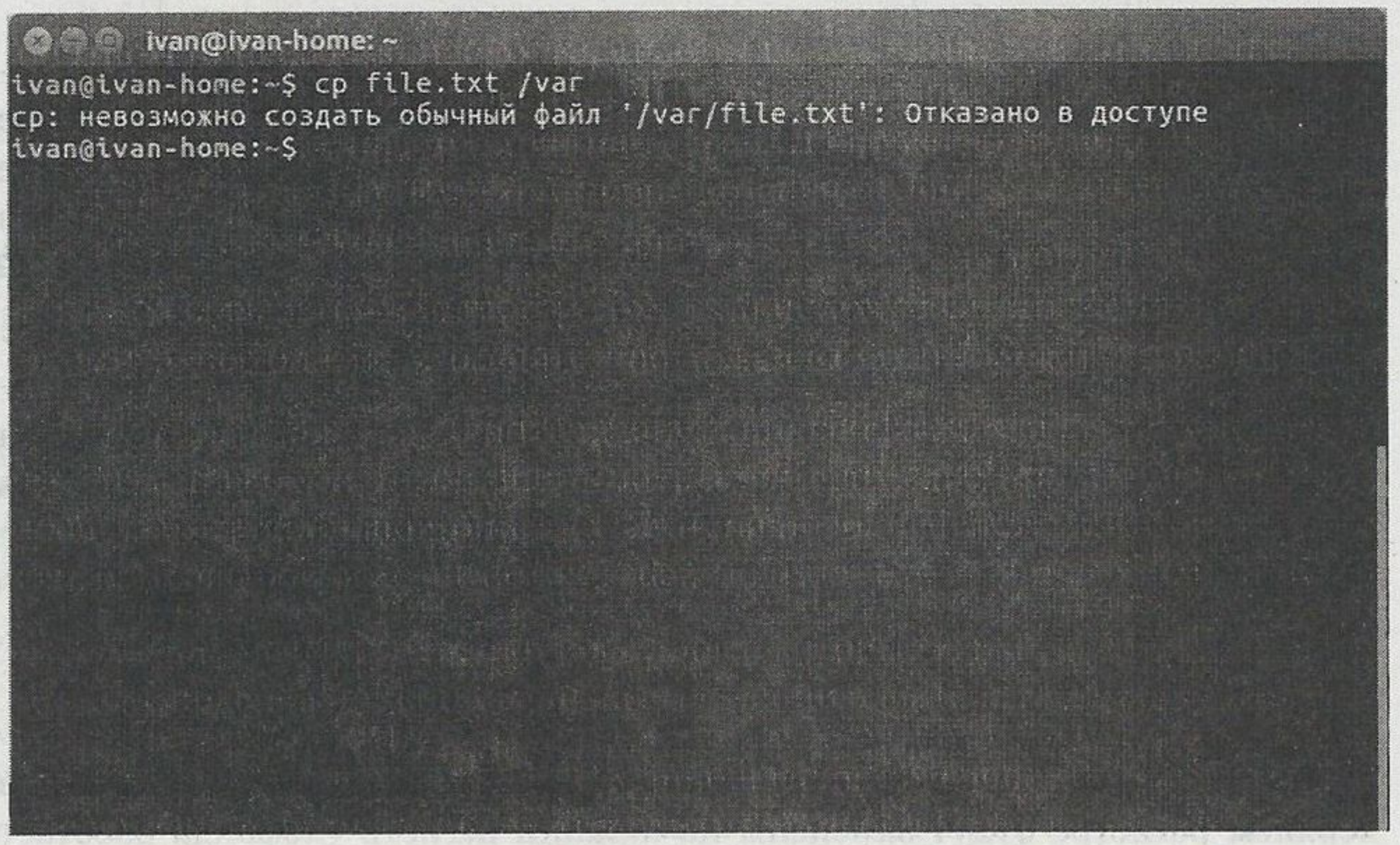


Рис. 4.10. Отказано в доступе

Вместо имени файла иногда очень удобно указать маску имени файла, как уже было показано. Например, можно удалить все резервные копии, созданные текстовым редактором: rm *.bak.

Если же требуется удалить все файлы в текущем каталоге, можно просто указать звездочку: rm *.

Аналогично можно использовать символ ?, который, в отличие от звездочки, заменяющей последовательность символов произвольной длины, заменяет всего один символ. Например, нам нужно удалить все файлы, имена которых состоят из трех букв и начинаются на a:

```
rm a??
```

4.7.2. Команды для работы с каталогами

Для создания каталога используется команда mkdir, а для изменения текущего каталога - команда cd. Следующие две команды создают каталог Music и изменяют текущий каталог:

```
mkdir Music
cd Music
```

Первая команда в текущем каталоге создает каталог Music, а вторая «заходит» в него. Текущий каталог отображается в приглашении командной строки.

Примечание. Linux чувствительна к регистру имен файлов и каталогов, поэтому каталоги Music и music - два абсолютно разных каталога.

Команда ls выводит содержимое каталога. У команды ls очень много разных ключей, с которыми вы можете ознакомиться в справочной системе. Как по мне, самый полезный из них - «-l», обеспечивающий вывод в так называемом «длинном» формате, когда кроме имени файла выводится и другая полезная информация о нем, а именно: права доступа, число ссылок (сразу после прав доступа), пользователь-владелец, группа-владелец, размер файла, дата последнего изменения и имя файла. Обратите внимание на права доступа: если первая буква в наборе прав доступа - d (directory), то перед нами каталог, а не файл. Впрочем, каталоги и так отмечаются синим цветом в выводе. Зеленым цветом отмечены файлы, имеющие право запуска (исполнимые файлы - программы и сценарии).

Если указать опцию -a (опции можно комбинировать, поэтому смело указывайте набор опций, например, -la), то будут отображены скрытые файлы и каталоги - имя которых начинается с точки (рис. 4.12). Имя каталога «.» означает текущий каталог, а имя «..» - родительский.

```
ivan@ivan-home: ~
ivan@ivan-home:~$ ls -l
итого 376
-rw-r--r-- 1 ivan ivan    153 Апр 18 17:42 1
-rw-r--r-- 1 ivan ivan      0 Апр 20 13:16 1.txt
-rwxr-xr-x 1 ivan ivan     46 Апр 19 20:15 args
-rw-r--r-- 1 ivan ivan   8980 Апр 17 22:42 examples.desktop
-rw-r--r-- 1 ivan ivan      0 Апр 20 13:23 file.txt
-rwxr-xr-x 1 ivan ivan     78 Апр 19 20:07 first
drwxr-xr-x 2 ivan ivan   4096 Апр 19 21:03 Music
drwxr-xr-x 2 ivan ivan   4096 Апр 19 11:29 Perfomance
-rw-r--r-- 1 ivan ivan   1102 Мар  2 21:13 Release.key
-rwxr-xr-x 1 ivan ivan    865 Апр 19 20:57 remove_spaces
-rw-r--r-- 1 ivan ivan 100699 Апр 18 09:30 sales_repost_03.doc
-rw-r--r-- 1 ivan ivan 100699 Апр 18 09:30 sales_repost_03 (копия).doc
-rw-r--r-- 1 ivan ivan 100699 Апр 18 09:40 sales_repost_04.doc
-rwxr-xr-x 1 ivan ivan     90 Апр 19 21:10 show_syslog
drwxr-xr-x 2 ivan ivan   4096 Апр 19 10:19 Video
drwxr-xr-x 2 ivan ivan   4096 Апр 17 22:50 Видео
drwxr-xr-x 2 ivan ivan   4096 Апр 17 22:50 Документы
drwxr-xr-x 4 ivan ivan   4096 Апр 19 10:19 Загрузки
drwxr-xr-x 2 ivan ivan   4096 Апр 17 22:50 Изображения
drwxr-xr-x 3 ivan ivan   4096 Апр 20 08:56 Музыка
drwxr-xr-x 2 ivan ivan   4096 Апр 17 22:50 Рабочий стол
drwxr-xr-x 2 ivan ivan   4096 Апр 17 22:50 Шаблоны
ivan@ivan-home:~$
```

Рис. 4.11. Вывод содержимого каталога

Рис. 4.12. Отображение скрытых файлов и каталогов.

По умолчанию ls сортирует список файлов в алфавитном порядке, причем сначала выводятся файлы, а потом - каталоги. Можно отсортировать список

по времени модификации, для этого используйте опцию -t. Дополнительные опции вы найдете в справочной системе. Если она у вас на английском, можете воспользоваться следующей ссылкой, в которой приводится перевод страницы руководства на русский язык http://rus-linux.net/MyLDP/consol/hdrguide/rusman/ls.htm.

Примечание. В Linux есть три специальных имени каталога: «.» - текущий каталог, «..» - родительский каталог, «~» - домашний каталог пользователя.

Команда rmdir используется для удаления непустого каталога. Если вам нужно удалить каталог вместе со всеми подкаталогами и файлами, тогда укажите опцию рекурсивного удаления команды rm:

```
rm -r Music
```

4.7.3. Команда ln

В Linux возможно, чтобы один и тот же файл существовал в системе под разными именами. Для этого используются ссылки. Ссылки бывают двух типов: жесткие и символические. Жесткие ссылки жестко привязываются к файлу — вы не можете удалить файл, пока на него указывает хотя бы одна жесткая ссылка. А вот если на файл указывают символические ссылки, его удалению ничто не помешает.

Жесткие ссылки не могут указывать на файл, который находится за пределами файловой системы. Предположим, у вас два Linux-раздела: один корневой, а второй используется для веб-сервера и монтируется к каталогу /var корневой файловой системы. Так вот, вы не можете создать в корневой файловой системе жесткую ссылку, которая ссылается на файл в файловой системе, подмонтированной к каталогу /var. Это очень важная особенность жестких ссылок. Если вам нужно создать ссылку на файл, который находится за пределами файловой системы, вам следует использовать символические ссылки.

Для создания ссылок используется команда ln:

```
ln some_file_with_long_name.doc short.doc
ln -s some_file_with_long_name.doc short_ss.doc
```

Первая команда создает жесткую ссылку short.doc, ссылающуюся на текстовый файл some_file_with_long_name.doc. Вторая команда создает символическую ссылку short_ss.doc, которая ссылается на этот же текстовый файл some_file_with_long_name.doc. В графическом файловом менеджере символические ссылки помечаются значком-стрелкой, а жесткие ссылки никак не отмечаются.

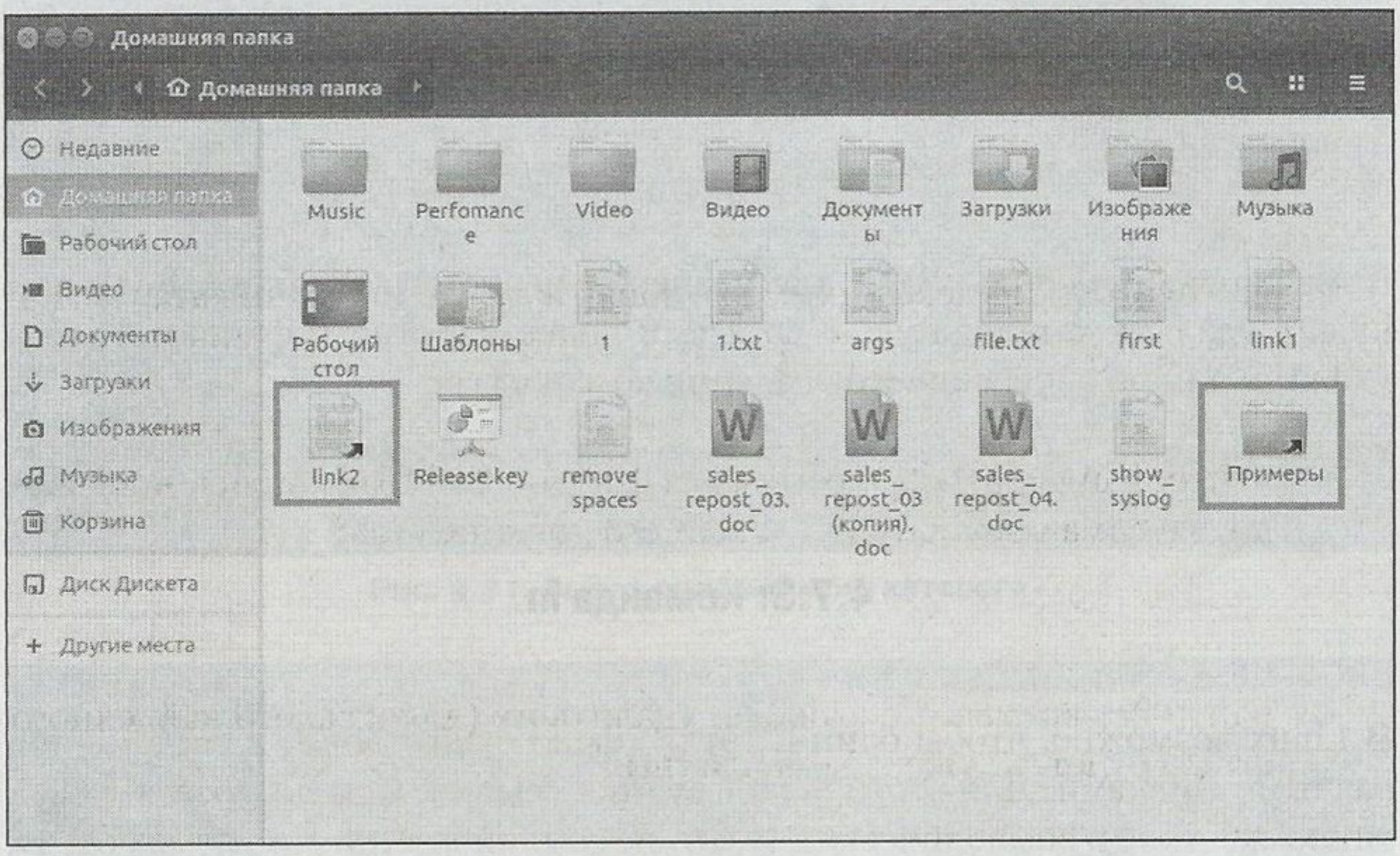


Рис. 4.13. Символьные ссылки в графическом файловом менеджере

4.7.4. Поиск файлов в консоли

Для поиска файлов в Linux предназначена программа find - мощная утилита со сложным синтаксисом, и далеко не всегда она нужна обычному пользователю. Синтаксис команды следующий:

```
find список_поиска выражение
```

Мощность команды find заключается во множестве самых разных параметров поиска, которые из-за их количества не так-то просто запомнить. К тому же find может выполнять операции над найденными файлами. Например, вы можете найти временные файлы и сразу удалить их.

Подробно опции команды find мы изучать не станем — это вы можете сделать самостоятельно с помощью команды man find. Здесь мы рассмотрим лишь несколько примеров использования этой команды:

- найти файлы с именем report.txt, поиск начать с корневого каталога (/):

```
find / -name report.txt
```

- найти файлы по маске *.doc:

```
find / -name '*.doc'
```

- найти файлы нулевого размера, поиск начать с текущего каталога (.):

```
find . -size 0c
```

- найти файлы, размер которых от 100 до 500 Мбайт, поиск производить в домашнем каталоге и всех его подкаталогах:

```
find ~ -size +100M -size -500M
```

- найти все временные файлы и удалить их (для каждого найденного файла будет запущена команда rm):

```
sudo find / -name *.tmp -ok rm {} \;
```

Вместо параметра -ok можно использовать параметр -exec, который также запускает указанную после него команду, но не запрашивает подтверждение выполнения этой команды для каждого файла.

Кроме команды find можно использовать команды which и locate. Первая выводит полный путь к программе или к сценарию, если программа или сценарий находится в списке каталогов, заданном в переменной окружения PATH:

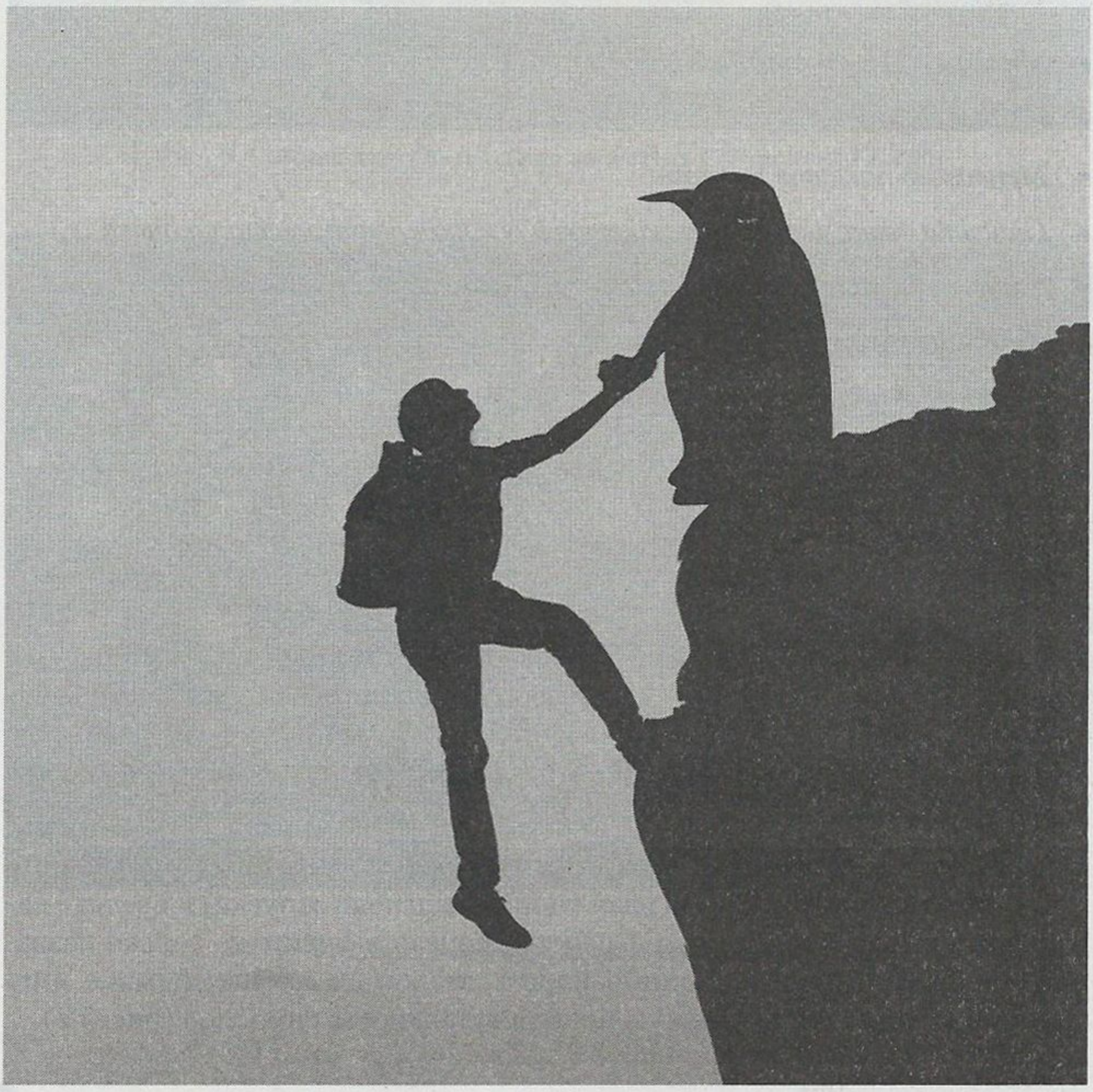
```
which php
```

Команда locate ищет в базе данных демона located файлы, соответствующие заданному образцу. Недостаток этой команды в том, что located имеется далеко не во всех дистрибутивах, поэтому команды locate у вас может и не быть. Зато если located имеется и запущен, поиск файлов будет осуществляться быстрее, чем с помощью find.

Примечание. Если вам часто приходится работать в консоли, например, при администрировании сервера VPS/VDS, то установите файловый менеджер mc (sudo apt install mc). Это классический двухпанельный менеджер, который существенно облегчает работу с файлами и каталогами. Кроме базовых операций, он поддерживает функции просмотра, редактирования и поиска файлов, а также обладает встроенным FTP-клиентом.

ГЛАВА 5.

Управление пакетами и процессами



В этой главе:

- *Как запустить установленную программу?*
- *Что такое пакеты?*
- *Установка программ*
- *Менеджеры пакетов dnf и apt*
- *Ubuntu Software*
- *Менеджер пакетов Synaptic*
- *Очистка кэша менеджера пакетов и освобождения места на диске*
- *Управление процессами*
- *Команды ps, top, kill*
- *Изменение приоритета процесса*

5.1. Запуск программ

Существует несколько способов запуска программ в Linux. Самый первый из них - запуск по команде. Откройте терминал и введите команду, запускающую программу. Если не помните точную команду, введите начальные символы команды и нажмите Tab. Автодополнение подскажет точную команду (рис. 5.1).

Второй способ - с помощью средства поиска файлов и программ Ubuntu. Откройте средство поиска (первая кнопка на панели запуска) и введите начальные символы программы. Средство поиска выводит не только названия приложений, но и названия файлов и каталогов в домашней папке, имена которых содержат введенную последовательность символов (рис. 5.2).

```
ivan@ivan-home: ~
ivan@ivan-home:~$ file
file          file2brl      filefrag      file-rename   file-roller   filezilla
ivan@ivan-home:~$ file
file          file2brl      filefrag      file-rename   file-roller   filezilla
ivan@ivan-home:~$ filezilla
```

Рис. 5.1. Пользователь пытался запустить FTP-клиент FileZilla

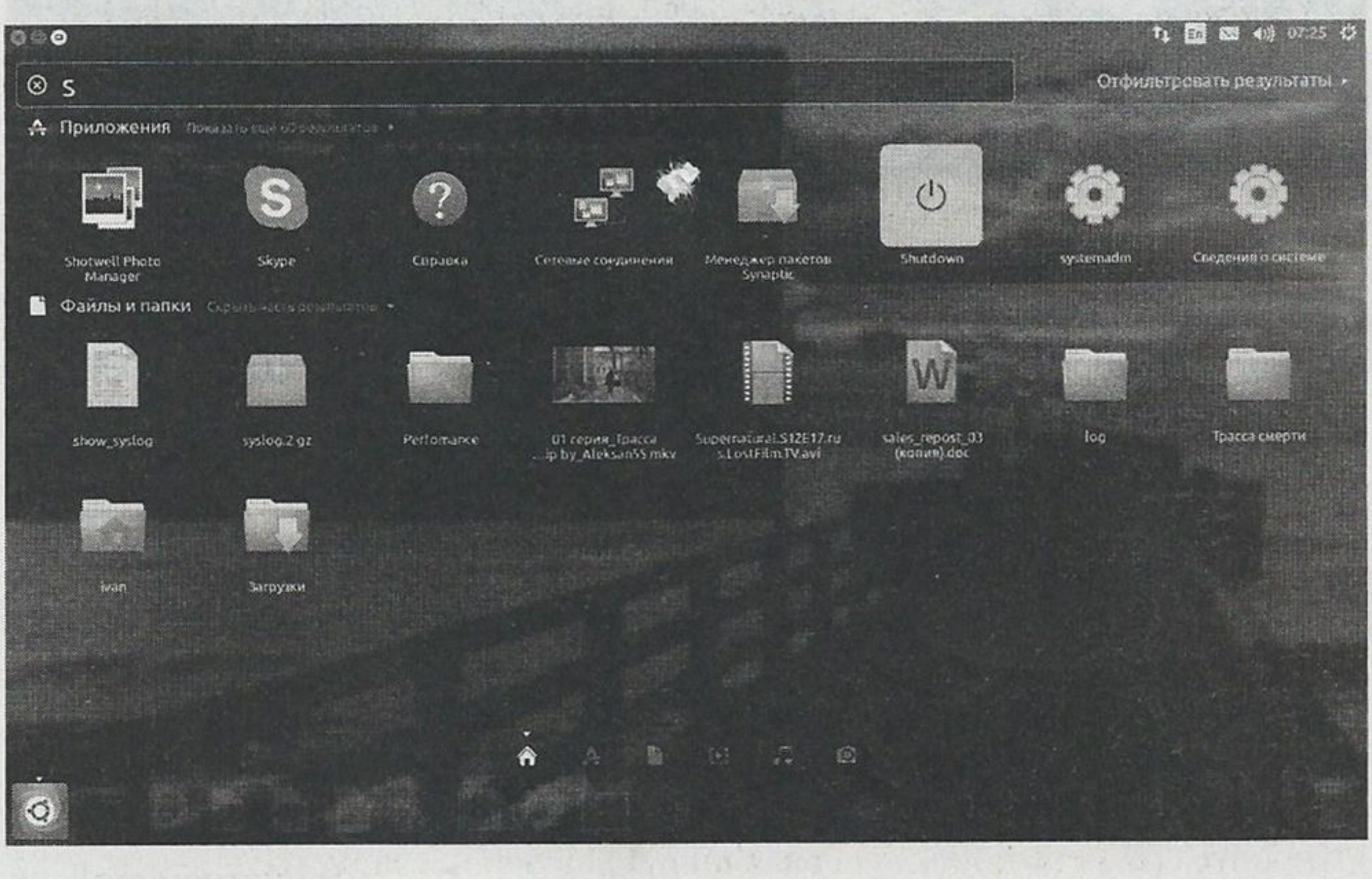


Рис. 5.2. Результат поиска программ

Когда вы запустите программу, вы можете закрепить ее на панели запуска (рис. 5.3). После этого можно будет запускать программу, просто нажав ее кнопку на панели запуска. Удалить программу с панели можно аналогично - щелкните на кнопке правой кнопкой мыши и выберите команду **Изъять из панели**.

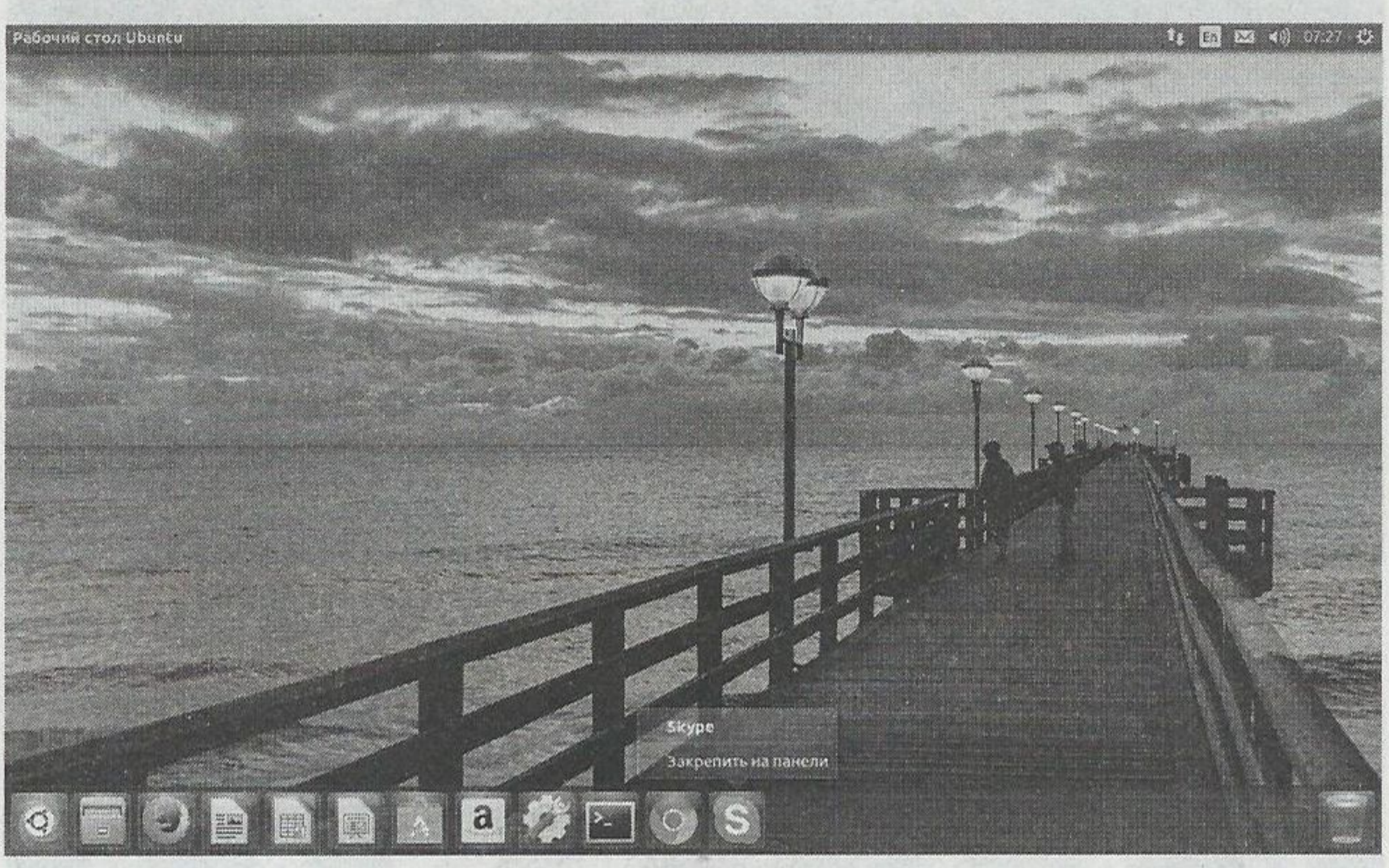


Рис. 5.3. Панель запуска

В других дистрибутивах (где используются KDE или GNOME) будет меню, используя которое можно легко выбрать то или иное приложение (если оно, конечно, установлено). В последних версиях GNOME меню было преобразовано в экран выбора приложений, см. рис. 5.4.

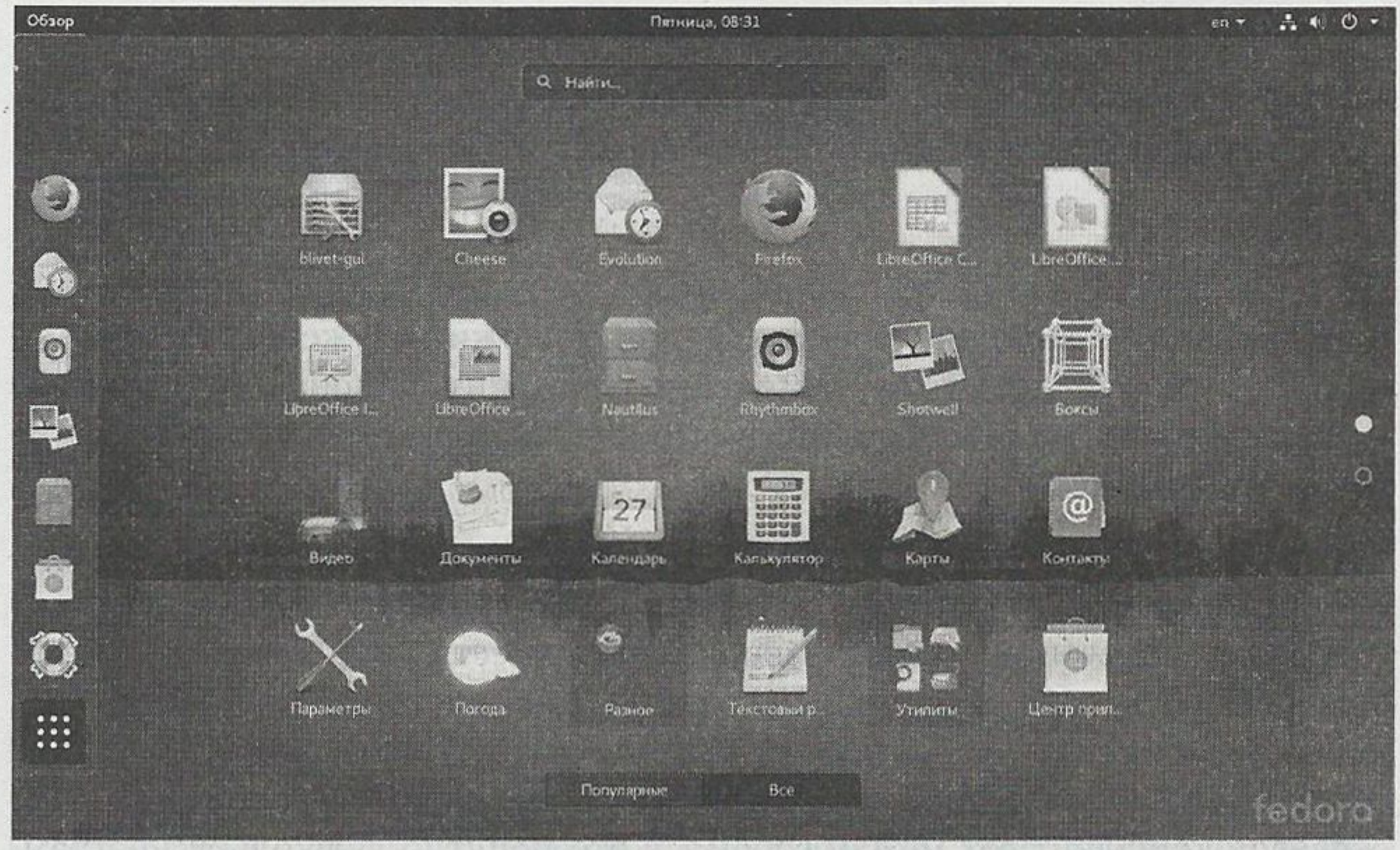


Рис. 5.4. Экран выбора приложений (Fedora 26)

5.2. Управление пакетами

5.2.1. Введение в пакеты

А теперь разберемся, откуда берутся программы. Ведь прежде, чем программу использовать, ее нужно установить. В Windows есть понятие установщика программы - как правило, это файл с названием setup.exe, который выполняет установку программы. В Linux такого нет. Все программы распространяются в пакетах - в файлах с расширением .rpm (в RedHat-совместимых дистрибутивах - Fedora, CentOS) или .deb (в Debian, Ubuntu и ее клонах).

Пакет содержит следующую информацию:

- Собственно, саму программу - один или несколько исполнимых файлов.
- Файлы, необходимые для работы программы - возможно, какие-то файлы данных, файлы конфигурации и т.д.
- Документацию
- Инструкции, которые нужно выполнить при установке и при удалении программы
- Информацию о зависимостях программы

- Информацию о конфликтах программы
- Информацию о разработчике программы, ее описание и прочую информацию, необходимую для поиска программы в репозитарии.

По сути, пакет - это архив специального формата, который кроме самой программы и ее дополнительных файлов содержит служебную информацию. Кстати, Windows тоже постепенно переходит на концепцию пакетов - файлы с расширением .msi - это ничто иное, как пакеты. Но не будем отвлекаться от Linux.

У пакета могут быть зависимости и конфликты. Когда программа требует для своей работы другую программу, это называется зависимостью. Менеджер пакетов разрешает зависимости, то есть устанавливает все необходимые для работы программы пакеты. Также у пакета могут быть конфликты - не всегда две программы могут сосуществовать одновременно на одном компьютере. Например, пакеты различных почтовых серверов, веб-серверов конфликтуют между собой - и это понятно, поскольку программы используют один и тот же порт и система просто не будет знать, какую из этих программ использовать. Менеджер пакетов разрешает и конфликты - при установке пакета, который конфликтует с уже установленным, он предлагает удалить уже установленный пакет. Можно, конечно, пренебречь конфликтом и установить программу на свой страх и риск - тогда не будут работать обе программы.

Пакеты хранятся в репозиториях. Репозитарий - это хранилище пакетов. Оно может быть локальным, например, инсталляционный DVD-диск, так и сетевым - у каждого дистрибутива есть собственный набор репозиториев. Иногда нужно подключить сторонний репозитарий. Не все возможные пакеты по тем или иным причинам (чаще всего это связано с лицензиями) находятся в официальных репозиториях. Поэтому подключаются сторонние репозитарии и оттуда устанавливаются программы. Обычно это делается путем редактирования конфигурационных файлов менеджера пакетов.

Существует два основных менеджера пакетов: apt и dnf. Первый используется в дистрибутивах Debian и Ubuntu, а второй - в Fedora и CentOS. Нужно отметить, что в более ранних версиях Debian/Ubuntu вместо apt использовалась команда apt-get - это тоже самое, просто ее сократили до apt в последних версиях Debian/Ubuntu. Что же касается Fedora и CentOS, то в более ранних версиях вместо dnf использовался менеджер пакетов yum. Имейте это в виду, если придется работать с более старыми версиями Fedora. К счастью, параметры у dnf и yum одинаковые, поэтому отличаться будет только сама команда.

5.2.2. Менеджер пакетов dnf

Синтаксис и параметры

Синтаксис вызова менеджера пакетов dnf следующий:

```
dnf <команда> [пакеты]
```

Параметры (команды) этого менеджера пакетов приведены в таблице 5.1.

Таблица 5.1. Команды dnf

dnf	Описание
check-update	Проверка наличия обновлений всех пакетов, сами обновления не устанавливаются.
check-update <пакет>	Производит проверку наличия обновлений определенного пакета. Обновления не устанавливаются.
groupinstall <группа>	Устанавливает все пакеты из указанной группы
grouplist	Выводит список всех групп пакетов
info <пакет>	Выводит информацию о пакете
install <пакеты>	Устанавливает один или несколько пакетов (пакеты перечисляются через пробел)
list	Отображает список всех пакетов (как установленные, так и доступные для установки)
list <маска>	Выводит список всех пакетов, имена которых соответствуют маске.
search <строка>	Производит поиск пакетов, в описаниях которых есть указанная строка
remove <пакет>	Удаляет пакет
upgrade	Проверяет обновление всех пакетов. Если обновления есть, они будут установлены
upgrade <пакет>	Проверяет обновление определенного пакета. Если обновления есть, они будут установлены.

На рис. 5.5 показан процесс установки пакета mc (команда sudo dnf install mc) в дистрибутиве Fedora. Обратите внимание, что для установки пакетов нужны права root, поэтому нужно добавлять команду sudo или выполнять сначала команду su, вводить пароль root, а потом - уже вводить команду dnf.

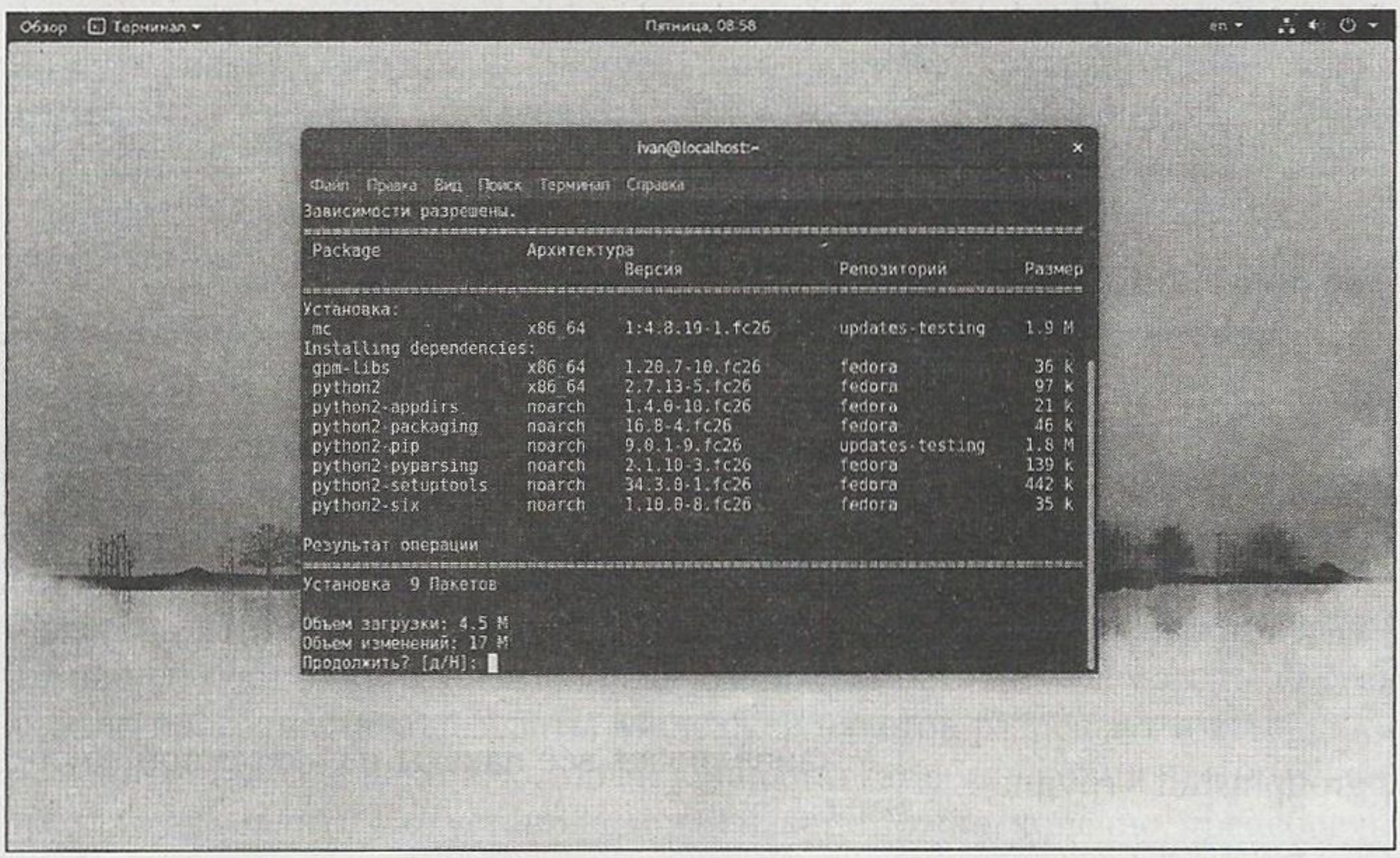


Рис. 5.5. Установка пакета в Fedora 26

Конфигурационные файлы dnf

Основной конфигурационный файл dnf называется /etc/dnf/dnf.conf. Однако репозитарии, как для менеджера yum, так и для менеджера dnf, описываются в каталоге /etc/yum.repos.d/ - это сделано для совместимости.

Как правило, вам не нужно редактировать файл конфигурации. Исключение может составить случай, когда доступ к Интернету осуществляется через прокси-сервер и в этом файле можно указать его параметры:

```
# Адрес прокси-сервера и его порт
proxy=http://proxy.firma.com:3128
# Имя пользователя и его пароль
proxy_username=admin
proxy_password=secret
```

Репозитарии описываются в файле с расширением .repo в каталоге /etc/yum/repos.d. В одном таком файле может быть описано несколько репози-

тариев. В листинге 5.1 приведено описание одного репозитария из файла fedora.repo. Всего в этом файле описаны три репозитария. Репозитарии описываются однотипно, поэтому нет смысла приводить листинг всего файла.

Листинг 5.1. Пример описания репозитария

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
baseurl=http://download.fedoraproject.org/pub/fedora/linux/
releases/$releasever/Everything/$basearch/os/
metalink=https://mirrors.fedoraproject.org/
metalink?repo=fedora-$releasever&arch=$basearch
enabled=1
#metadata_expire=7d
repo_gpgcheck=0
type=rpm
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-
$basearch
skip_if_unavailable=False
```

Параметр name задает название репозитария. Параметр baseurl - путь к файлам репозитария. Как правило, это URL, указывающий веб-сервер, на котором размещены файлы репозитария.

Если параметр enabled установлен в 1, репозитарий считается активным. Вместо удаления репозитария лучше установите параметр enabled в 0, тогда yum будет игнорировать его.

Можно выключить параметр gpgcheck, если возникли проблемы с GPG-ключом репозитария и он не проходит проверку, а установить пакеты все же нужно. Хотя такая ситуация должна вас насторожить.

Очистка кэша

Кэш пакетов, загружаемых из удаленного репозитария, хранится в каталоге /var/cache/dnf. За время работы системы здесь накапливается приличный объем информации, поэтому время от времени кэш пакетов нужно очищать, то есть удалять пакеты, хранящиеся в этом каталоге. Для очистки кэша используется команда clean:

```
dnf clean all
```

Команду clean рекомендуется периодически вводить для очистки кэша пакетов. После установки пакета его файл (с расширением rpm) продолжает находиться в кэше, следовательно, занимает дисковое пространство. Периодически нужно очищать кэш для удаления находящихся в нем RPM-файлов. Особенно процедура очистки кэша рекомендуется тем, что активно устанавливает различное программное обеспечение и у кого жесткий диск небольшого размера.

5.2.3. Менеджер пакетов apt

Синтаксис и команды

Формат вызова этого менеджера пакетов следующий:

```
sudo apt [опции] команды [пакет]
sudo apt-get [опции] команды [пакет]
```

Первая команда используется в более новых дистрибутивах, вторая - как в новых, так и в старых. В старых дистрибутивах первая команда работать не будет. Напомню, что данный менеджер пакетов используется в Debian-совместимых дистрибутивах (Debian, Ubuntu, Kubuntu и прочите клоны Ubuntu).

Обратите внимание: команда sudo нужна для запуска программы с правами root, поскольку для установки пакетов нужны права суперпользователя. Чтобы у пользователя была возможность запуска sudo, он должен быть внесен в /etc/sudoers (см. гл. 11). По умолчанию пользователь, которого вы создаете при установке Debian, не вносится в этот файл, поэтому нужно или получать права root командой su, или добавить пользователя в файл /etc/sudoers.

Основные команды менеджера пакетов apt приведены в таблице 5.2.

Таблица 5.2. Основные команды менеджера пакетов apt

Команда	Описание
install [список пакетов]	Устанавливает пакеты из списка (пакеты указываются через пробел)
remove [список пакетов]	Удаляет пакеты из списка (пакеты указываются через пробел)
check	Поиск нарушенных зависимостей

clean	Очищает локальное хранилище полученных пакетов. При установке пакеты из репозитария загружаются в каталог var/cache/apt/archive. При интенсивной установке программного обеспечения в этом каталоге накапливается довольно много пакетов, поэтому очистка хранилища помогает сэкономить дисковое пространство.
upgrade [список пакетов]	Обновляет указанные пакеты
dist-upgrade	Обновляет всю систему
update	Синхронизирует внутреннюю базу данных о пакетах с источниками пакетов, которые описаны в /etc/apt/sources.list

Конфигурационные файлы

Список репозитариев хранится в файле /etc/apt/sources.list. В отличие от Fedora, все репозитарии в Ubuntu описываются в одном файле. Каждый репозитарий описывается в новой строке, например:

```
deb http://ru.archive.ubuntu.com/ubuntu/ zesty main
restricted
deb http://ru.archive.ubuntu.com/ubuntu/ zesty-updates main
restricted
```

Чтобы добавить репозитарий, просто добавьте строку, описывающую репозитарий в этот файл. После этого нужно ввести команду:

```
sudo apt update
```

5.2.4. Графический менеджер Synaptic

Для облегчения процесса управления пакетами (поиск, установка, удаления) начинающие пользователи могут установить графический менеджер пакетов Synaptic. Установить Synaptic можно командой:

```
sudo apt install synaptic
```

Запустить его можно так:

```
sudo synaptic
```

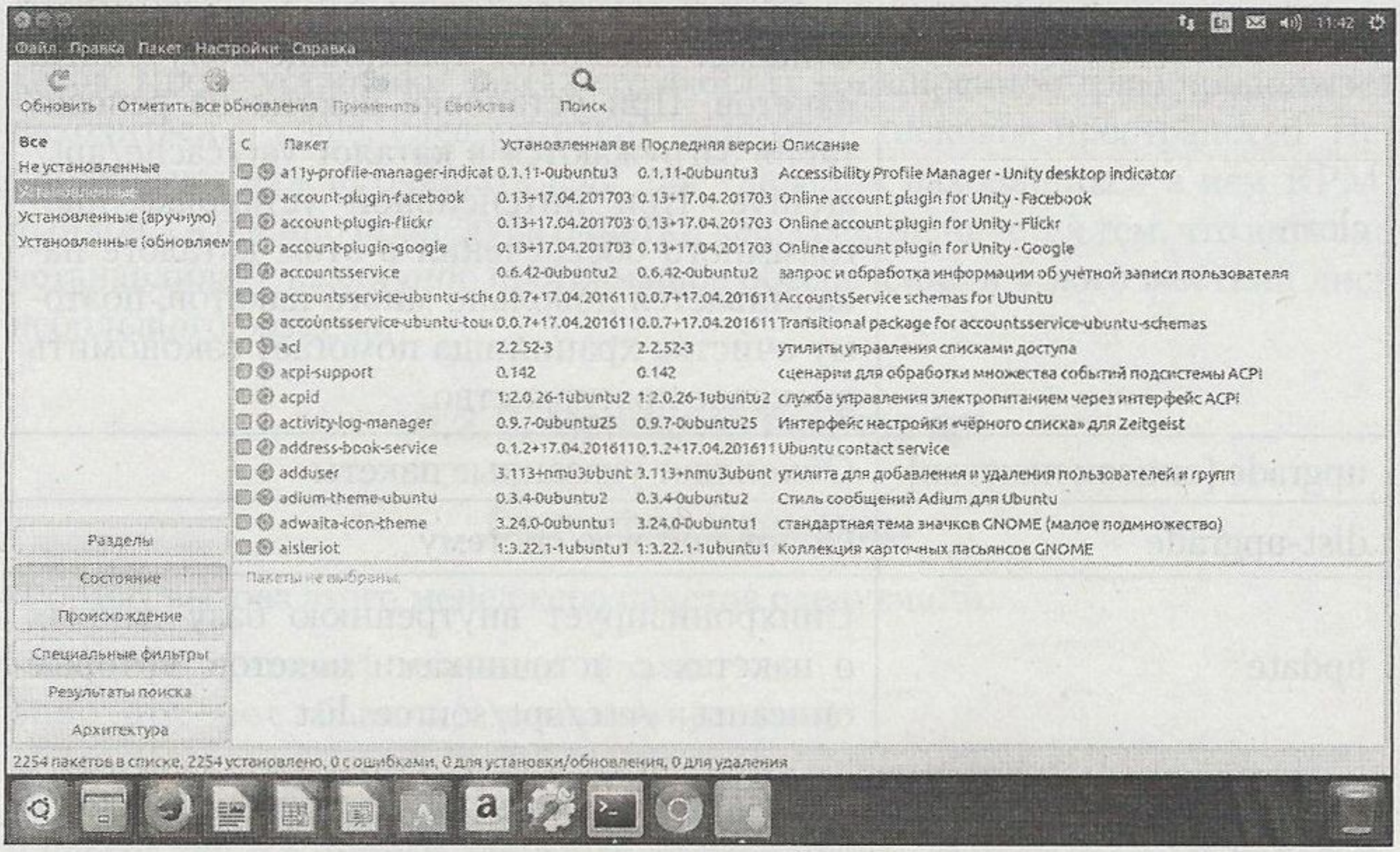


Рис. 5.6. Приложение Synaptic

Приложение Synaptic изображено на рис. 5.6. Установить пакет с его помощью проще простого:

- Нажмите кнопку Поиск и введите название пакета или то, что вы ищете. Например, если вы хотите установить какой-то модуль для php, введите просто php. Программа отобразит результат поиска (рис. 5.7)
- Выберите пакеты, которые вы хотите установить - щелкните на пакете правой кнопкой мыши и выберите **Отметить для установки** (рис. 5.7) Возможно, появится окно, предлагающее установить дополнительные пакеты - разрешение зависимостей. Согласитесь с установкой.
- Для начала установки нажмите кнопку **Применить** на панели Synaptic. В окне **Краткое описание** вы получите список пакетов, которые должны быть установлены, а также будет приведена информация о том, какой размер будет загружен из Интернета и сколько места займут устанавливаемые программы (рис. 5.8).
- Если вы со всем согласны, нажмите кнопку **Применить** в окне **Краткое описание**. Дождитесь, пока пакеты будут установлены (рис. 5.9). Во время установки вы увидите окно с индикатором хода прогресса.

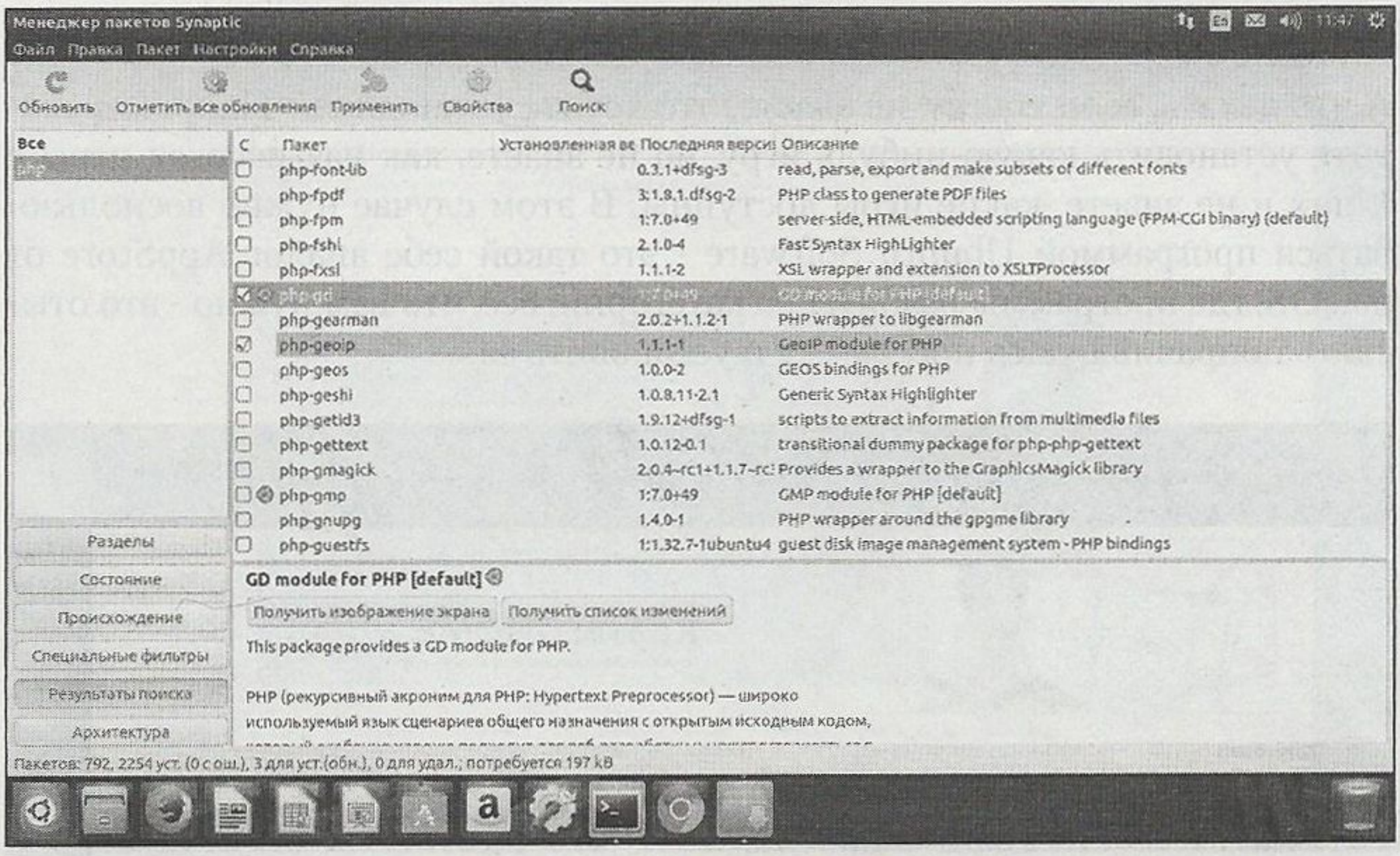


Рис. 5.7. Результат поиска пакета. Необходимые для установки пакеты отмечены

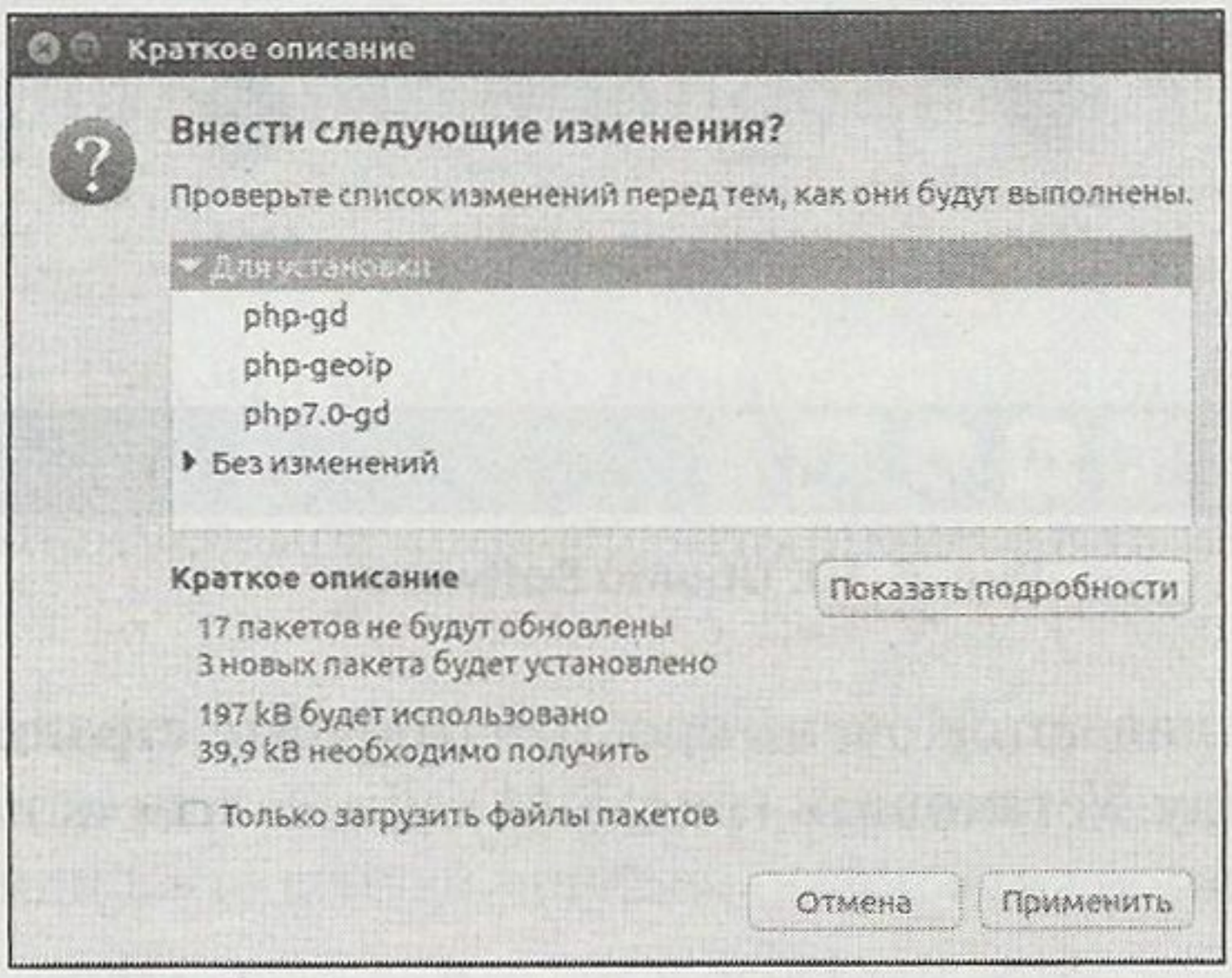


Рис. 5.8. Описание планируемых действий

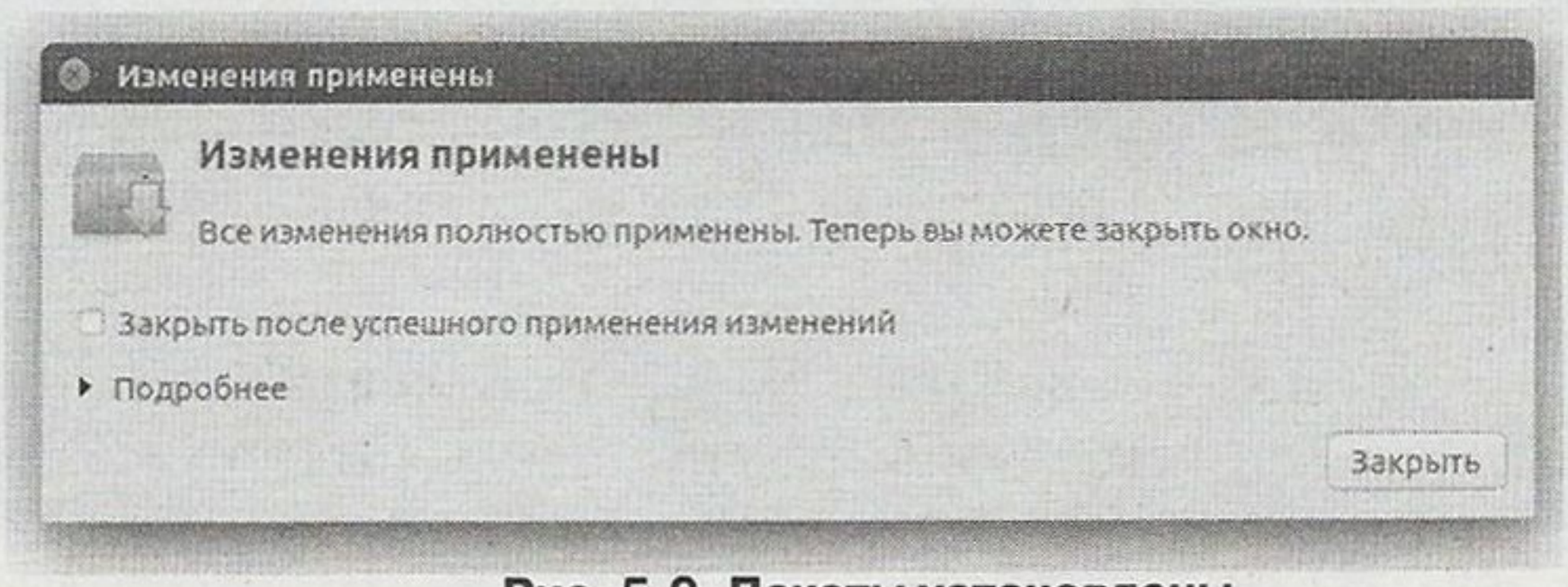


Рис. 5.9. Пакеты установлены

5.2.5. Ubuntu Software

А что делать, если совсем не знаете, что хотите установить? Например, хотите установить какую-нибудь игру, но не знаете, как называются игры в Linux и не знаете, какие игры доступны. В этом случае нужно воспользоваться программой Ubuntu Software - это такой себе аналог AppStore от macOS, где программы разбиты на категории, все, что вам нужно - это отыскать понравившуюся программу и установить ее.

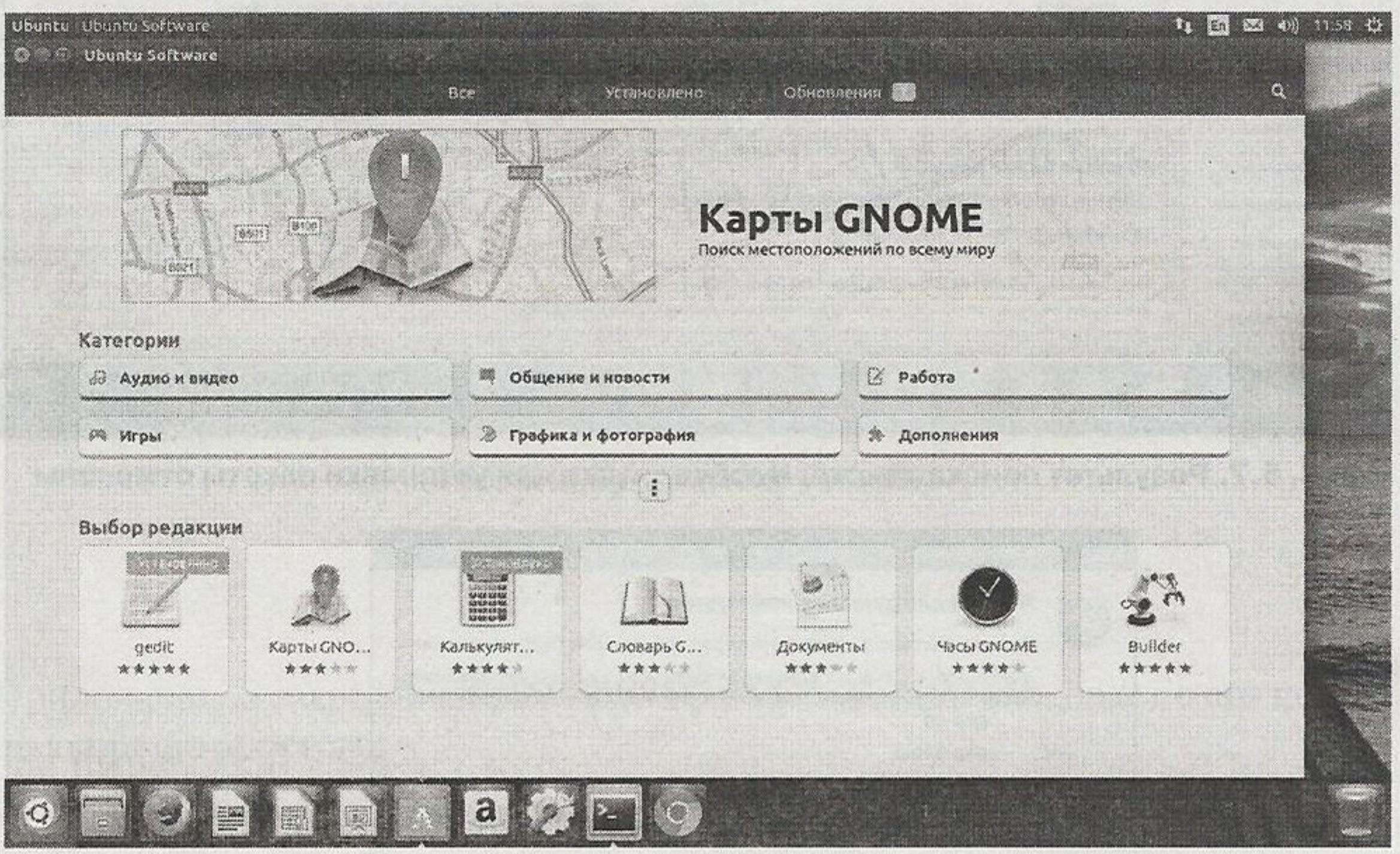


Рис. 5.10. Ubuntu Software

Программы устанавливаются очень просто - откройте страничку программы и нажмите кнопку **Установить** (рис. 5.11). Далее вам останется только дождаться установки.

Рис. 5.11. Установка программ посредством Ubuntu Software

5.3. Управление процессами

5.3.1. Команда ps

Linux устроена так, что каждому процессу присваивается уникальный номер - PID (Process ID, ИД процесса), используя который можно управлять процессом, например, можно завершить процесс или изменить его приоритет.

Узнать PID можно с помощью команды ps. Команда ps, введенная без параметров, просто показывает список процессов, запущенных на текущем терминале. Видно, что сейчас запущен bash и сама команда ps, см. рис. 5.12.

```
ivan@ivan-home: ~
ivan@ivan-home:~$ ps
  PID TTY          TIME CMD
 2500 pts/0    00:00:00 bash
 3927 pts/0    00:00:00 ps
ivan@ivan-home:~$
```

Рис. 5.12. Запущенные процессы

Параметр -a позволяет вывести список всех процессов пользователя. Если нужно вывести процессы какого-то определенного пользователя, тогда используйте параметр -u:

```
$ ps -u ivan
```

Команда ps сортирует процессы по PID. Колонка TTY - это терминал, к которому привязан процесс. Если в этой колонке вы видите знак ?, значит, процесс не привязан ни к одному из терминалов. Как правило, это системные процессы-службы. Они запускаются без привязки к терминалу. Чтобы отобразить только процессы без привязки к терминалу, используется опция -x (рис. 5.13).

```
ivan@ivan-home: ~
  PID TTY      STAT   TIME COMMAND
 1357 ?        Ss     0:00 /lib/systemd/systemd --user
 1358 ?        S      0:00 (sd-pam)
 1364 ?        Sl     0:00 /usr/bin/gnome-keyring-daemon --daemonize --login
 1367 ?        Ss     0:00 /bin/sh /usr/lib/gnome-session/run-systemd-session ub
untu-session.target
 1399 ?        Ss     0:01 /usr/bin/dbus-daemon --session --address=systemd: --n
ofork --nopidfile --systemd-activation
 1468 ?        Ss     0:00 /usr/bin/ssh-agent /usr/bin/im-launch /usr/lib/gnome-
session/run-systemd-session ubuntu-session.target
 1486 ?        Ssl    0:00 /usr/bin/ibus-daemon --daemonize --xim --address unix
:tmpdir=/tmp/ibus
 1492 ?        Ssl    0:00 /usr/lib/gvfs/gvfsd
 1500 ?        Sl     0:00 /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o bi
g_writes
 1506 ?        Sl     0:00 /usr/lib/ibus/ibus-dconf
 1509 ?        Sl     0:00 /usr/lib/ibus/ibus-ui-gtk3
 1516 ?        Sl     0:00 /usr/lib/ibus/ibus-x11 --kill-daemon
 1549 ?        Sl     0:00 /usr/lib/ibus/ibus-engine-simple
 1611 ?        S      0:00 systemctl --user start --wait ubuntu-session.target
 1626 ?        Ssl    0:00 /usr/lib/x86_64-linux-gnu/indicator-session/indicator
-session-service
 1628 ?        Ssl    0:00 /usr/lib/x86_64-linux-gnu/indicator-datetime/indicato
--Далее--
```

Рис. 5.13. Команда ps -x | more

Колонка STAT - это состояние, в котором находится процесс. Возможные состояния:

- R - процесс выполняется в данный момент времени.
- S - ожидание (процесс “спит”).
- D - процесс ожидает ввода/вывода.
- T - процесс остановлен, но пока еще не удален из таблицы процессов.

- X - процесс мертв.
- W - процесс в подкачке.
- Z - процесс зомби (процесс-родитель еще не считал код завершения дочернего процесса, хотя тот уже завершен).

Узнать PID процесса по его имени можно так:

```
ps -A | grep <имя>
```

Например:

```
ps -A | grep proftpd
```

Здесь опция -A выводит все процессы в системе.

5.3.2. Команда nice

Когда мы знаем PID процесса, мы можем изменить его приоритет. В некоторых случаях полезно изменить приоритет процесса. Например, можно повысить приоритет процесса, выполняющего резервное копирование, чтобы программа успела за ночь создать все необходимые резервные копии и чтобы этот процесс утром потом не мешал нормальной работе сервера.

Запустить программу с определенным приоритетом можно командой nice:

```
# nice -n <приоритет> команда аргументы
```

Здесь приоритет задается от -20 (максимальный приоритет) до 19 (минимальный). Если процесс уже был запущен и вы не можете его прерывать, но повысить приоритет нужно, используйте команду renice:

```
# renice -n <приоритет> -p PID
```

5.3.3. Команда kill

Если процесс завис и его нельзя завершить, как обычно, тогда для его аварийного завершения используется команда kill. Формат вызова этой команды следующий:

```
$ kill [опции] PID
```

Конечно, перед этим нужно узнать PID процесса. Используя параметры программы, можно по-разному завершить процесс. Самый эффективный сигнал 9 (KILL) - означает аварийное завершение процесса. Программа не может игнорировать или как-либо обработать этот процесс.

Если нужно попытаться корректно завершить работу программы, ей отправляют сигнал 15 (TERM), означающий, что программа должна освободить все занятые ресурсы, сохранить все данные. Вот только если программа зависла и не отвечает на запросы пользователя, этот сигнал мало чем поможет, но попытаться стоит.

Сигнал 19 (STOP) позволяет временно приостановить работу программы, а сигнал 18 (CONT) - возобновить приостановленный ранее процесс.

Для сетевых служб полезен сигнал 1 (HUP), означающий, что процесс должен перезапуститься и перечитать файл конфигурации. Полезно, когда вы изменили файл конфигурации и хотите, чтобы демон был перезапущен (хотя для этого правильнее использовать команду service). Обычная программа при получении сигнала 1 завершает работу.

Пример отправки сигнала:

```
$ kill -9 1035
```

Если вам лень получать PID процесса, можно завершить его и по имени, используя команду killall, например:

```
$ killall mc
```

Вот только если в вашей системе есть два процесса с именем mc, например, один в одном терминале, а другой - в другом, то будут завершены оба процесса. Если это то, что вам нужно, используйте killall, в противном случае лучше использовать команду kill для завершения именно того процесса, который можно завершить.

Примечание. Обычный пользователь может завершить только свои процессы, суперпользователь может завершить любой процесс.

5.3.4. Команда top

Как было отмечено ранее, программа ps по умолчанию сортирует процессы по колонке PID, а не по колонке TIME. Конечно, можно использовать раз-

личные параметры программы, чтобы добиться нужного нам вывода, но все равно программа не будет показывать ситуацию в реальном времени. Если же вам нужно знать, что происходит с вашими процессами в реальном времени, вам нужно использовать программу top (рис. 5.14).

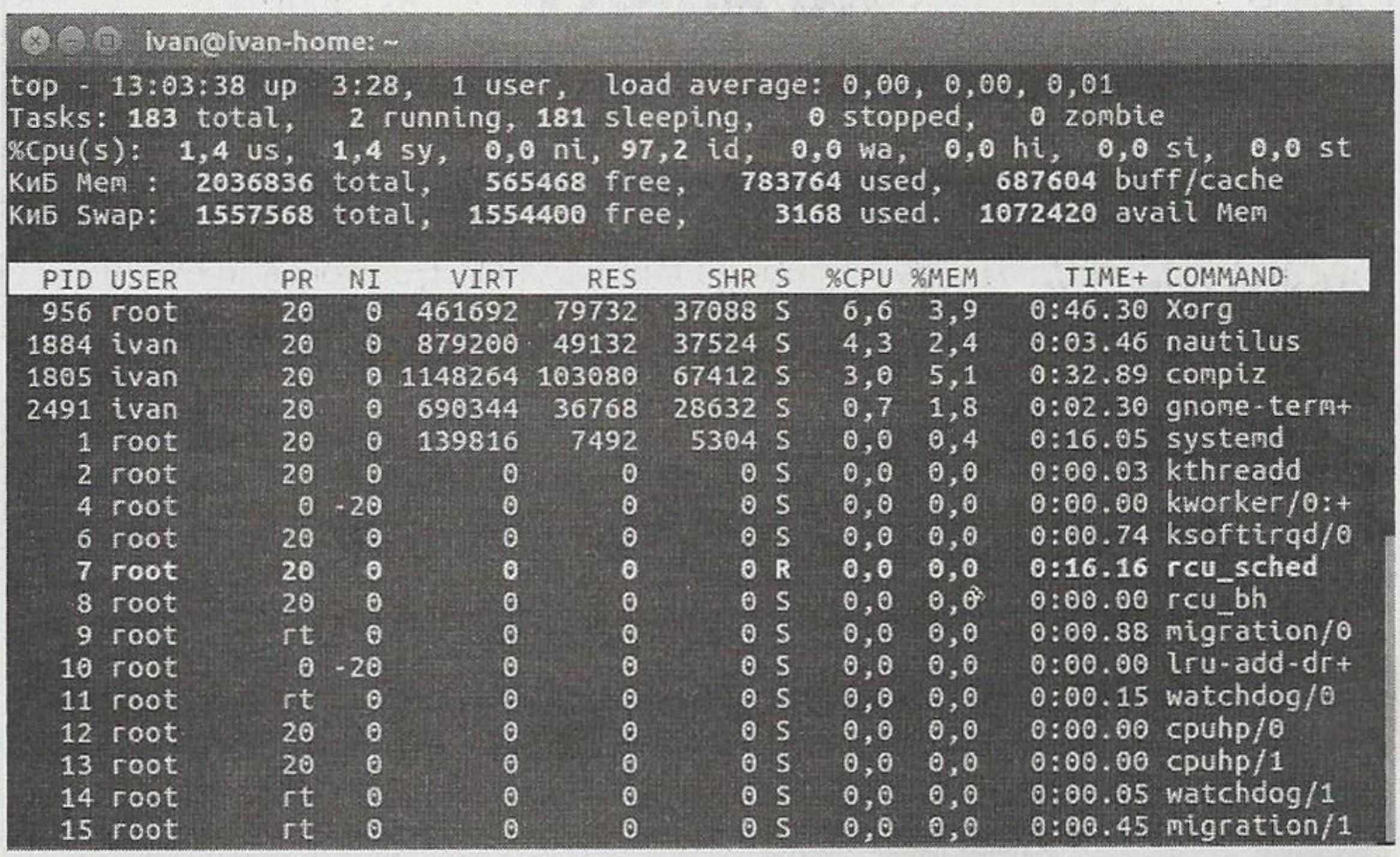


Рис. 5.14. Команда top

Назначение колонок программы описано в таблице 5.3.

Таблица 5.3. Колонки программы top

Колонка	Описание
PID	PID процесса
USER	Владелец процесса (пользователь, запустивший программу)
PR	Приоритет процесса
NI	Значение nice (см. ранее)
VIRT	Виртуальная память, которая используется процессом
RES	Размер процесса, который не перемещается в область подкачки.

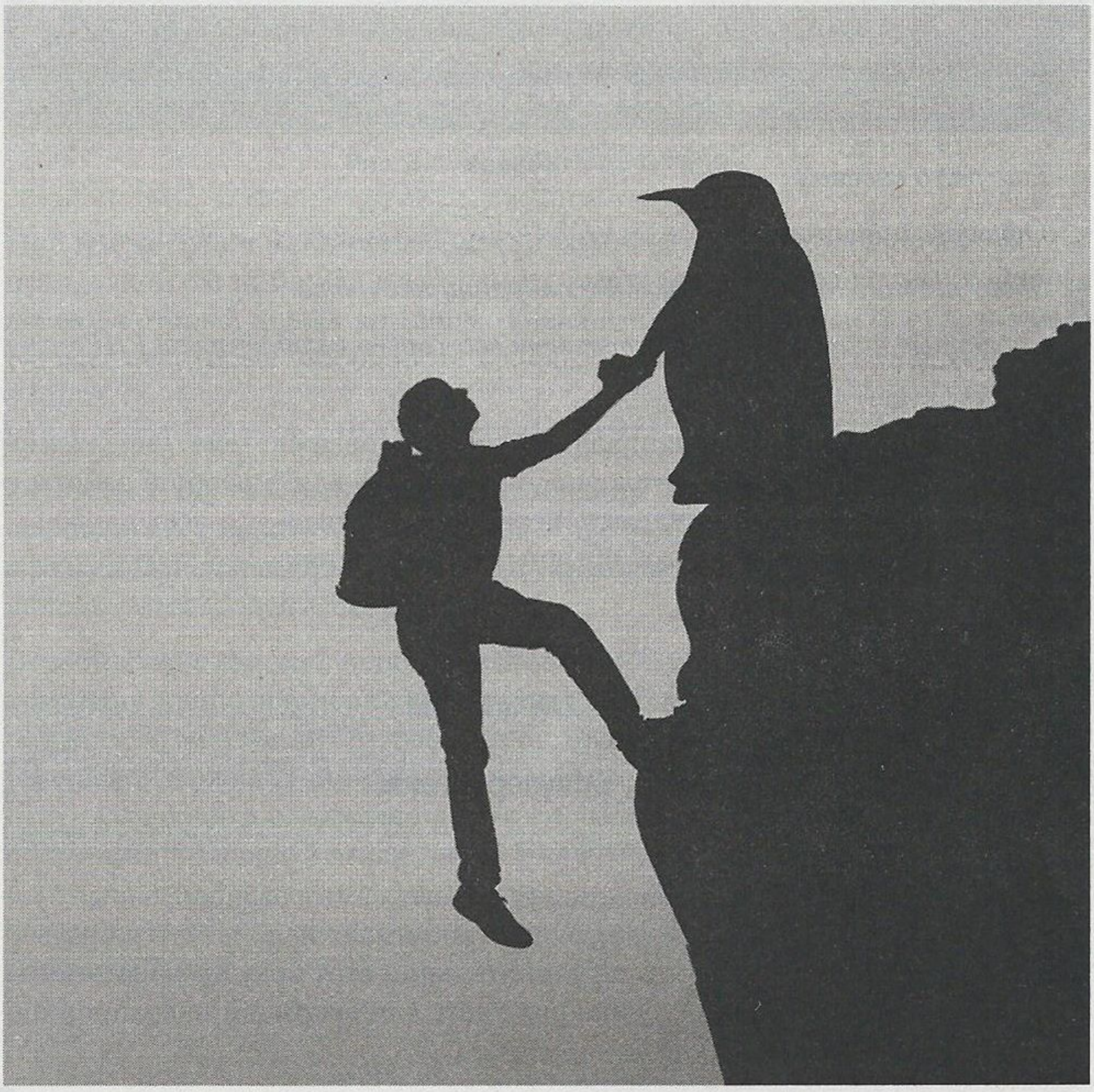
SHR	Разделяемая память, используемая процессом
S	Состояние процесса (см. ранее)
%CPU	Процессорное время, занимаемое процессом в данный момент
%MEM	Память, используемая процессом
TIME+	Процессорное время, которое было потрачено с момента запуска процесса
COMMAND	Команда запуска процесса

При просмотре списка программы top вы можете управлять сортировкой процессов с помощью нажатия клавиши F, которая изменяет колонку, по которой сортируется список процессов. По умолчанию сортировка осуществляется по процессорному времени.

Нажатие клавиши U показывает только процессы определенного пользователя. После нажатия U нужно будет ввести имя пользователя, процессы которого вы хотите просмотреть, или нажать Enter, чтобы просмотреть процессы всех пользователей.

ГЛАВА 6.

Популярные программы в Linux



В этой главе:

- *Браузеры Firefox и Chromium*
- *Почтовый клиент Thunderbird*
- *Установка Skype в Linux*
- *BitTorrent-клиент Transmission: качаем фильмы и музыку с торрентов*
- *Мультимедиа-приложения*
- *Еще раз о кодеках*
- *Офисные приложения*
- *Shotwell Photo Manager: основные операции над фото*
- *Серверные приложения. Собственный веб-сервер с поддержкой PHP*

Ранее была рассмотрена командная строка, файловая система, управление пакетами в Linux - все то, безусловно, очень важно для успешной работы в системе. Но настало время сделать перерыв, отвлечься ото всех сложных материй и просто поговорить об обычных приложениях.

6.1. Интернет-приложения

6.1.1. Браузер Firefox

Как и в Windows, в Linux вы можете установить самые разнообразные браузеры. По умолчанию в современных дистрибутивах используется браузер Mozilla Firefox (рис. 6.1). И это не удивительно - этот браузер обладает открытым кодом и как нельзя лучше подходит для открытой операционной системы.

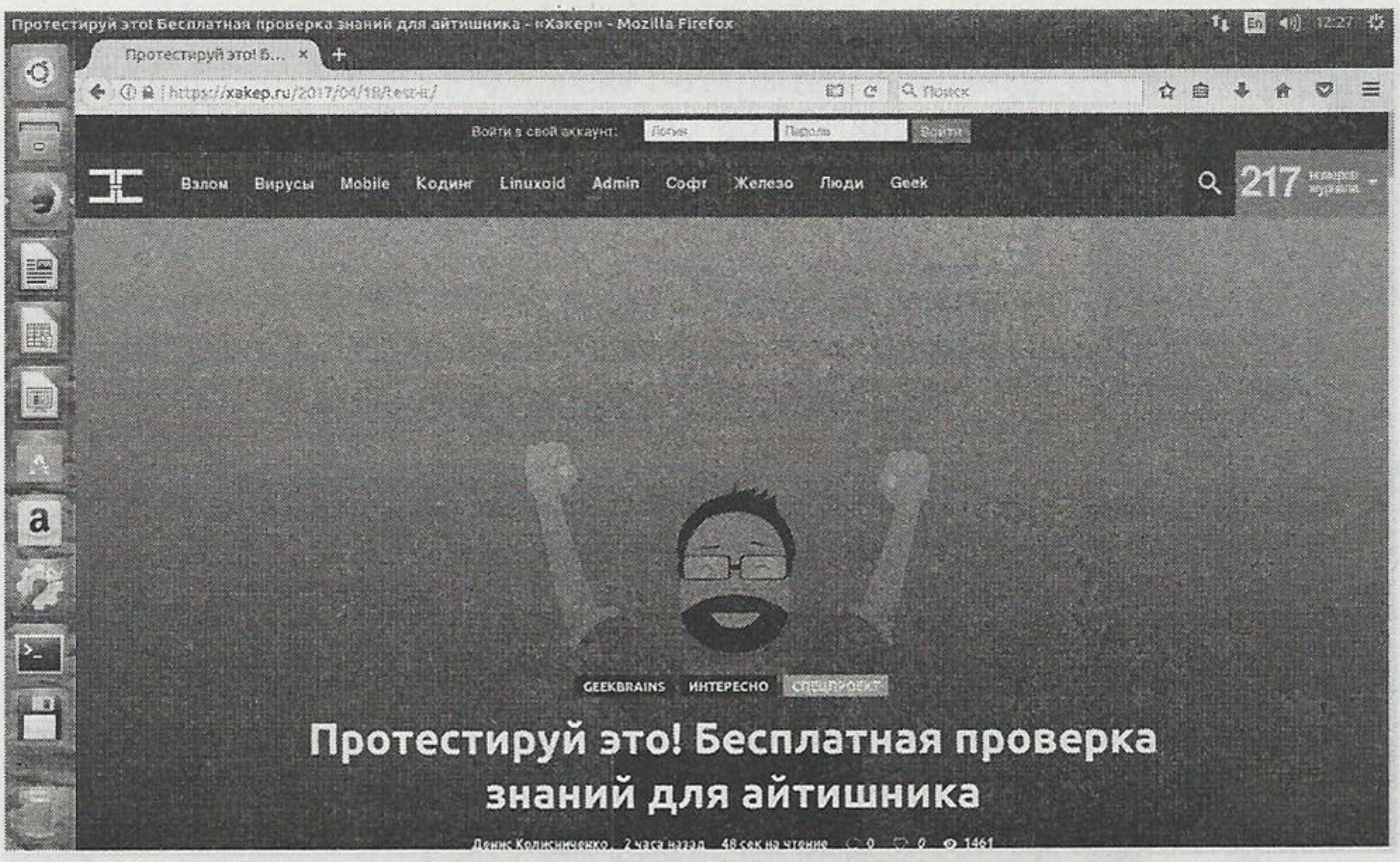


Рис. 6.1. Браузер Mozilla Firefox

Нет, мы не будем разбираться, как использовать браузер. По меньшей степени, такой бы материал в этой книге выглядел очень бы глупо. Зачем человеку, который дошел до Linux, объяснять, как пользоваться браузером? Но есть некоторые особенности использования Firefox - как в целом, так и в Linux.

Если вы до этого использовали другой браузер, то наверняка вам будет интересно, как сменить поисковую машину. Не секрет, что многие из нас предпочитают использовать Яндекс. В настройках браузера нужно перейти в раздел Поиск и выбрать в нем другую поисковую машину (по умолчанию используется Google), см. рис. 6.2.

Второй нюанс связан с использованием Adobe Flash в Firefox. Официально компания Adobe остановила поддержку Flash в Linux еще в 2012 году. С тех пор кстати, многие веб-разработчики отказались от использования Flash и теперь поддержка этой технологии не так критична. Если все же вам нужен Flash, установить его можно двумя способами. Первый заключается в использовании Google Chrome, но он уже не поддерживает 32-битные версии Linux, поэтому если у вас таковая, то использовать вы его не сможете. Да и не каждый пользователь хочет устанавливать на свой Linux проприетарный (закрытый) код.

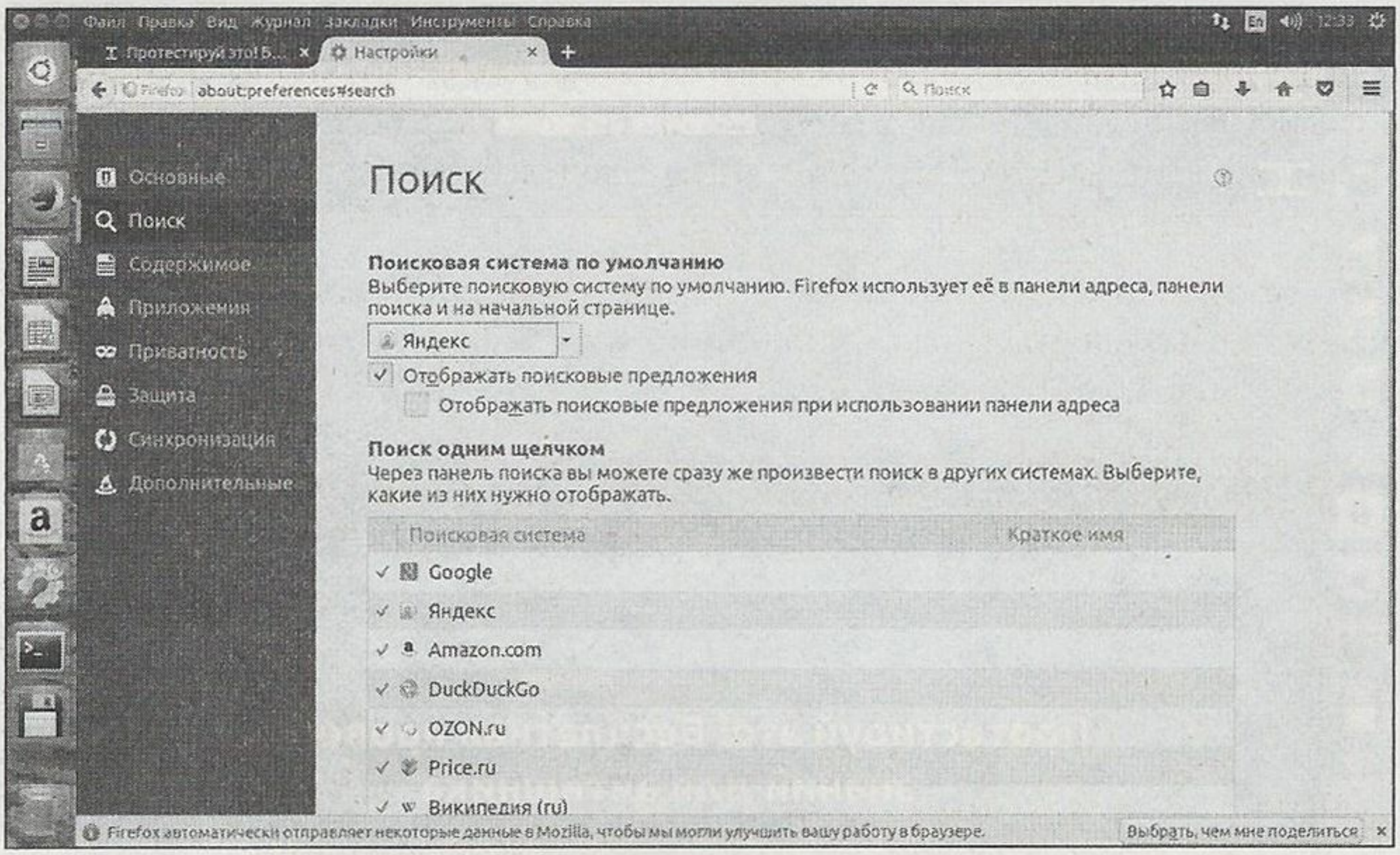


Рис. 6.2. Выбор поисковой системы

Второй способ следующий: введите в адресной строке браузера следующий URL apt://flashplugin-installer и следуйте инструкциями установщика (рис. 6.3).

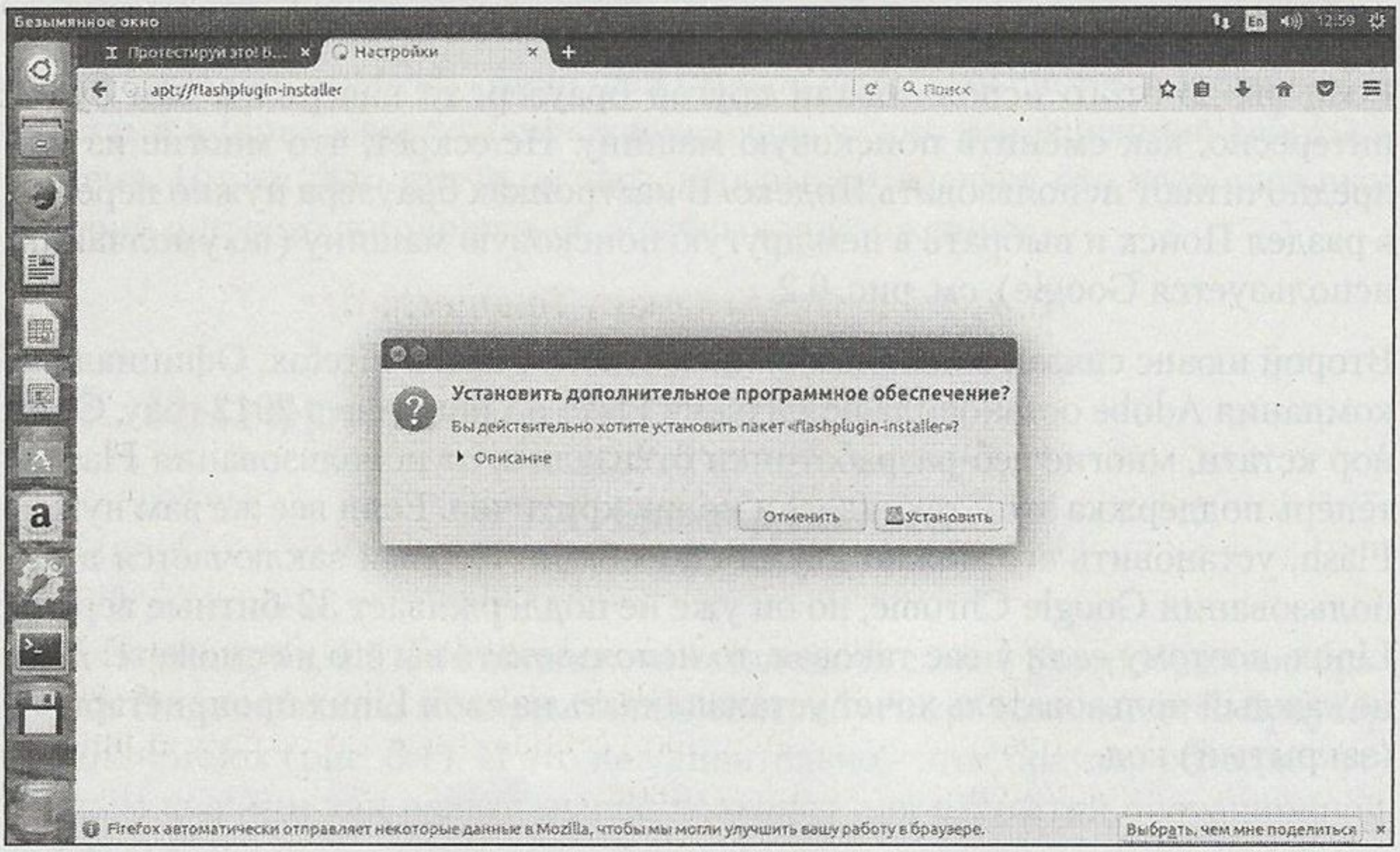


Рис. 6.3. Установка Flash в Linux

Если вы привыкли к Chrome, то можете попробовать установить браузер Chromium. Chromium - это проект с открытым исходным кодом, на котором и основан Chrome. Тем не менее, там до сих пор нет нескольких полезных функций, например, поддержки H.264. С другой стороны, Google не собирает статистику с пользователей Chromium.

Для установки Chromium установите пакет chromium-browser. Остальные необходимые пакеты вроде chromium-browser-l10n (интернационализация) будут установлены автоматически.

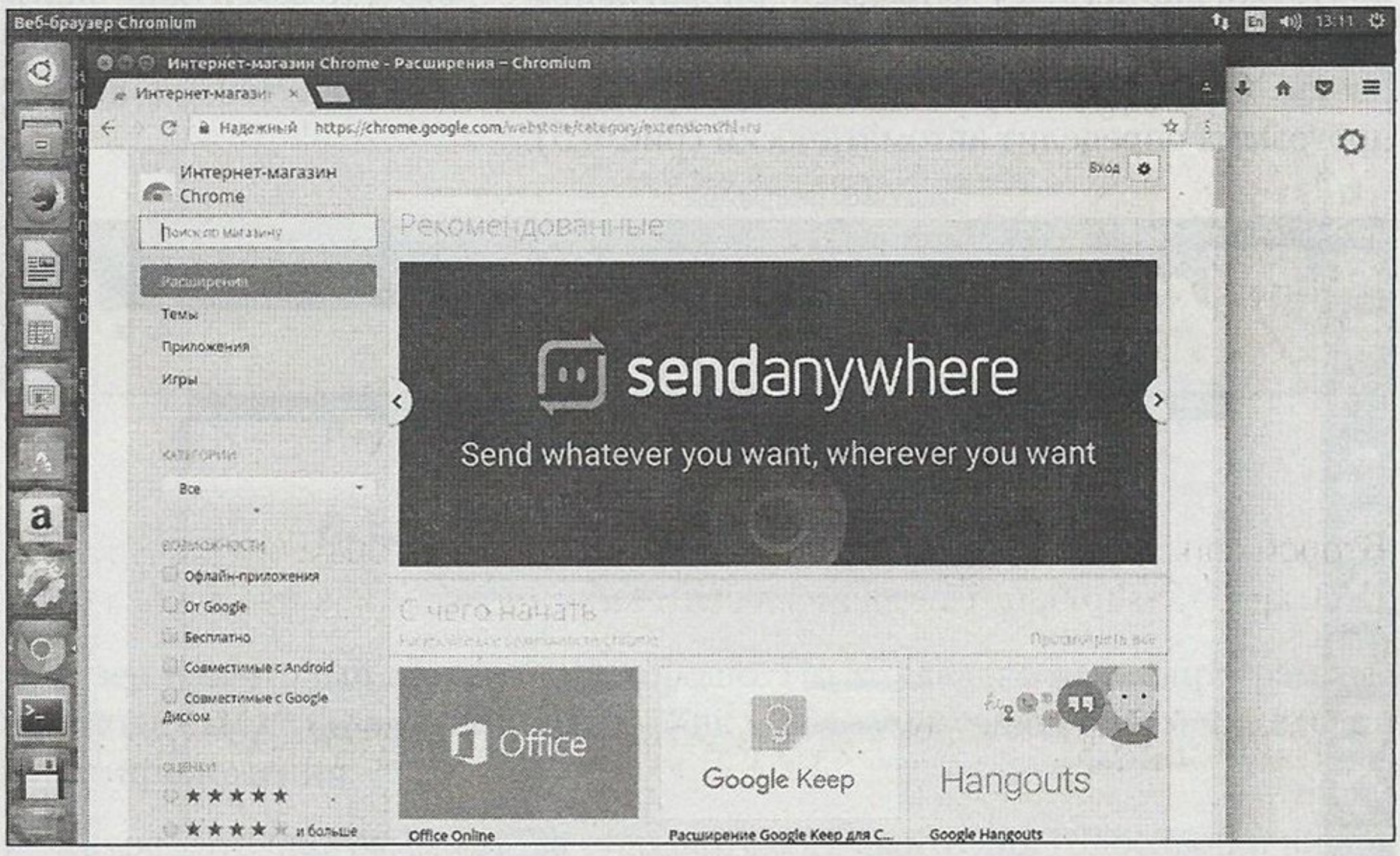


Рис. 6.4. Браузер Chromium

6.1.2. Почтовый клиент Thunderbird

В последнее время наблюдается некоторая стандартизация в сфере программного обеспечения. Например, раньше в качестве браузера по умолчанию мог использоваться абсолютно любой - все зависело от дистрибутива. Помню, когда-то очень популярным Konqueror - он устанавливался по умолчанию в дистрибутивах, где по умолчанию использовалась графическая среда KDE. Встречались и другие браузеры, например, Pale Moon, Conkeror и др. Сейчас же по умолчанию устанавливается Firefox, причем в большинстве современных дистрибутивов.

Аналогичная ситуация наблюдается и с почтовым клиентом. Разработчики разных дистрибутивов отказались от всяких разных почтовых программ в пользу Mozilla Thunderbird.

Нужно отметить, что Mozilla Thunderbird - очень удобный почтовый клиент. Мне очень нравится его мастер настройки учетной записи. Если у вас учетная запись на публичном почтовом сервере вроде gmail.com, mail.ru, то все, что вам нужно знать - это электронный адрес и пароль, который вы используете для входа в почтовый ящик через ваш браузер. Если даже у вас почтовый ящик (как у меня) на корпоративном сервере, то вам нужно знать электронный адрес, пароль, а также адреса серверов входящей и исходящей почты. Остальные параметры - порт, протокол SSL, тип аутентификации программа определит автоматически (рис. 6.5).

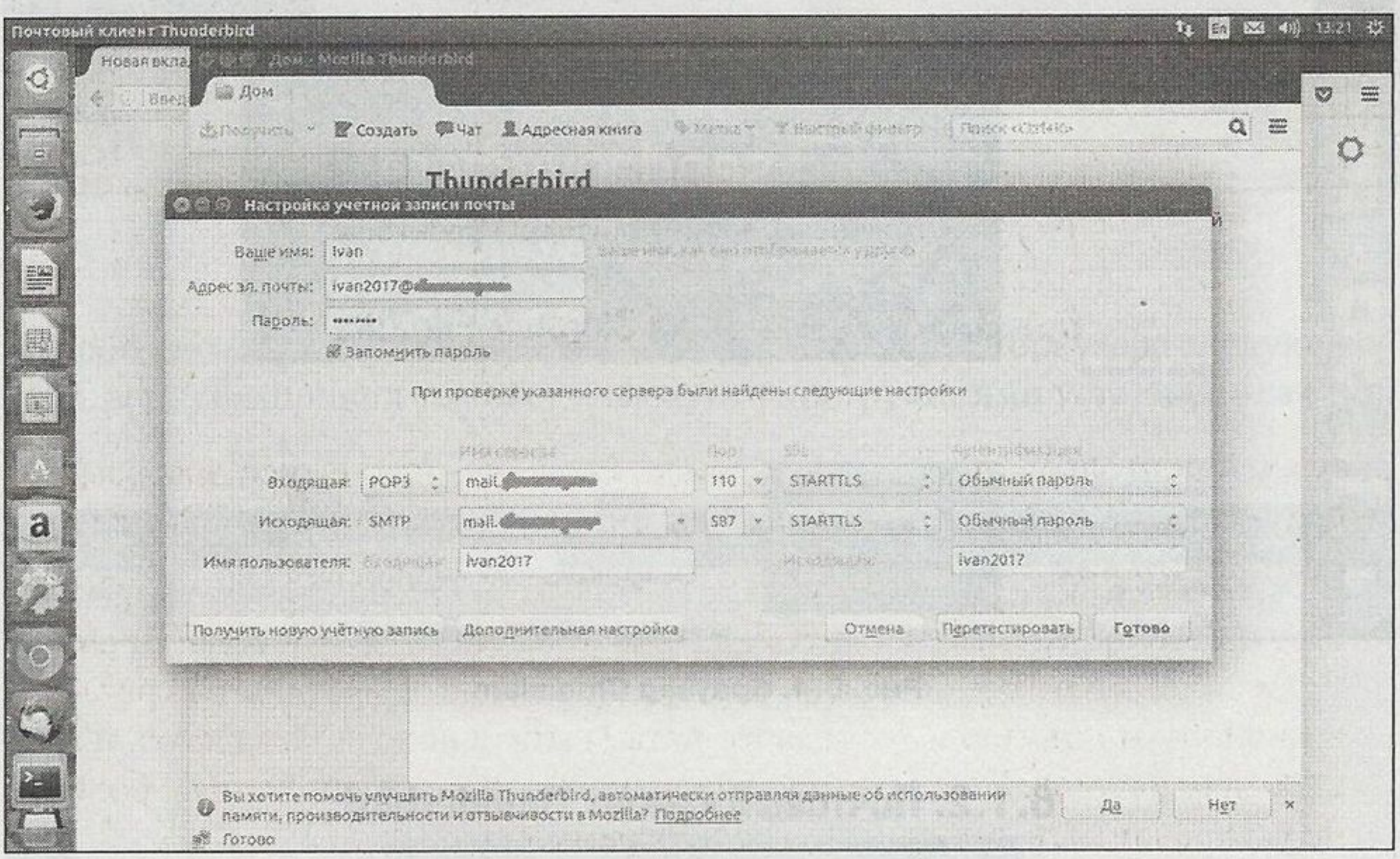


Рис. 6.5. Почтовый клиент правильно определил параметры учетной записи

Сам же почтовый клиент мало чем отличается от других почтовых клиентов - почта поступает во входящие вашей учетной записи, поддерживаются отдельные папки Входящие (для разных почтовых ящиков) благодаря чему входящая корреспонденция не перемешивается, поддерживаются фильтры, позволяющие «разнести» корреспонденцию по разным папкам. В общем, приятный, интуитивно понятный интерфейс.

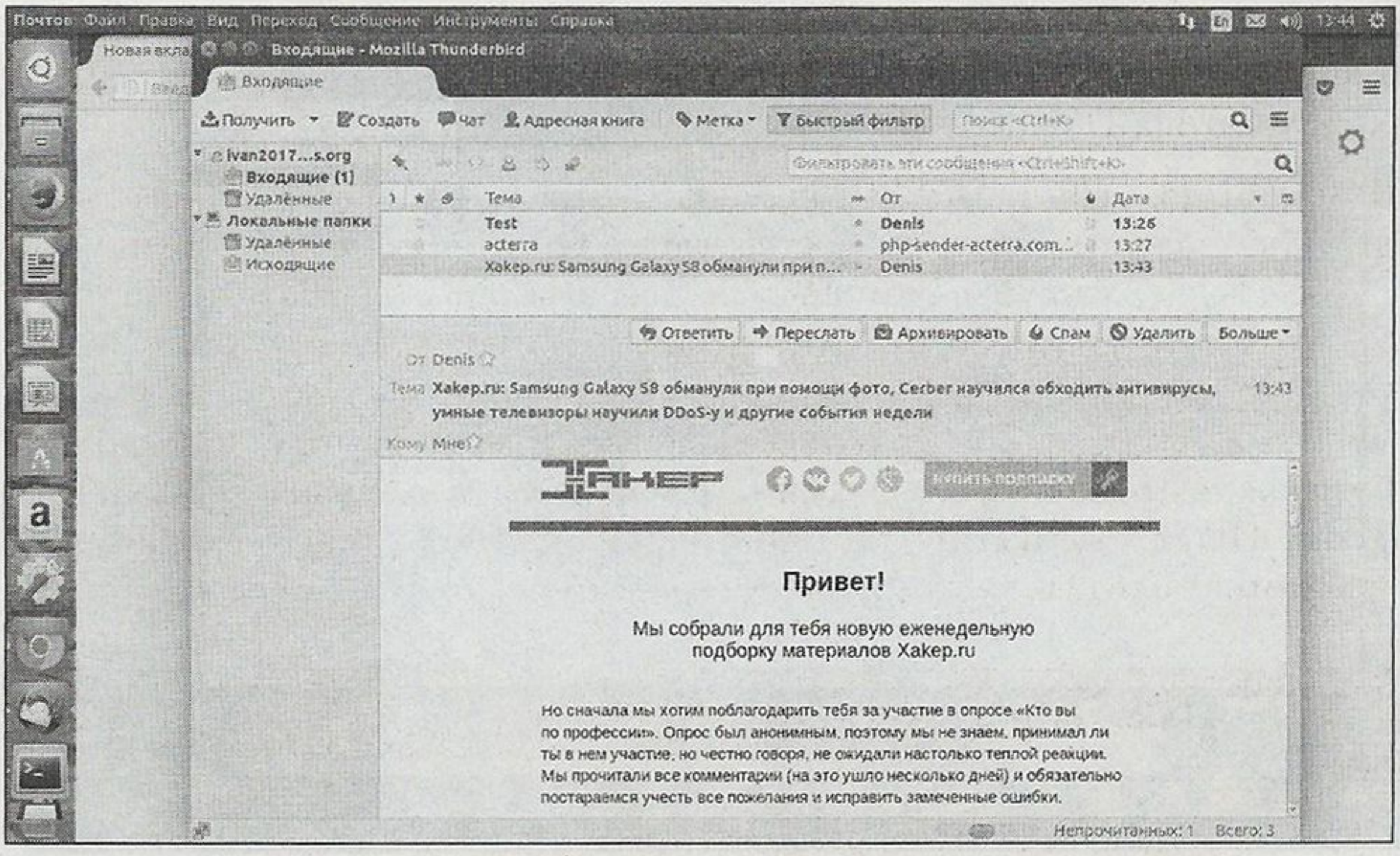


Рис. 6.6. Почтовый клиент Mozilla Thunderbird

6.1.3. Устанавливаем Skype и FileZilla

Браузер и почтовый клиент - это хорошо. Но это далеко не все программы, необходимые на сегодняшний день. Как минимум нужно установить следующие приложения:

- Skype - эта программа стала стандартом де-факто для голосовой и видео связи, а также вытеснила все мессенджеры вроде ICQ - ведь Skype можно использовать и для текстового чата (в том числе группового), и для передачи файлов. По большому счету можно установить Skype и больше не устанавливать никакие другие программы.
- FileZilla - лучший FTP-клиент. Не понимаю, почему он не устанавливается по умолчанию.

Установка Skype

Итак, приступим. Начнем со Skype и дистрибутива Ubuntu. В Ubuntu установить Skype очень и очень просто. Откройте окно **Параметры системы**, далее в нем выберите **Программы и обновления**. В открывшемся окне перейдите на вкладку **Другое ПО** и включите флажок **Партнеры Canonical** (рис. 6.7). Закрывать окна не нужно.

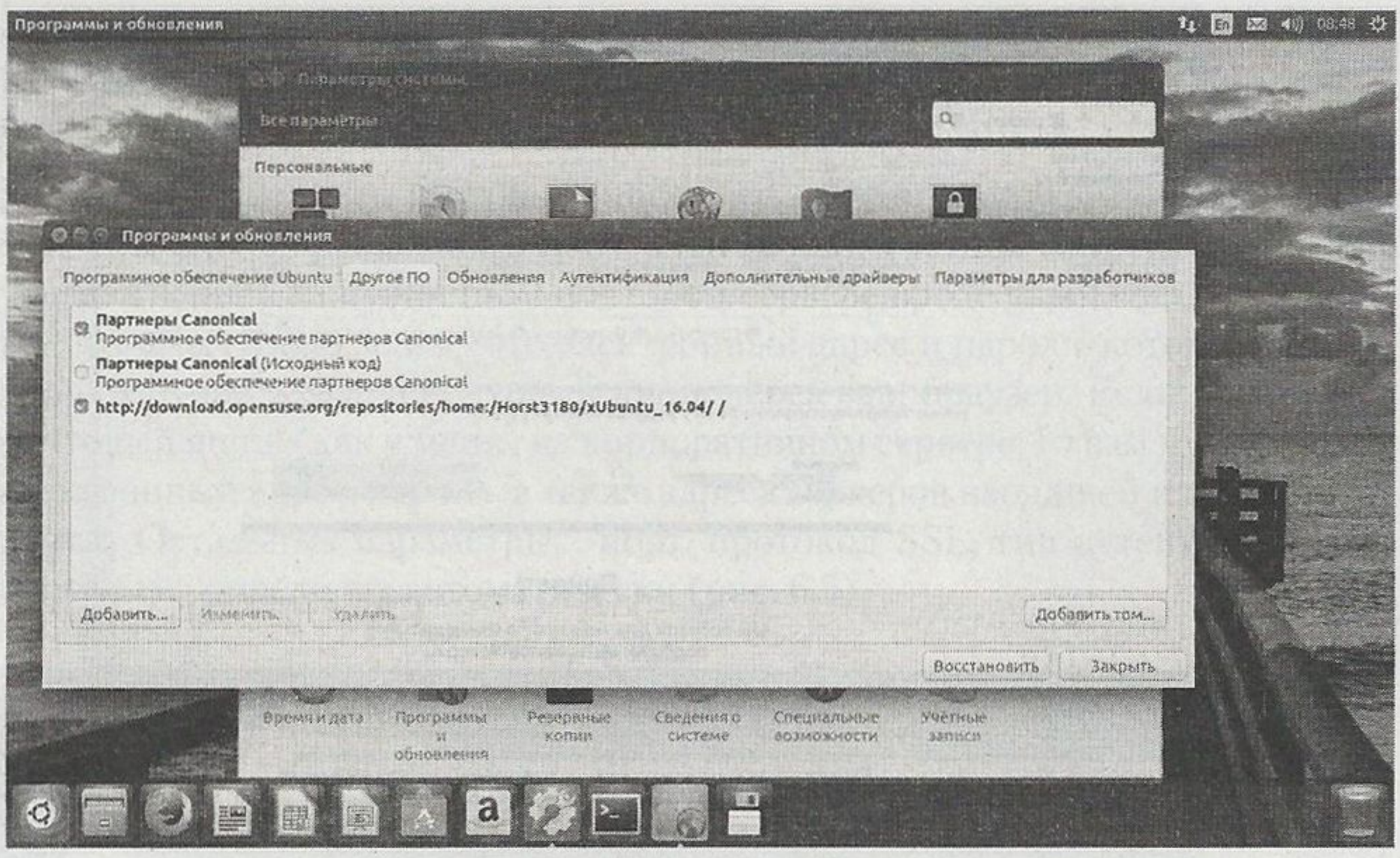


Рис. 6.7. Добавление репозитория

После этого откройте терминал и введите команды:

```
sudo apt update
sudo apt install skype
```

Первая команда обновит информацию о доступных пакетах (рис. 6.8), а вторая - установит skype (рис. 6.9). Для установки Skype нужно будет загрузить примерно 87 Мб информации с Интернета - если у вас мобильное 3G соединение, учитывайте это.

Если вы используете Lubuntu, введите еще команду установки pulseaudio:

```
sudo apt install pulseaudio
```

Осталось только запустить Skype, используя средство поиска (первая кнопка на панели запуска), см. рис. 6.10. После запуска Skype для большего удобства щелкните на его значке правой кнопкой мыши и выберите команду **Закрепить на панели** - так вы упростите запуск программы (рис. 6.11).

Теперь рассмотрим процедуру установки Skype в Fedora. Сразу скажу - будет сложнее. Именно поэтому в начале этой книги я рекомендовал дистрибутив Ubuntu для новичков. Но если выбора у вас нет (например, купили

```
ivan@ivan-home: ~
ivan@ivan-home:~$ sudo apt update
[sudo] пароль для ivan:
Сущ:1 http://ua.archive.ubuntu.com/ubuntu zesty InRelease
Пол:2 http://ua.archive.ubuntu.com/ubuntu zesty-updates InRelease [89,2 kB]
Пол:3 http://archive.canonical.com/ubuntu zesty InRelease [10,2 kB]
Сущ:4 http://ua.archive.ubuntu.com/ubuntu zesty-backports InRelease
Игн:5 http://download.opensuse.org/repositories/home:/Horst3180/xUbuntu_16.04  I
nRelease
Сущ:6 http://security.ubuntu.com/ubuntu zesty-security InRelease
Сущ:7 http://download.opensuse.org/repositories/home:/Horst3180/xUbuntu_16.04  R
elease
Пол:8 http://archive.canonical.com/ubuntu zesty/partner i386 Packages [2.144 B]
Пол:9 http://archive.canonical.com/ubuntu zesty/partner amd64 Packages [1.840 B]
Пол:10 http://archive.canonical.com/ubuntu zesty/partner Translation-en [1.084 B
]
Получено 104 kБ за 1с (53,7 kБ/с)
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Все пакеты имеют последние версии.
ivan@ivan-home:~$
```

Рис. 6.8. Команда apt update

```
ivan@ivan-home: ~
Настраивается пакет libxi6:i386 (2:1.7.9-1) …
Настраивается пакет libcairo2:i386 (1.14.8-1) …
Настраивается пакет libasound2-plugins:i386 (1.1.1-1ubuntu1) …
Настраивается пакет libqt4-xmlpatterns:i386 (4:4.8.7+dfsg-7ubuntu1) …
Настраивается пакет libgl1-mesa-glx:i386 (17.0.3-1ubuntu1) …
update-alternatives: используется /usr/lib/i386-linux-gnu/mesa/ld.so.conf для пр
едоставления /etc/ld.so.conf.d/i386-linux-gnu_GL.conf (i386-linux-gnu_gl_conf) в
 автоматическом режиме
Настраивается пакет libtheora0:i386 (1.1.1+dfsg.1-14) …
Настраивается пакет gstreamer1.0-plugins-base:i386 (1.10.4-1ubuntu1) …
Настраивается пакет libqt4-declarative:i386 (4:4.8.7+dfsg-7ubuntu1) …
Настраивается пакет libqtgui4:i386 (4:4.8.7+dfsg-7ubuntu1) …
Настраивается пакет qt-at-spi:i386 (0.4.0-5) …
Настраивается пакет libdbusmenu-qt2:i386 (0.9.3+16.04.20160218-0ubuntu1) …
Настраивается пакет libqt4-opengl:i386 (4:4.8.7+dfsg-7ubuntu1) …
Настраивается пакет sni-qt:i386 (0.2.7+17.04.20170112-0ubuntu1) …
Настраивается пакет libqtwebkit4:i386 (2.3.2-0ubuntu13) …
Настраивается пакет skype-bin:i386 (4.3.0.37-0ubuntu0.12.04.1) …
Настраивается пакет skype (4.3.0.37-0ubuntu0.12.04.1) …
Обрабатываются триггеры для libc-bin (2.24-9ubuntu2) …
Обрабатываются триггеры для dbus (1.10.10-1ubuntu2) …
ivan@ivan-home:~$
```

Рис. 6.9. Skype установлен

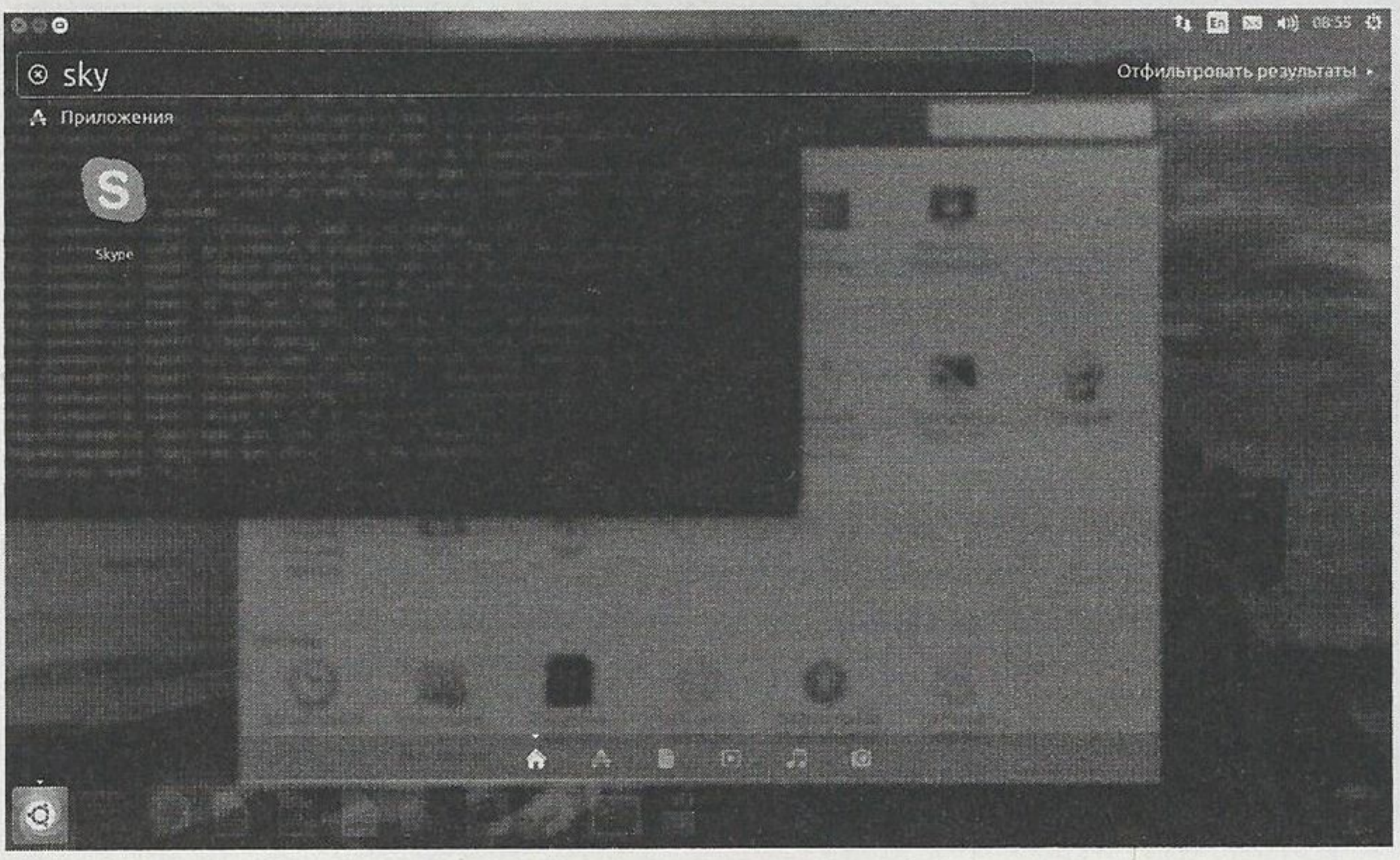


Рис. 6.10. Поиск приложения

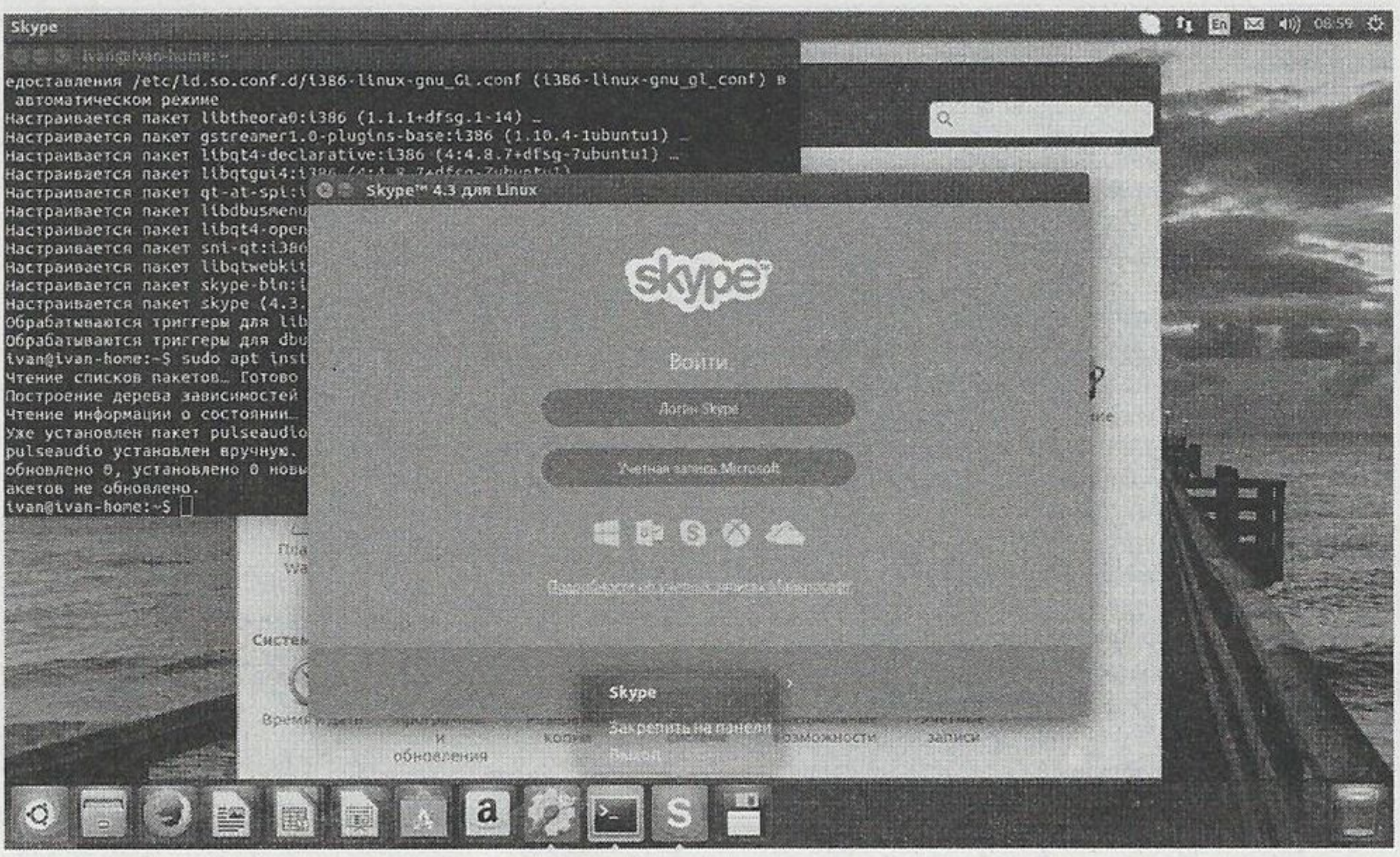


Рис. 6.11. Skype запущен

компьютер с установленным дистрибутивом Fedora), то приготовьте вводить множество команд в терминале. Первым делом введите команду:

```
su
```

Она необходима для получения прав root. Установим необходимые пакеты:

```
dnf install alsa-lib.i686 fontconfig.i686 freetype.
i686 glib2.i686 libSM.i686 libXScrnSaver.i686 libXi.
i686 libXrandr.i686 libXrender.i686 libXv.i686 libstdc++.
i686 pulseaudio-libs.i686 qt.i686 qt-x11.i686 zlib.i686
qtwebkit.i686
```

У вас должна получиться одна длинная команда. После установки пакетов нужно загрузить Skype:

```
cd /tmp
wget --trust-server-names http://www.skype.com/go/getskype-
linux-dynamic
```

Распакуем архив со Skype:

```
mkdir /opt/skype
tar xvf skype-4.3* -C /opt/skype --strip-components=1
```

В принципе, можно найти исполнимый файл в /opt/skype и запустить. Но хочется, чтобы было красиво, поэтому нужно создать ланчер - кнопку запуска. Для этого создайте файл create_launcher:

```
cd ~
touch create_launcher
chmod +x create_launcher
gedit create_launcher
```

Содержимое этого файла приведено в листинге 6.1.

Листинг 6.1. Файл create_launcher

```
#!/bin/bash
ln -s /opt/skype/skype.desktop /usr/share/applications/
skype.desktop

for icon in /opt/skype/icons/*; do
ressuffix="${icon##*_}"
res="${ressuffix%%.*}"
ln -s "$icon" /usr/share/icons/hicolor/"$res"/apps/skype.
png
```

```
done

gtk-update-icon-cache /usr/share/icons/hicolor

touch /usr/bin/skype
chmod 755 /usr/bin/skype
cat << EOF > /usr/bin/skype
#!/bin/sh
export SKYPE_HOME="/opt/skype"

\$SKYPE_HOME/skype --resources=\$SKYPE_HOME \$*
EOF
```

Данный сценарий делает следующее:

- Создает кнопку запуска
- Связывает иконку с кнопкой запуска
- Обновляет кэш пиктограмм GTK
- Создает пустой файл /usr/bin/skype и устанавливает права доступа к нему
- Добавляет в файл /usr/bin/skype следующие строки:

```
#!/bin/sh
export SKYPE_HOME="/opt/skype"

$SKYPE_HOME/skype --resources=$SKYPE_HOME $*
```

По сути, выполняет весь процесс настройки Skype после его распаковки. Запустите сценарий:

```
./create_launcher
```

Данный сценарий нужно запустить всего лишь один раз - только после установки. Далее можете смело его удалить:

```
rm create_launcher
```

Запустить Skype можно или через меню GNOME или с помощью команды:

```
skype
```

Установка FileZilla

FileZilla - лучший FTP-клиент, поверьте моему опыту как веб-разработчика. Особенно он хорошо себя показывает при закачке на FTP-сервер множества мелких файлов, например, при загрузке готовой CMS вроде Joomla! или WordPress, когда приходится загружать несколько тысяч небольших файлов (PHP-сценарии, шаблоны, картинки, таблицы стилей и т.д.).

Установить FileZilla очень просто (именно поэтому я решил начать со Skype - чтобы потом можно было отдохнуть):

```
sudo apt install filezilla
su; dnf install filezilla
```

Первую команду нужно ввести в Ubuntu (или любом другом Debian-совместимом дистрибутиве), вторая команда предназначена для Fedora. Сама программа FileZilla изображена на рис. 6.12.

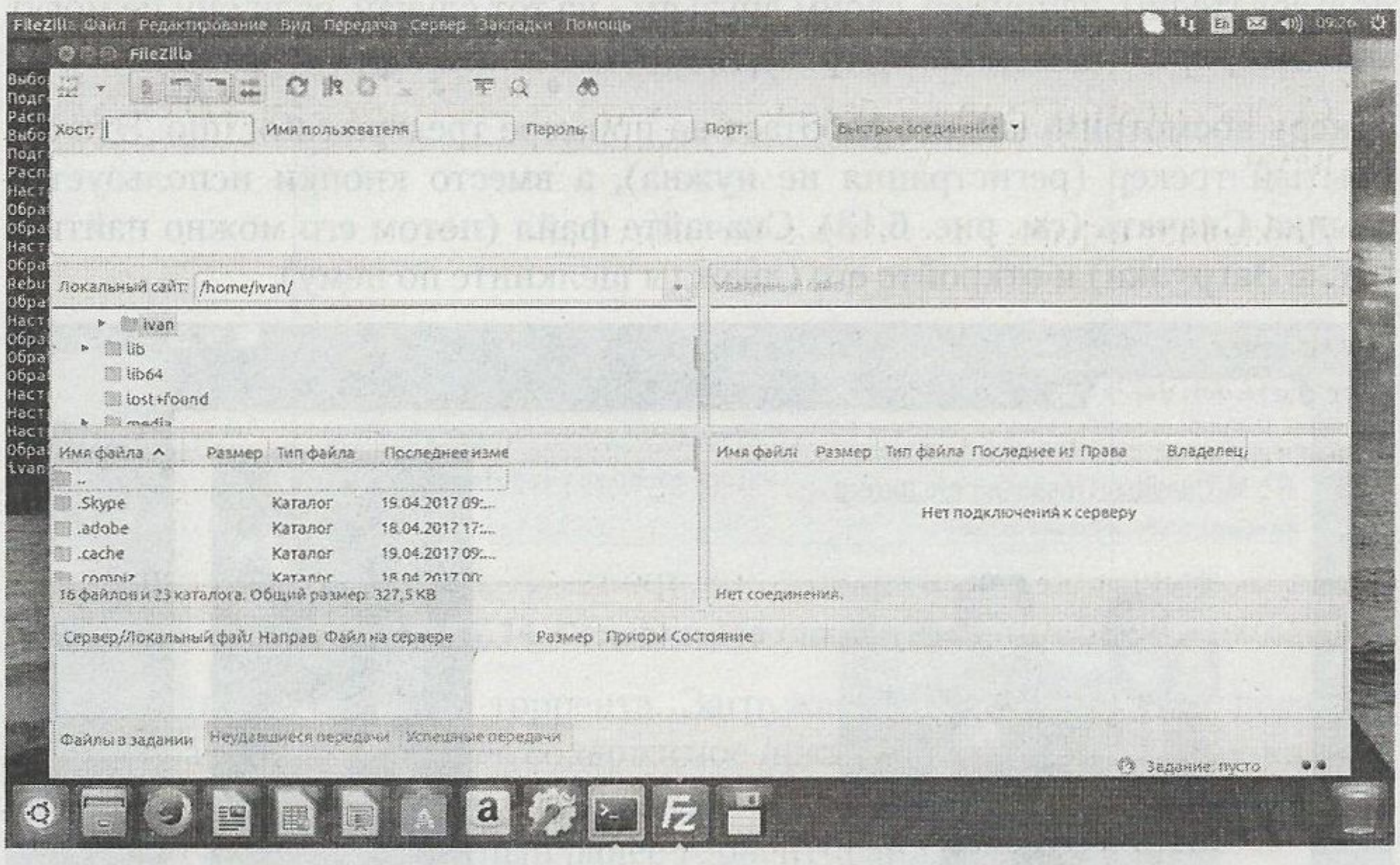


Рис. 6.12. FTP-клиент FileZilla

6.1.4. BitTorrent-клиент Transmission

Как проще всего скачать фильмы и музыку? Конечно же, через торренты. Не хочется углубляться в теорию, лучше расскажу, как это работает - для тех, кто не знает. А те, кто знаком с торрентами, могут смело пропустить

следующий абзац и перейти к рассмотрению того, как торренты работают в Linux.

Первым делом вам нужно найти сайт торрент-трекера, с помощью Google, думаю, это не проблема. Некоторые трекеры являются закрытыми (нужна регистрация), на некоторых же регистрация не нужна. Далее нужно выбрать интересующий вас контент, например, фильм. На страничке с контентом обязательно будет кнопка или ссылка загрузки торрент-файла, например, Скачать <название контента> или просто Скачать торрент. Вы поймете. При щелчке на ней, вы загрузите torrent-файл, который нужно открыть в торрент-клиенте. В Windows многие пользователи используют uTorrent, но его установка довольно проблематична в Linux, поэтому проще использовать Transmission, который, между прочим, установлен по умолчанию. Далее Transmission загрузит выбранный вами контент. Обратите внимание: торрент-файл имеет небольшой размер. В нем указано просто, на компьютерах каких пользователей нужно искать нужный вам контент. Торрент-файл не содержит сам контент. Но вы можете передать его другим пользователям, например, своим друзьям - на тот случай, если они не могут зарегистрироваться на сайте трекера, чтобы скачать торрент-файл.

Теперь посмотрим, как это работает на примере трекера rutor.info. Это открытый трекер (регистрация не нужна), а вместо кнопки используется ссылка **Скачать** (см. рис. 6.13). Скачайте файл (потом его можно найти в папке **Загрузки**) и откройте его (дважды щелкните по нему).

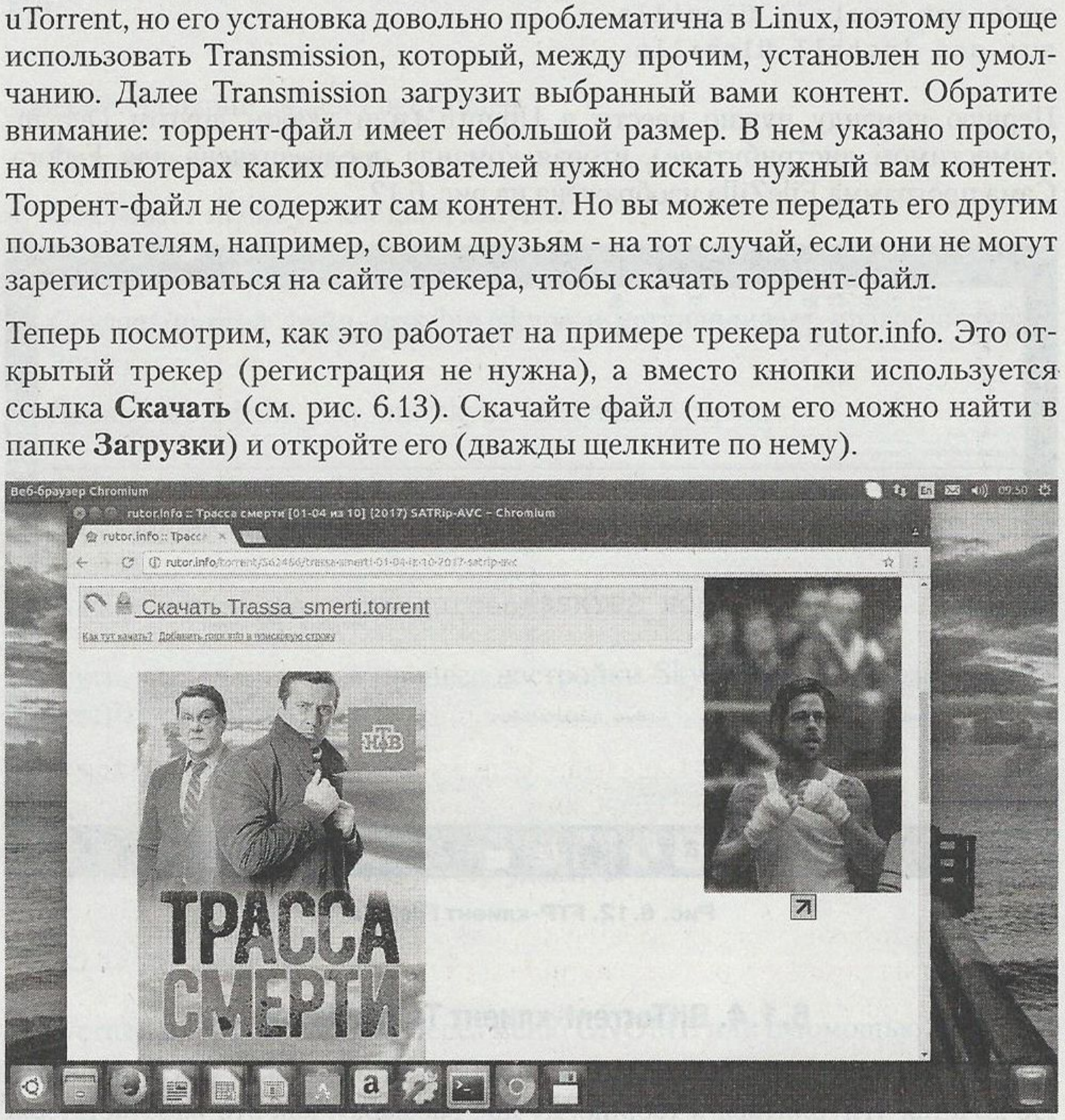


Рис. 6.13. Сайт торрент-трекера

При открытии торрент-файла клиент Transmission отобразит содержимое торрента. В нашем случае - список серий сериала - 4 видео-файла (рис. 6.14). При желании вы можете выбрать не все серии, а, например, только первую - вдруг сериал вам не понравится, и нет смысла загружать все серии. Сейчас объем небольшой - чуть больше 2 Гб, а бывают сериалы, которые занимают 20 Гб и больше - многое еще зависит от качества фильмов. Зачем ждать, пока будут загружены все 20 Гб, если можно загрузить первую серию, и если она вам не понравится, просто все удалить?

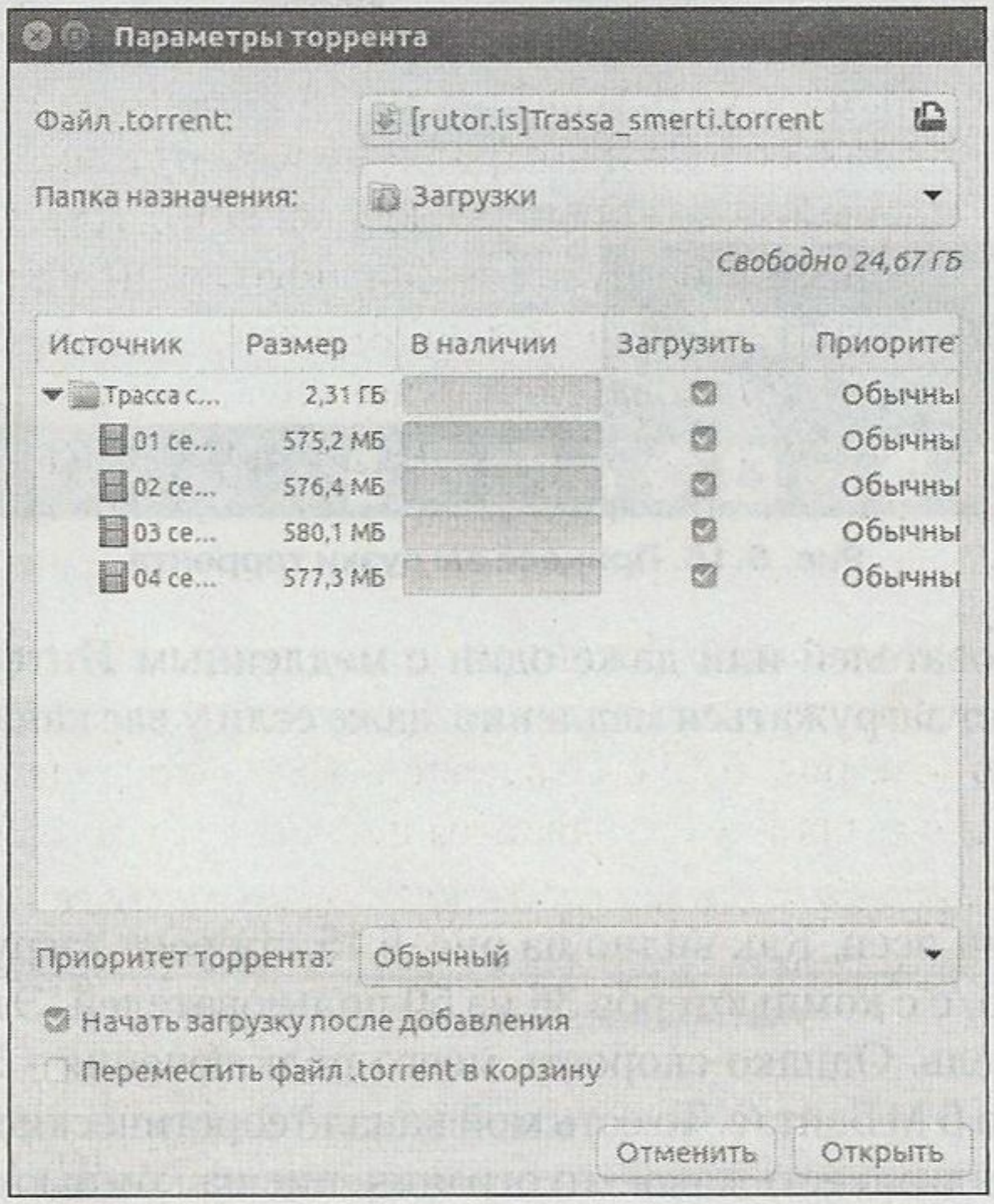


Рис. 6.14. Содержимое торрент-файла

Далее начнется загрузка торрента. Загружаемый контент будет помещен также в папку **Загрузки**, ее содержимое показано на рис. 6.15 - как видите, есть папка с загружаемым торрентом (ее название соответствует названию торрента) и сам загруженный ранее торрент-файл.

Теперь несколько слов о времени загрузки. Поскольку контент реально содержится на компьютерах других пользователей, то скорость загрузки зависит от скорости Интернета этих пользователей и от числа самих пользователей. Рассмотрим ситуации:

- Много пользователей, хорошая пропускная способность у каждого, хорошая пропускная способность у вас - контент будет загружен быстро.

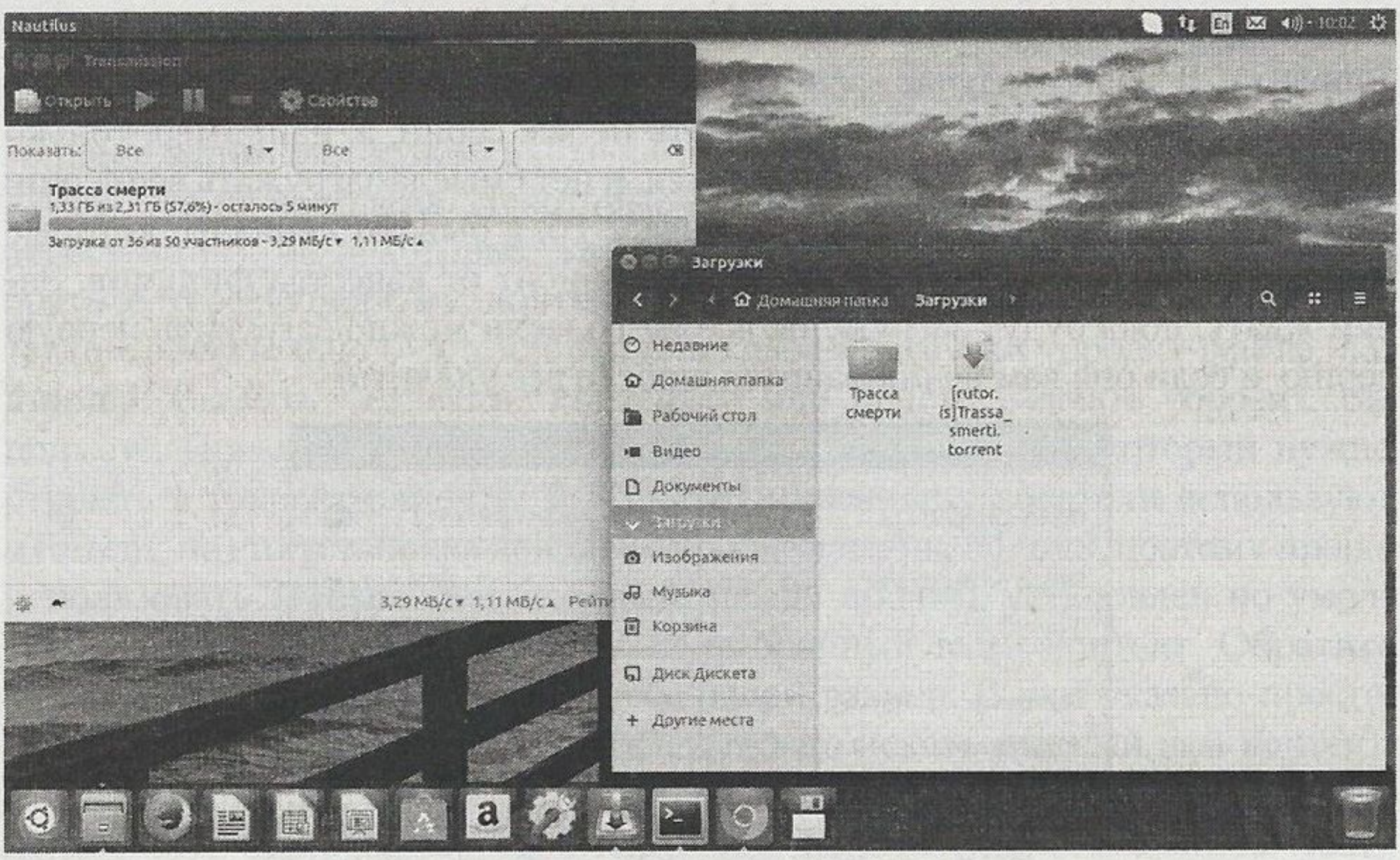


Рис. 6.15. Процесс загрузки торрента

- Мало пользователей или даже один с медленным Интернет-каналом - контент будет загружаться медленно, даже если у вас канал со скоростью 1 Гбит/с.

Думаю, принцип ясен. Как видно из рис. 6.15, торрент загружается со скоростью 3.29 МБ/с с компьютеров 36 из 50 пользователей. Это довольно хороший показатель. Однако скорость моего подключения - 50 Мбит/с, что примерно равно 6 МБайт/с. То есть мой канал теоретически может работать и быстрее, но видимо есть какое-то ограничение на компьютерах пользователей, где содержится контент.

Внимание! После загрузки контента ваш торрент-клиент будет «раздавать» этот контент другим пользователям, то есть вы станете одним из участников торрент-сети. Если у вас Интернет тарифицируется по трафику (то есть существует какое-то ограничение на объем передаваемой информации в месяц), лучше закрыть торрент-клиент или удалить торрент с раздачи, нажав кнопку - (красная) в торрент-клиенте.

6.2. Мультимедиа-приложения

В прошлом разделе мы загрузили некоторый мультимедиа-контент с помощью торрент-клиента (рис. 6.16). Теперь попытаемся его воспроизвести.

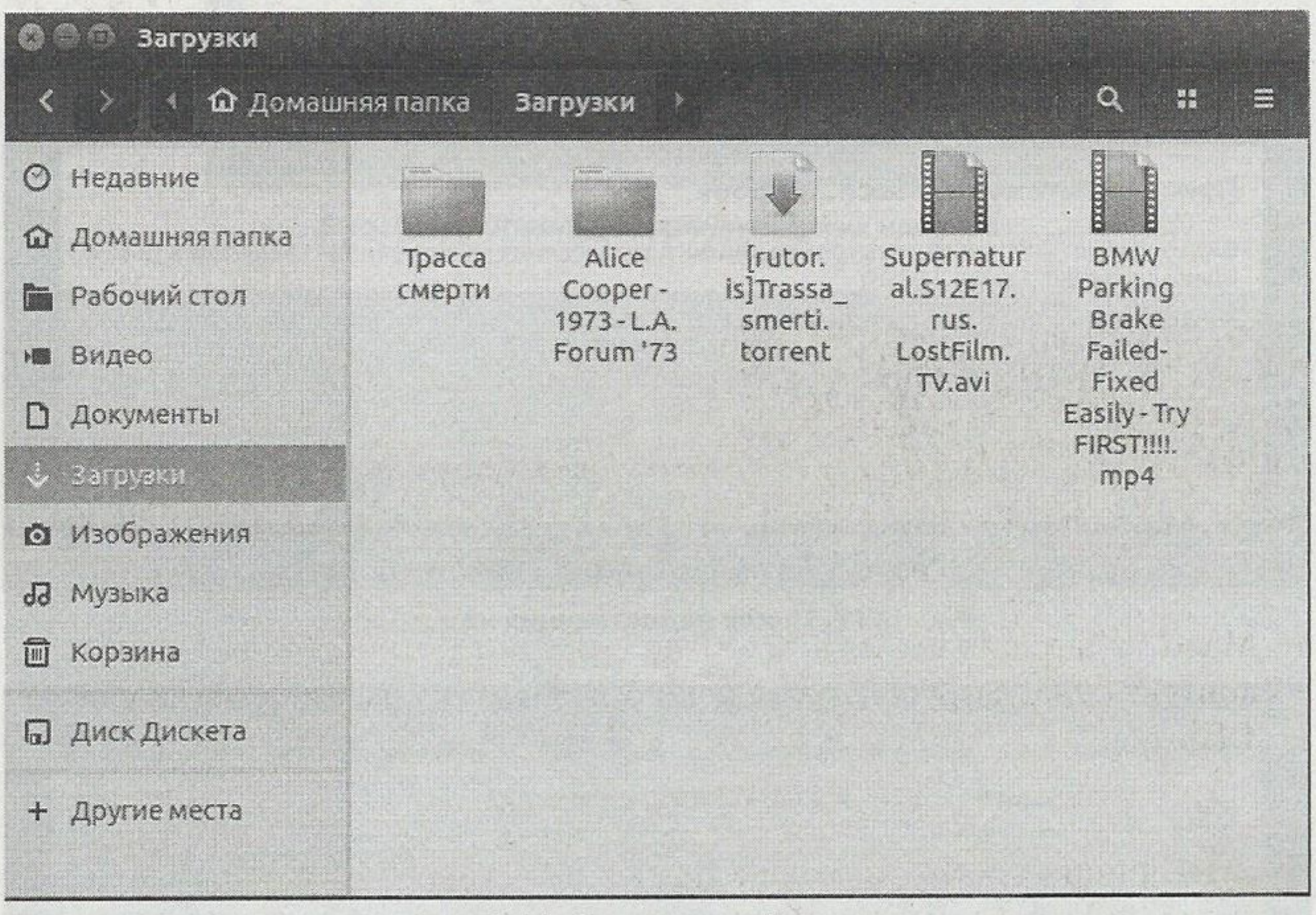


Рис. 6.16. Загруженный контент

Начнем с кодеков. В главе 2 было показано, как установить кодеки. Кодеки - это программа, способная декодировать контент (в нашем случае - музыку или видео) из формата, в котором этот контент закодирован. Существует множество различных мультимедиа-форматов, для каждого из них используется свой кодек. Если у вас не воспроизводится один фильм (или воспроизводится корректно, например, нет звука), а с воспроизведением другого фильма проблем нет, то проблема в кодеках. Просто не установлен кодек того формата, в котором закодирован файл фильма.

Поэтому начать нужно с установки кодеков. Если вы не сделали это, когда читали главу 2, самое время это сделать. Иначе вы не сможете воспроизвести ни музыку, ни видео. Исключение разве что составляет дистрибутив Fedora, в котором есть поддержка формата MP3 (хотя кроме этого формата есть еще множество других) «из коробки», то есть ничего не нужно устанавливать для поддержки MP3.

После установки кодеков можно попробовать воспроизвести фильм или музыку. Просто щелкните на файле фильма для его просмотра. При первом запуске проигрывателя **Видео**, даже не смотря на то, что вы установили кодеки в главе 2, вы можете увидеть сообщение о том, что некоторых надстроек не хватает. Приложение выполнит поиск недостающих надстроек (рис. 6.17) и предложит их установить (рис. 6.18).

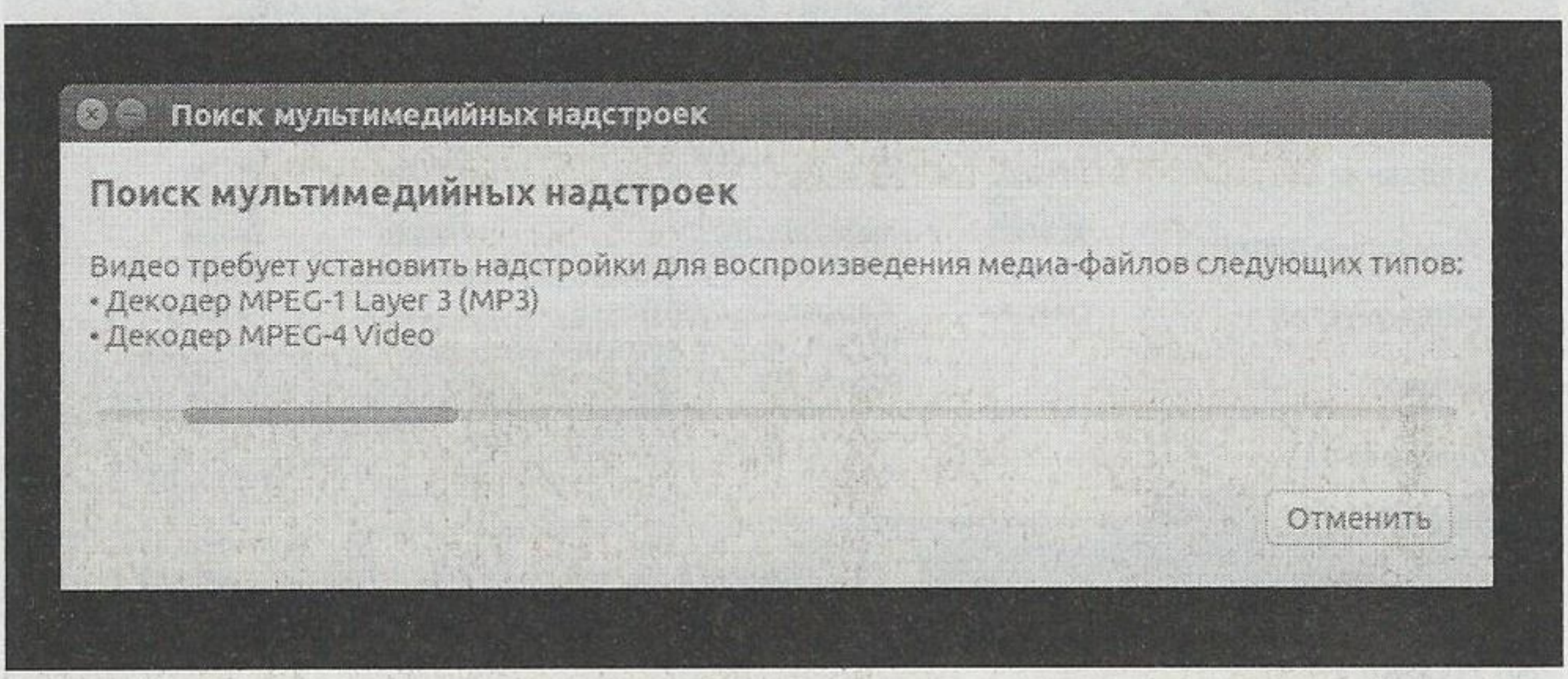


Рис. 6.17. Поиск недостающих надстроек

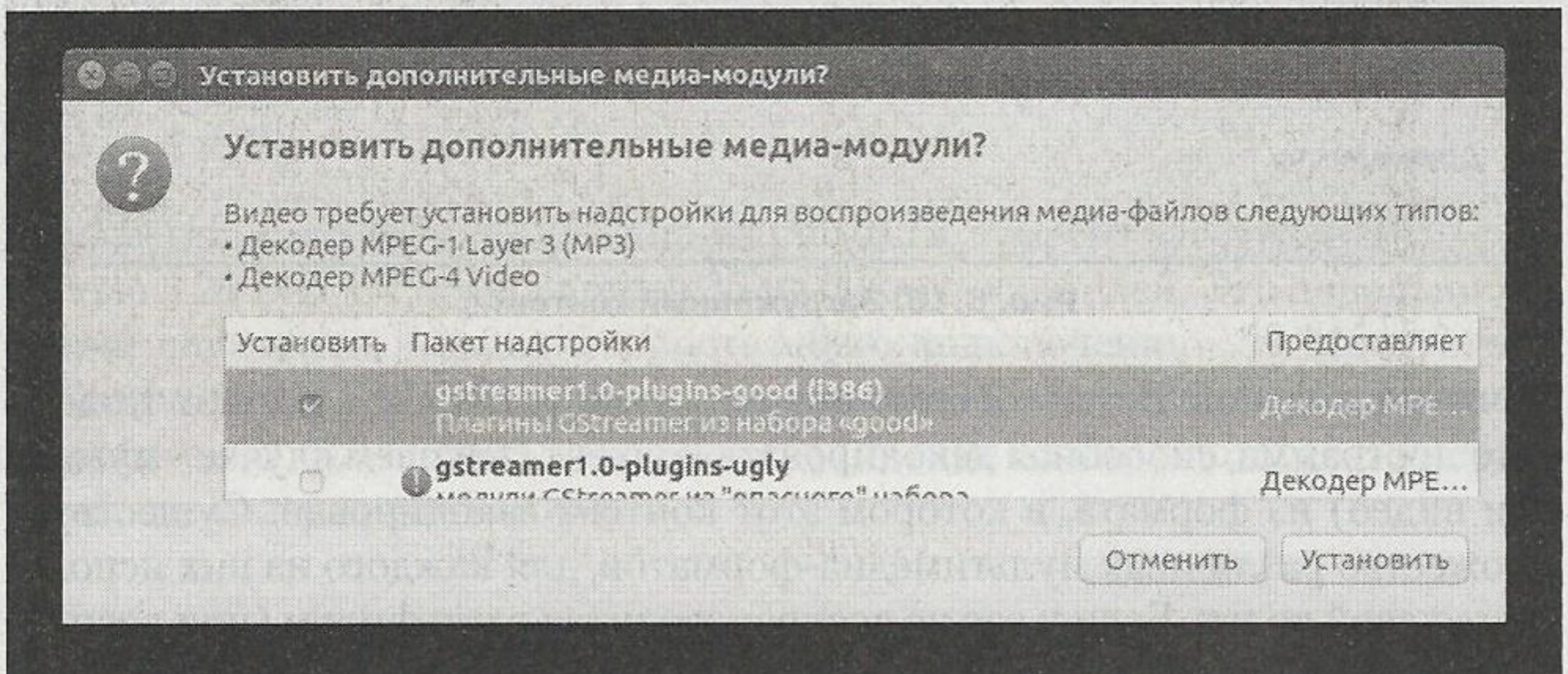


Рис. 6.18. Нажмите кнопку Установить

Далее вы увидите список устанавливаемых пакетов. Нажмите кнопку **Продолжить** (рис. 6.19). Введите свой пароль и дождитесь завершения установки. После того как пакеты будут установлены, перезапустите проигрыватель - просто закройте его и снова откройте видео-файл. Начнется воспроизведение (рис. 6.20).

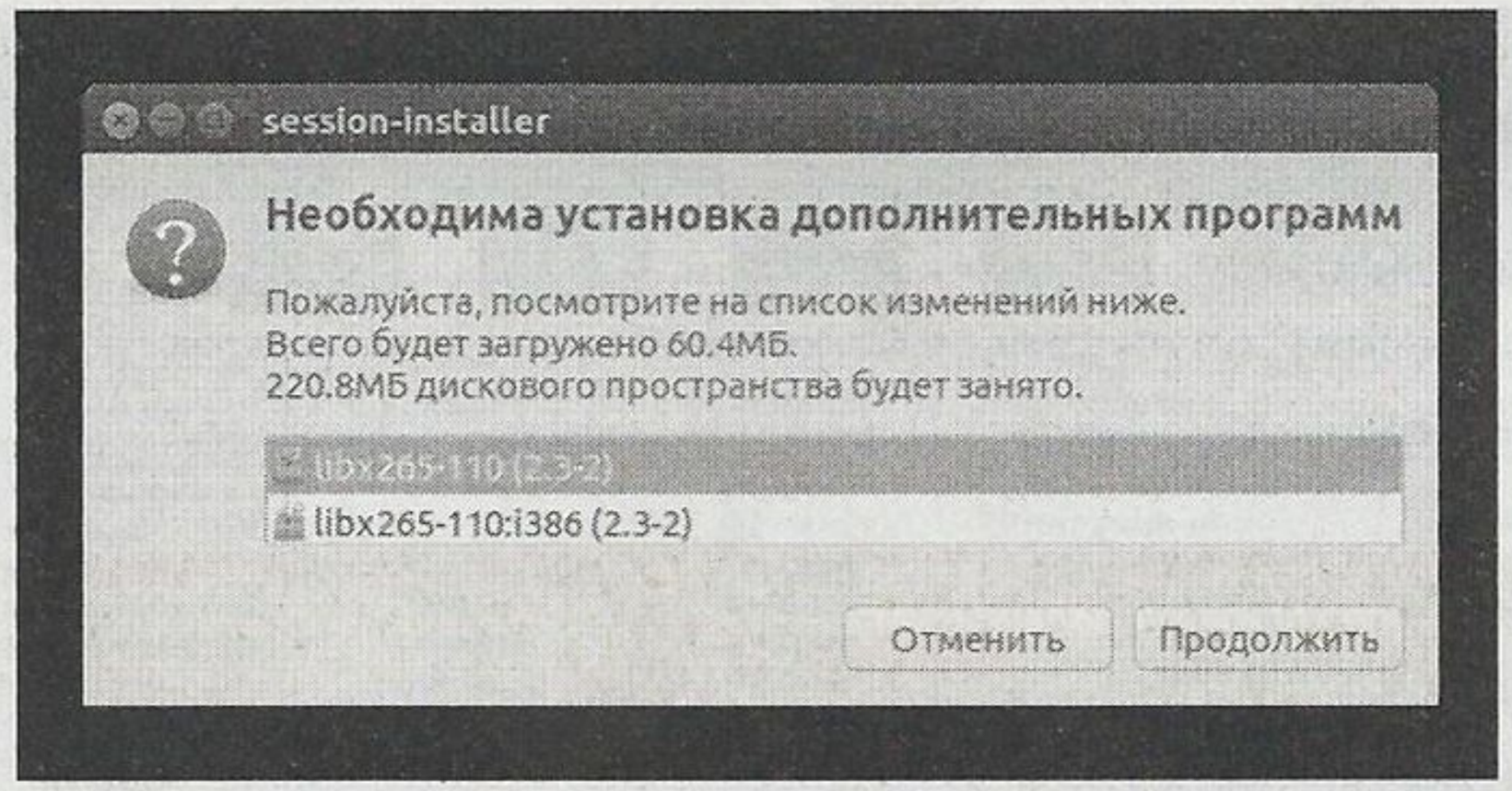


Рис. 6.19. Список устанавливаемых пакетов

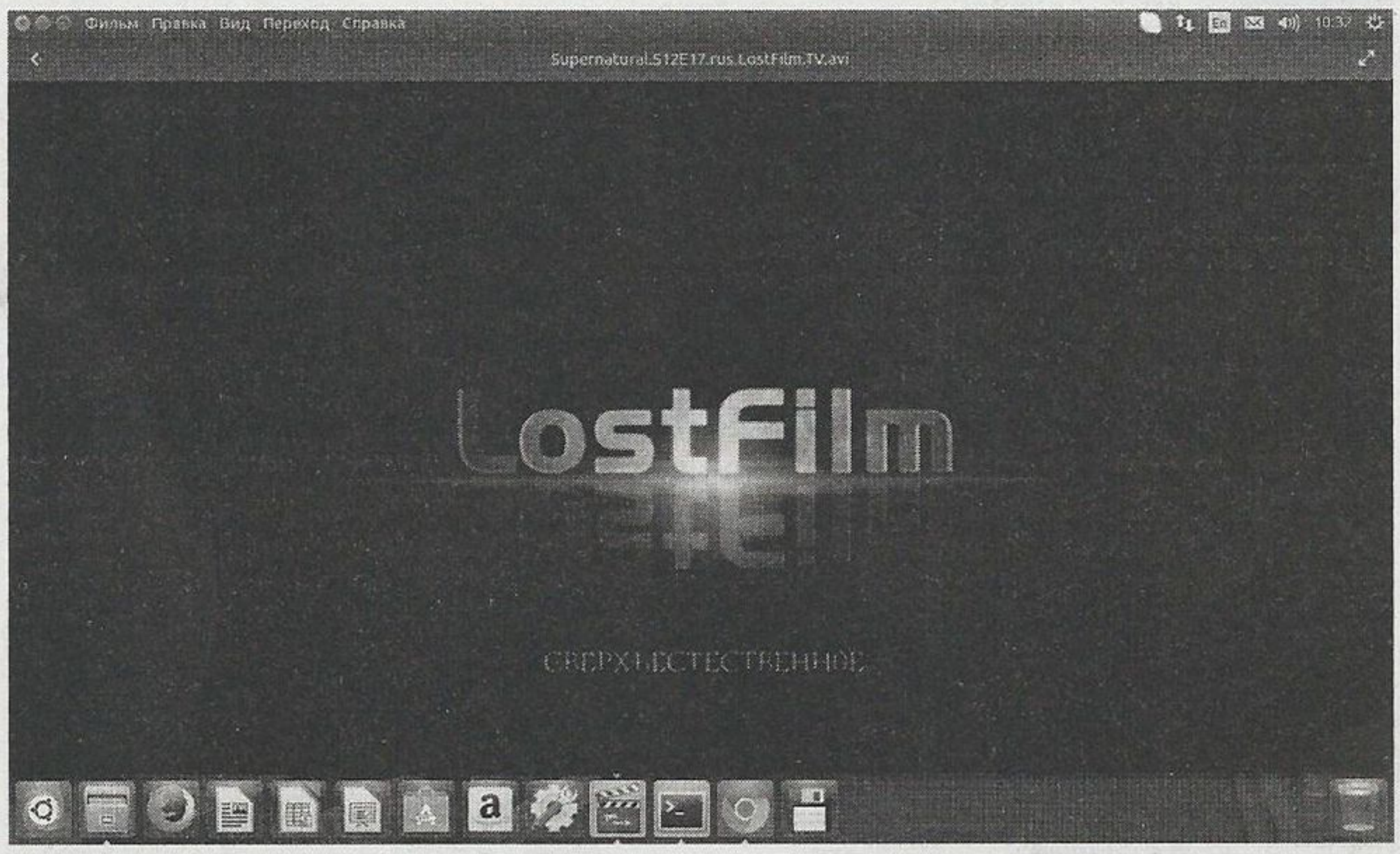


Рис. 6.20. Воспроизведение фильма

Также после установки дополнительных пакетов в файловом менеджере у вас появятся миниатюры видео-файлов (рис. 6.21) - если раньше отображались просто пиктограммы, то сейчас - кадры из фильмов.

С видео вроде бы все понятно. Теперь перейдем к музыке. Для воспроизведения музыки в современных дистрибутивах используется проигрыватель Rhythmbox, установленный по умолчанию. Запустите его. Проигрыватель ищет музыкальные файлы в папке Музыка, но он не найдет загруженные с торрентов композиции, поскольку мы загрузили их в папку Загрузки. Вы-

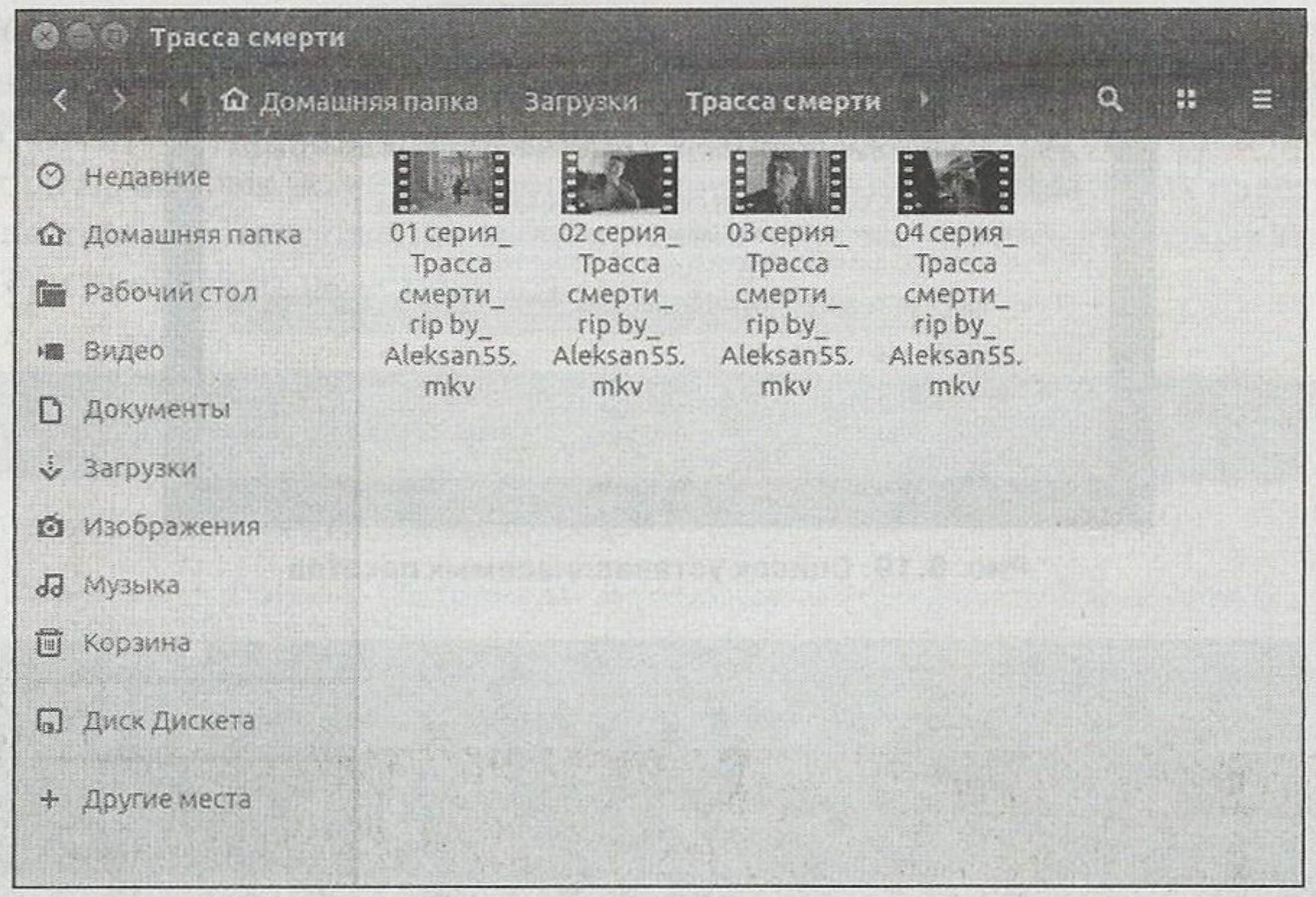


Рис. 6.21. Миниатюры фильмов в файловом менеджере

берите расположение музыкальных файлов, прежде чем приступить к воспроизведению (рис. 6.22). Можно также переместить загруженную музыку из папки Загрузки в папку Музыка, если вам так будет удобнее.

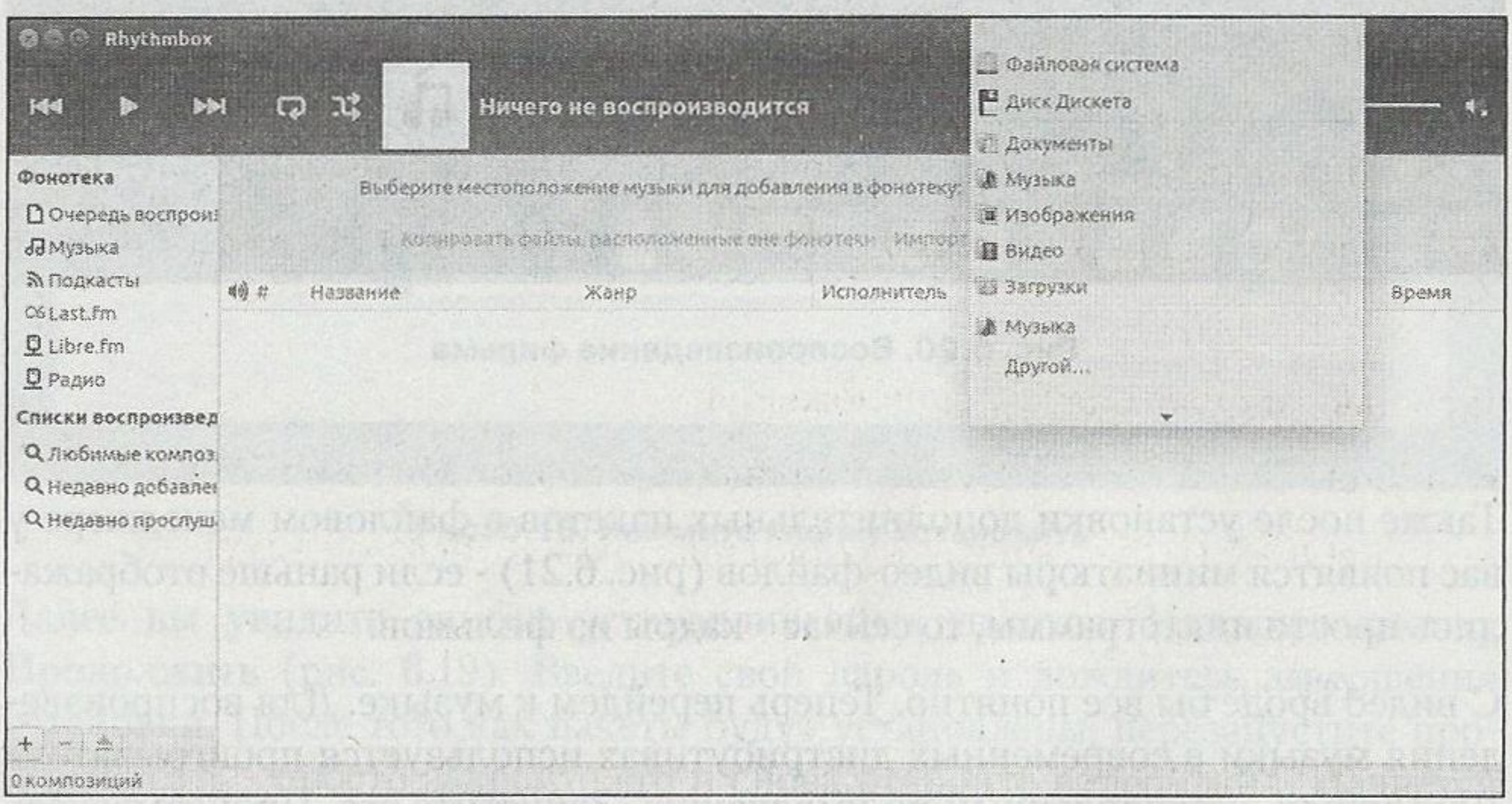
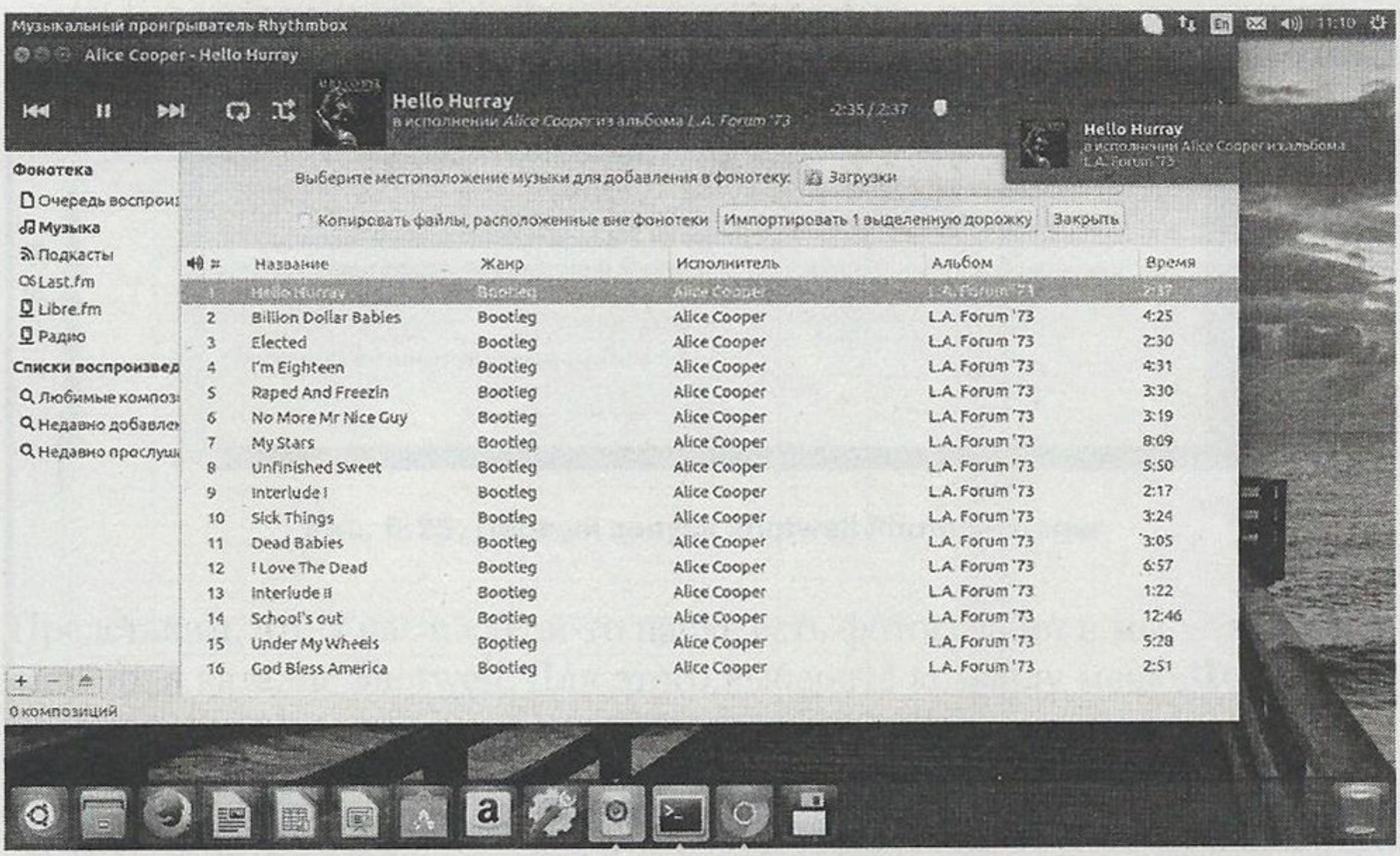


Рис. 6.22. Выбор расположения музыкальных файлов



#	Название	Жанр	Исполнитель	Альбом	Время
1	Hello Hurray	Bootleg	Alice Cooper	L.A. Forum '73	2:37
2	Billion Dollar Babies	Bootleg	Alice Cooper	L.A. Forum '73	4:25
3	Elected	Bootleg	Alice Cooper	L.A. Forum '73	2:30
4	I'm Eighteen	Bootleg	Alice Cooper	L.A. Forum '73	4:31
5	Raped And Freezin	Bootleg	Alice Cooper	L.A. Forum '73	3:30
6	No More Mr Nice Guy	Bootleg	Alice Cooper	L.A. Forum '73	3:19
7	My Stars	Bootleg	Alice Cooper	L.A. Forum '73	8:09
8	Unfinished Sweet	Bootleg	Alice Cooper	L.A. Forum '73	5:50
9	Interlude I	Bootleg	Alice Cooper	L.A. Forum '73	2:17
10	Sick Things	Bootleg	Alice Cooper	L.A. Forum '73	3:24
11	Dead Babies	Bootleg	Alice Cooper	L.A. Forum '73	3:05
12	I Love The Dead	Bootleg	Alice Cooper	L.A. Forum '73	6:57
13	Interlude II	Bootleg	Alice Cooper	L.A. Forum '73	1:22
14	School's out	Bootleg	Alice Cooper	L.A. Forum '73	12:46
15	Under My Wheels	Bootleg	Alice Cooper	L.A. Forum '73	5:28
16	God Bless America	Bootleg	Alice Cooper	L.A. Forum '73	2:51

Рис. 6.23. Проигрыватель Rhythmbox в действии

6.3. Офисные приложения

Во всех современных дистрибутивах установлен по умолчанию пакет офисных приложений LibreOffice. Преимущества его очевидны:

- Приятный и удобный интерфейс пользователя. Интерфейс LibreOffice мне нравится больше, чем интерфейс последних версий Microsoft Office.
- Поддержка форматов (в том числе самых последних) Microsoft Office

В состав пакета входят три приложения:

- LibreOffice Writer - текстовый процессор, аналог Microsoft Word.
- LibreOffice Calc - электронная таблица, аналог Microsoft Excel.
- LibreOffice Impress - средство создания презентаций, аналог Microsoft PowerPoint.

Подробно все эти приложения мы рассматривать не будем - думаю, вы с ними разберетесь самостоятельно в процесс работы.

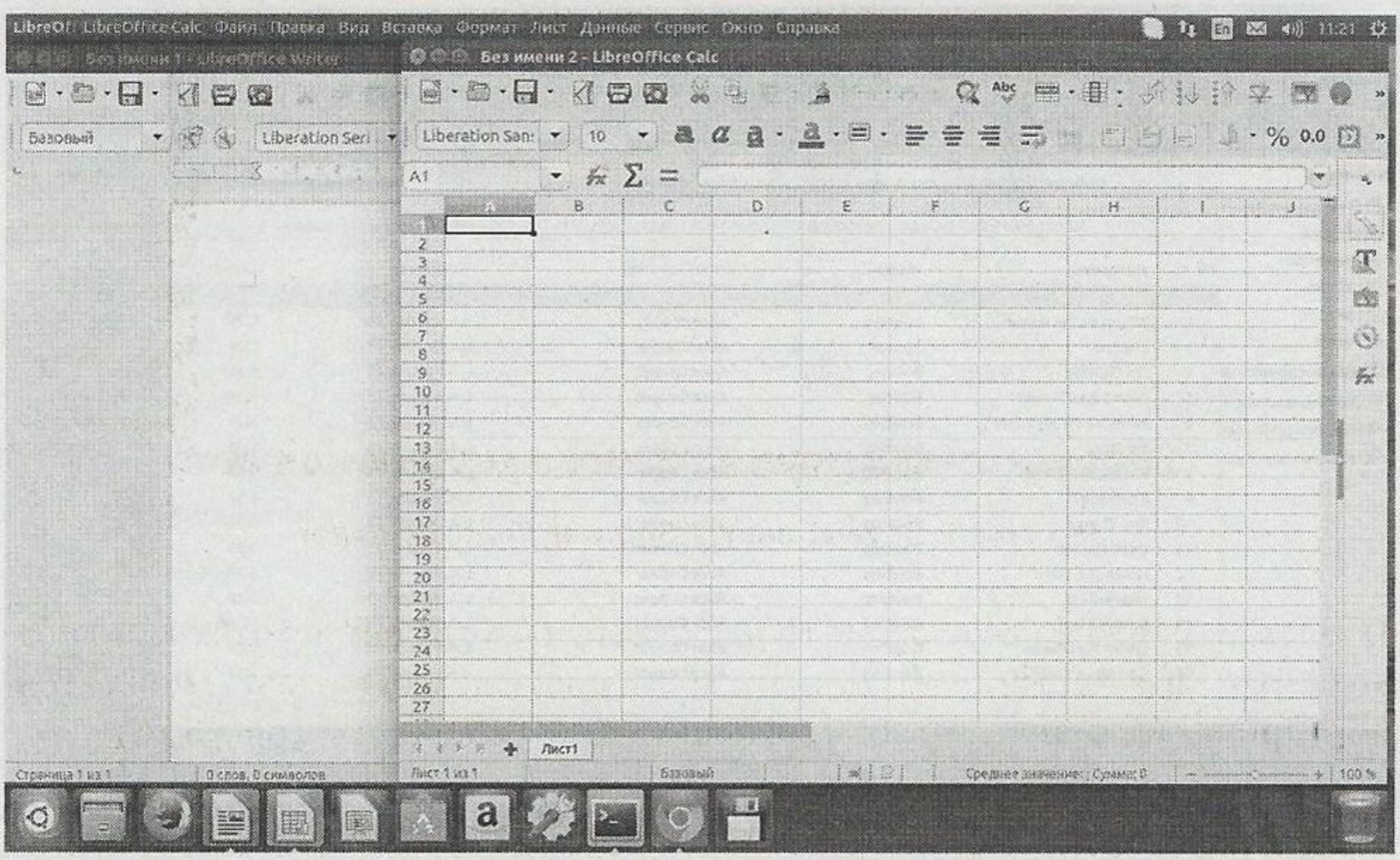


Рис. 6.24. Приложения Writer (на заднем плане) и Calc

6.4. Приложения для работы с графикой

В большинстве случаев для работы с графическими файлами нужно два приложения:

- Средство просмотра и каталогизации изображений.
- Средство редактирования изображений.

6.4.1. Shotwell Photo Manager

Довольно неплохой программой является Shotwell Photo Manager. Она используется для просмотра и каталогизации фотографий - как раз то, что нужно. При первом запуске программа покажет вам, что импортировать фото в каталог можно тремя способами:

- С помощью команды **Файл, Импортировать из папки.**
- Путем перетаскивания фото в окно Shotwell.
- Путем подключения камеры к компьютеру и импорта фото из нее.

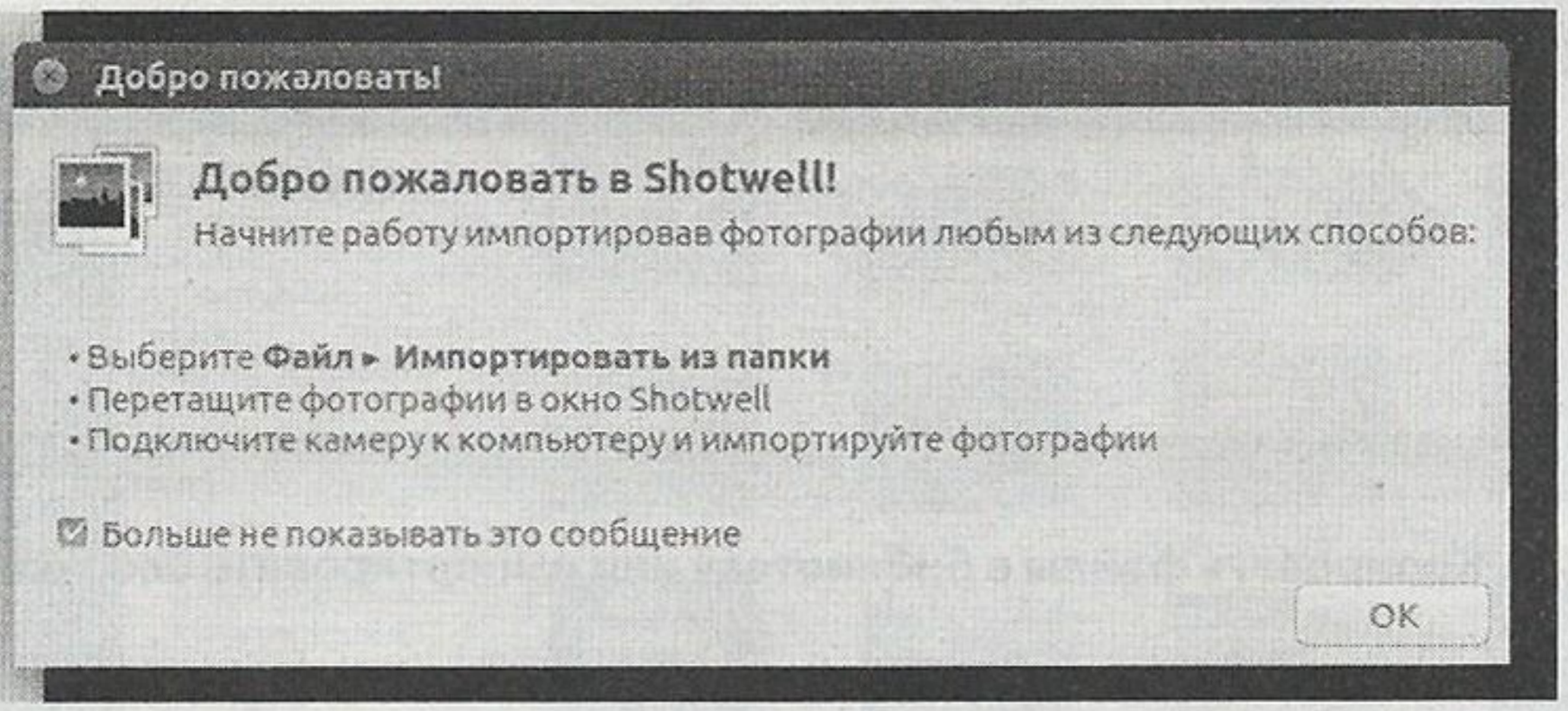


Рис. 6.25. Первый запуск Shotwell Photo Manager

Представим, что у нас в какой-то папке есть фотографии и мы хотим добавить их в каталог Shotwell. Для этого выберите команду меню **Файл, Импортировать из папки** и выберите папку с изображениями (рис. 6.26).

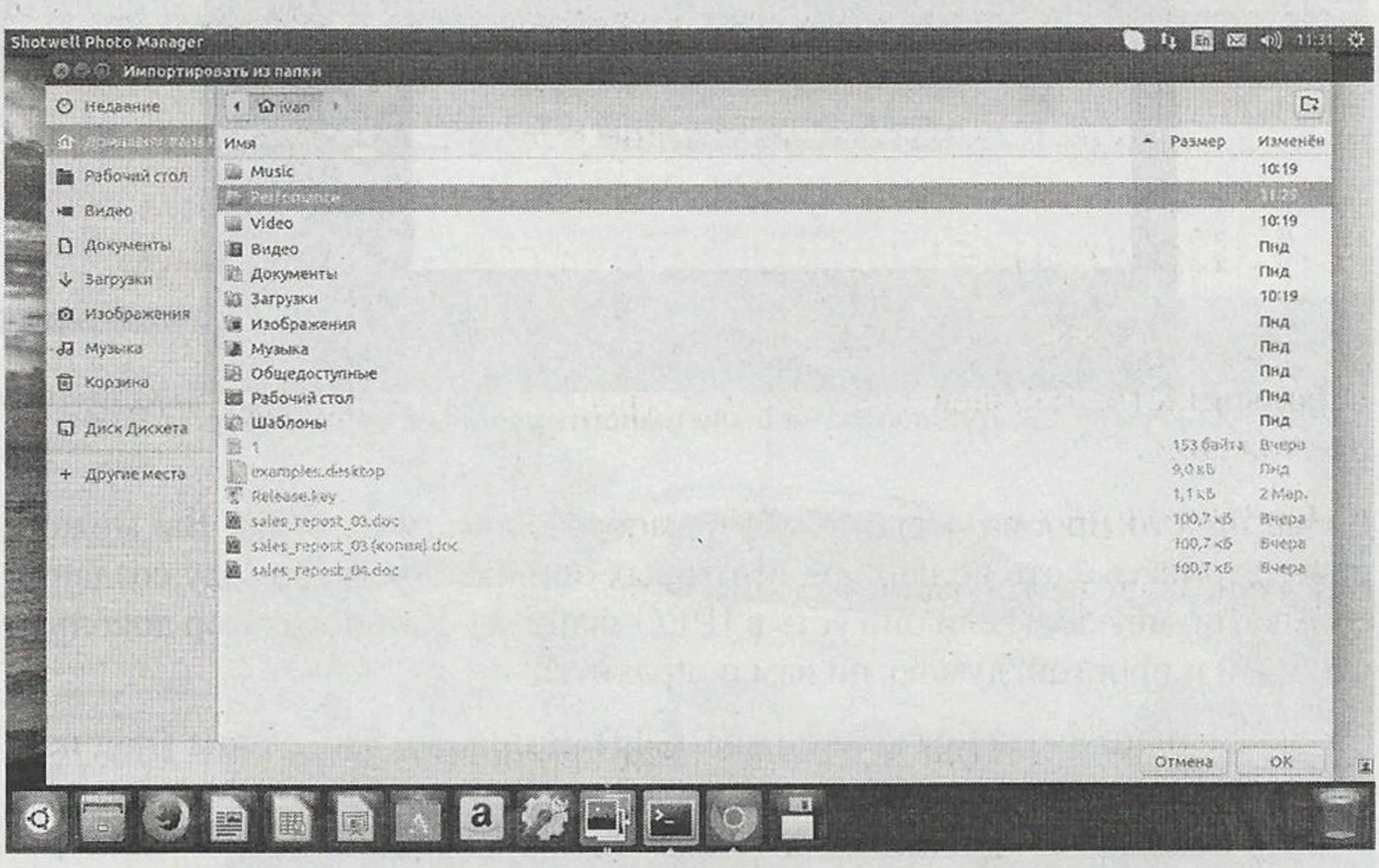


Рис. 6.26. Импорт изображений из папки

Как любой другой приличный каталогизатор, Shotwell спросит вас, хотите ли вы копировать фото в папку библиотеки или импортировать без копирования. В первом случае вы сможете безболезненно обращаться с файлами (программа поддерживает базовые функции редактирования), не боясь, что оригиналы будут повреждены. Во втором - вы сэкономите место на диске. Я предпочитаю второй вариант, так как места на моем SSD-диске совсем немного. Поэтому я нажал кнопку **Импорт в.**

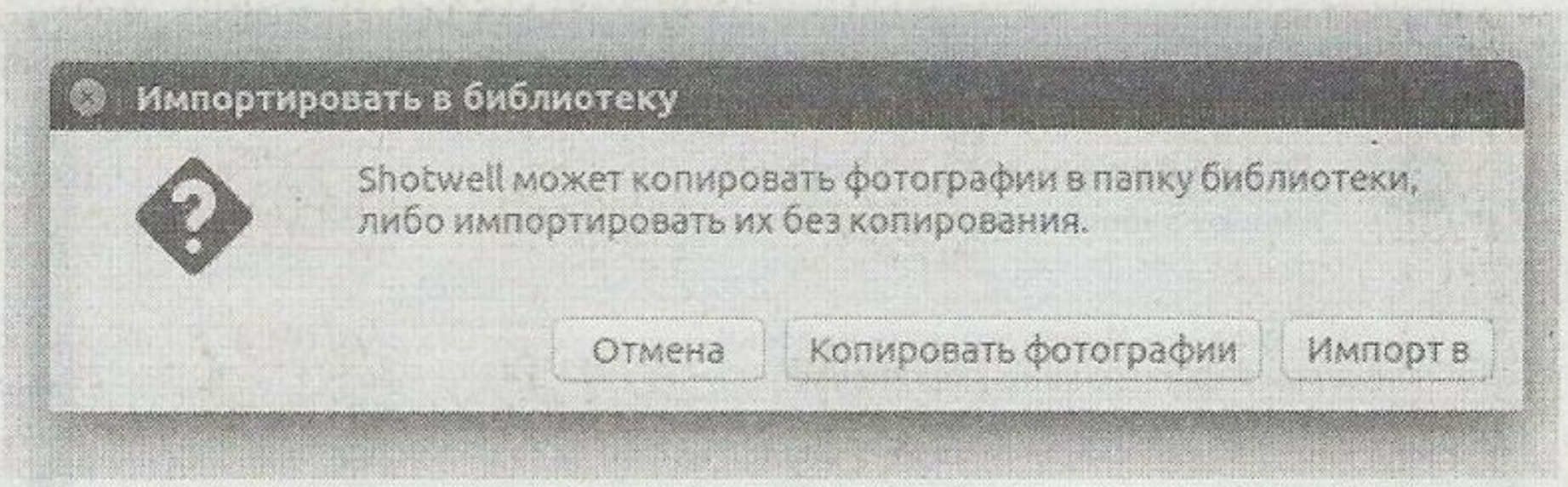


Рис. 6.27. Копировать файлы в библиотеку или импортировать без копирования

Примечательно, но программа успешно справилась с обработкой дубликатов фото - дубликаты не был импортированы в библиотеку (рис. 6.28).

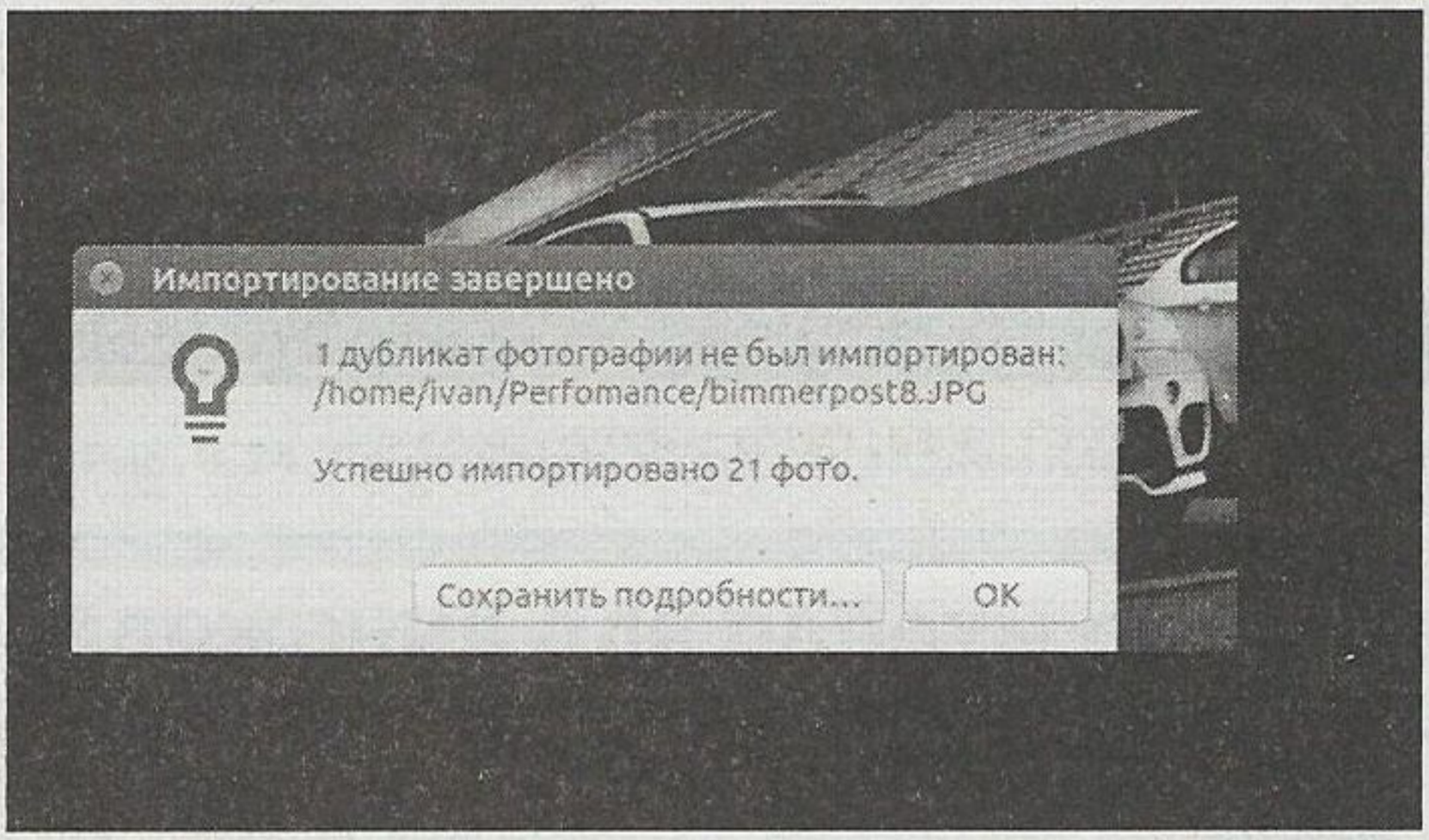


Рис. 6.28. Дубликаты не были импортированы в библиотеку

Далее просто просматривайте ваши изображения (рис. 6.29). Вы можете просматривать фото по папкам, в которых они находятся, по году создания снимка, по меткам (если они есть в JPEG-файлах). Каталогизатор довольно удобный и простой, думаю, он вам понравится.

Откройте одно из изображений (рис. 6.30). В нижней части окна (под изображением) вы найдете функции базового редактирования - поворот, подрезка (crop), выравнивание (позволяет выровнять фото с заваленным горизонтом), удаление эффекта красных глаз и функции коррекции фото (рис. 6.31).

Если функций редактирования фото вам будет недостаточно, тогда нужно установить полноценный графический редактор - GIMP. Его называют аналогом Photoshop для Linux. Пусть сравнение с Photoshop и не совсем хорошее, но GIMP - это очень мощный графический редактор. Вот только по умолчанию он не устанавливается. Впрочем ничего удивительного - при установке Windows графический редактор уровня Photoshop тоже не уста-

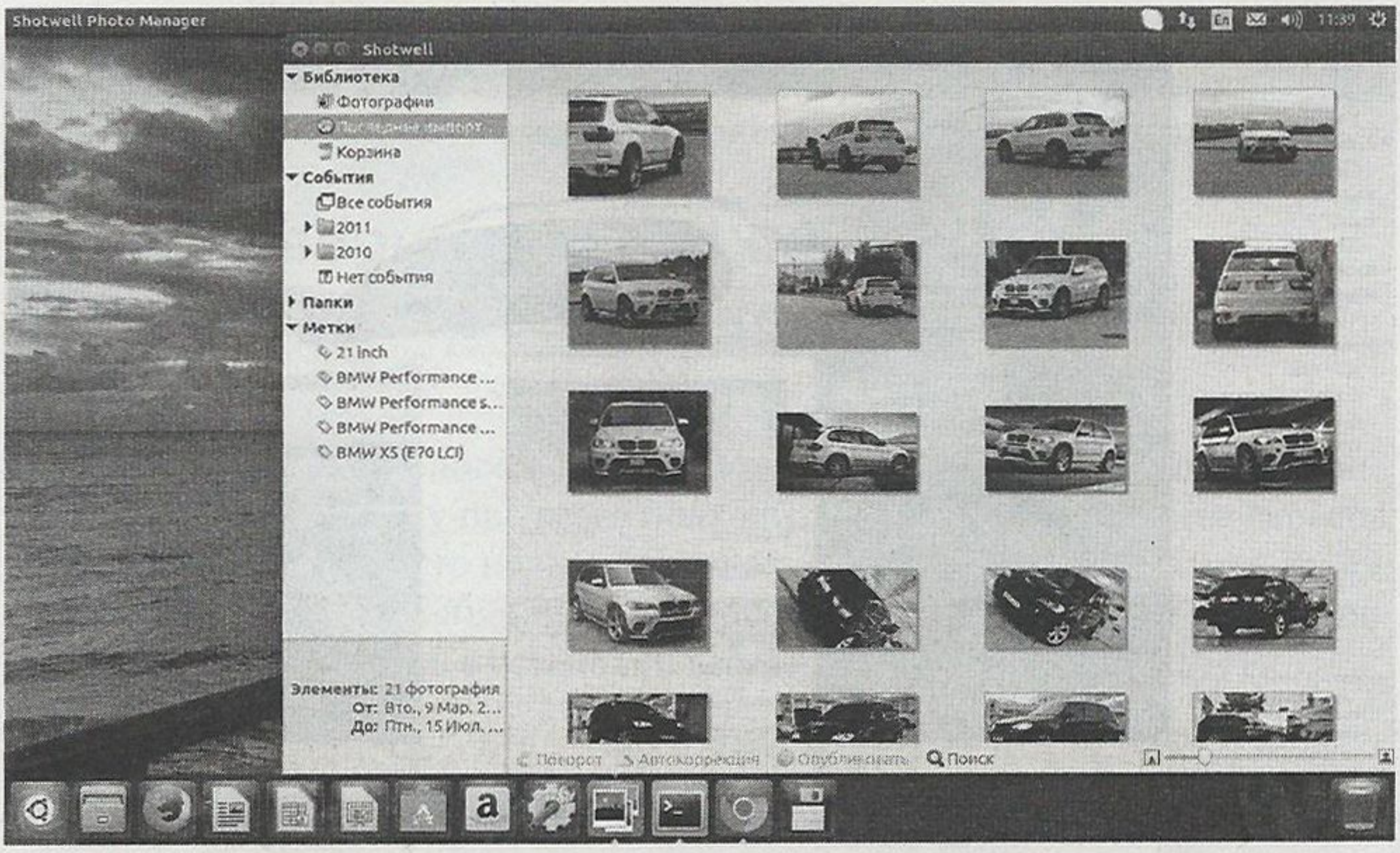


Рис. 6.29. Фото импортированы

Рис. 6.30. Просмотр фото

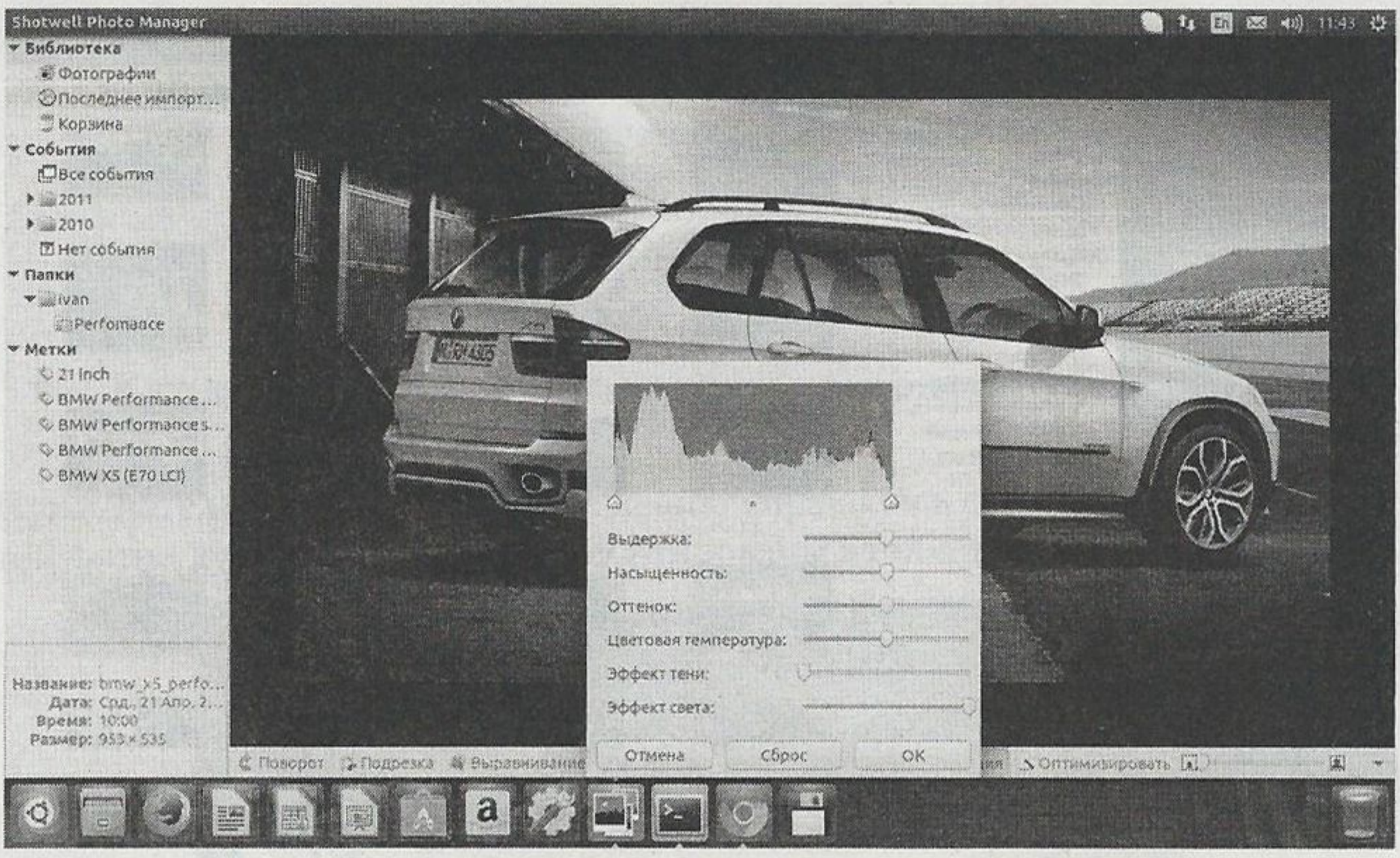


Рис. 6.31. Коррекция фото

навливается. Поэтому его придется устанавливать вручную. Как обычно, первая команда - для Ubuntu, вторая - для Fedora:

```
sudo apt install gimp
su; dnf install gimp
```

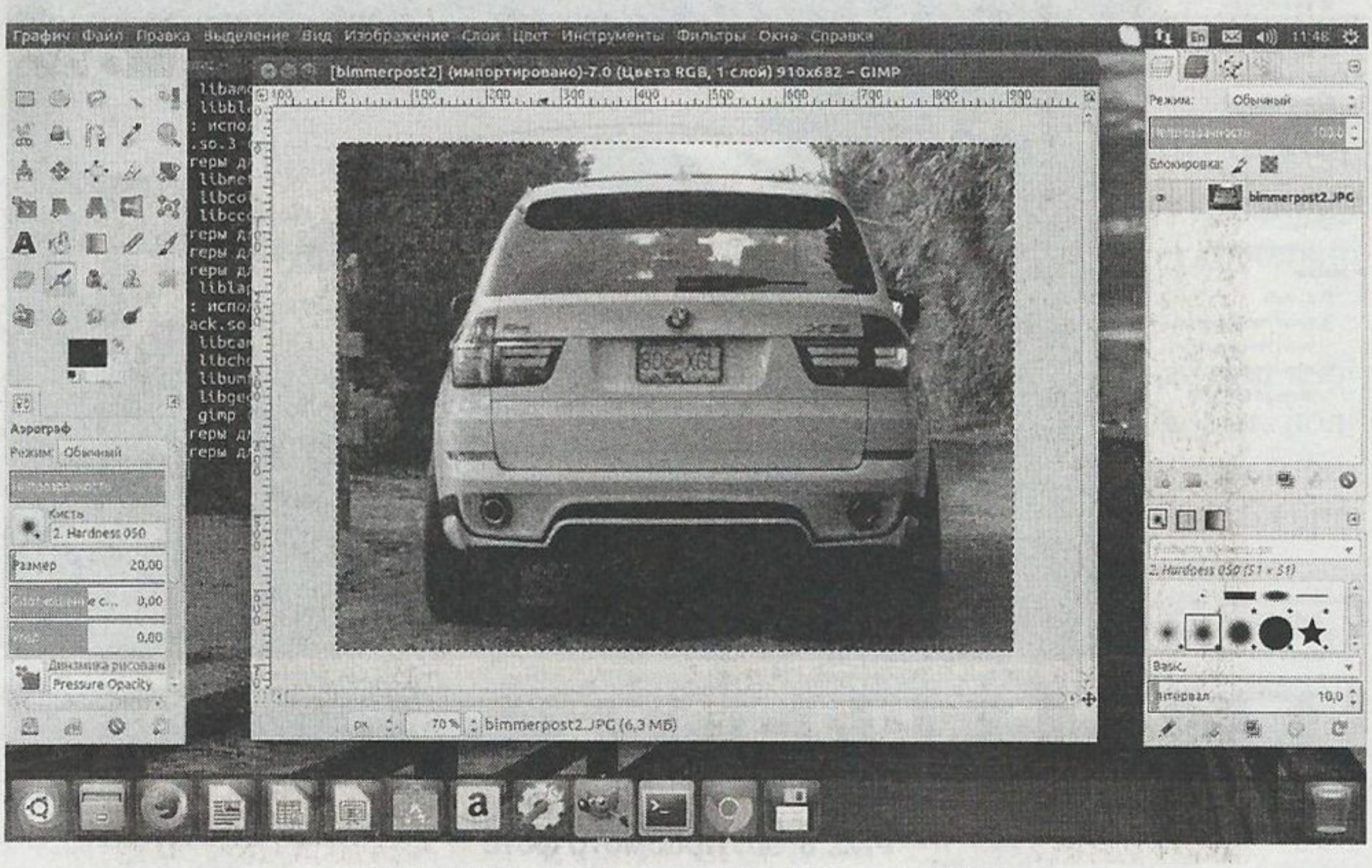


Рис. 6.32. Графический редактор GIMP

GIMP - это довольно серьезная программа, заслуживающая написания отдельной книги. В Интернете вы найдете множество сайтов и просто статей, посвященных этой программе, если, конечно, она вас заинтересовала.

6.5. Серверные приложения. Собственный веб-сервер

Раньше было сказано довольно много о серверах, серверных приложениях и о том, что Linux лучше всего подходит для работы в качестве сервера. После всего этого просто нельзя было не рассказать, как превратить любой компьютер под управлением Linux в сервер. Но серверы бывают разные, поэтому предлагаю начать с установки веб-сервера с поддержкой PHP.

Что для этого нужно сделать? Просто установить пакеты apache2 и php:

```
sudo apt install apache2 php
```

В результате будут установлены некоторые дополнительные пакеты, а именно:

- php7.0 - собственно, сам PHP, версия 7.0
- php7.0-cli - поддержка запуска PHP из командной строки
- libapache2-mod-php7.0 - модуль интеграции PHP с веб-сервером Apache 2.

Полный список устанавливаемых пакетов изображен на рис. 6.33.

```
ivan@ivan-home: ~
ivan@ivan-home:~$ sudo apt install apache2 php
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Будут установлены следующие дополнительные пакеты:
  apache2-bin apache2-data apache2-utils libapache2-mod-php7.0 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php-common
  php7.0 php7.0-cli php7.0-common php7.0-json php7.0-opcache php7.0-readline
Предлагаемые пакеты:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom php-pear
НОВЫЕ пакеты, которые будут установлены:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-php7.0 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0 php
  php-common php7.0 php7.0-cli php7.0-common php7.0-json php7.0-opcache
  php7.0-readline
обновлено 0, установлено 18 новых пакетов, для удаления отмечено 0 пакетов, и 0
пакетов не обновлено.
Необходимо скачать 5.057 kБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 20,6 М
В.
Хотите продолжить? [Д/н]
```

Рис. 6.33. Установка всех необходимых пакетов

Конечно, опытные разработчики PHP захотят доустановить расширения PHP (библиотеки), обеспечивающие тот или иной функционал. Как минимум, нужно доустановить следующие пакеты:

- php-curl - популярное расширение cURL;
- php-gd - графическая библиотека GD, добавляет функции для работы с графикой;
- php-imap - модуль для работы с протоколом IMAP;
- php-mail - класс, предоставляющий различные интерфейсы для отправки электронной почты;
- php-mysql - модуль для работы с БД MySQL;
- mysql-server - система управления базой данных MySQL;
- mysql-client - клиент для работы с MySQL.

Все эти пакеты (как и многие другие, содержащиеся в репозитарии и имеющие отношение к PHP) можно установить по мере необходимости. А мы приступим к дальнейшей настройке сервера.

По сути, если вы создаете не production-сервер, а сервер для собственного использования, например, если вы хотите научиться программированию на PHP, то настройка на этом заканчивается и можно запустить сервер командой:

```
sudo service apache2 start
```

Затем откройте браузер и введите адрес http://localhost. Вы должны увидеть Apache2 Ubuntu Default Page (см. рис. 6.34).

Теперь проверим, работает ли PHP. Перейдите в каталог /var/www/html. Это корневой каталог документов. Если вы поместите туда файл test.php, то он будет доступен по адресу http://localhost/test.php.

Введите команды:

```
cd /var/www/html
sudo touch test.php
sudo gedit test.php
```

Первая команда сменяет каталог, вторая - создает файл test.php, третья - запускает редактор для редактирования только что созданного файла. Введите следующие строки:

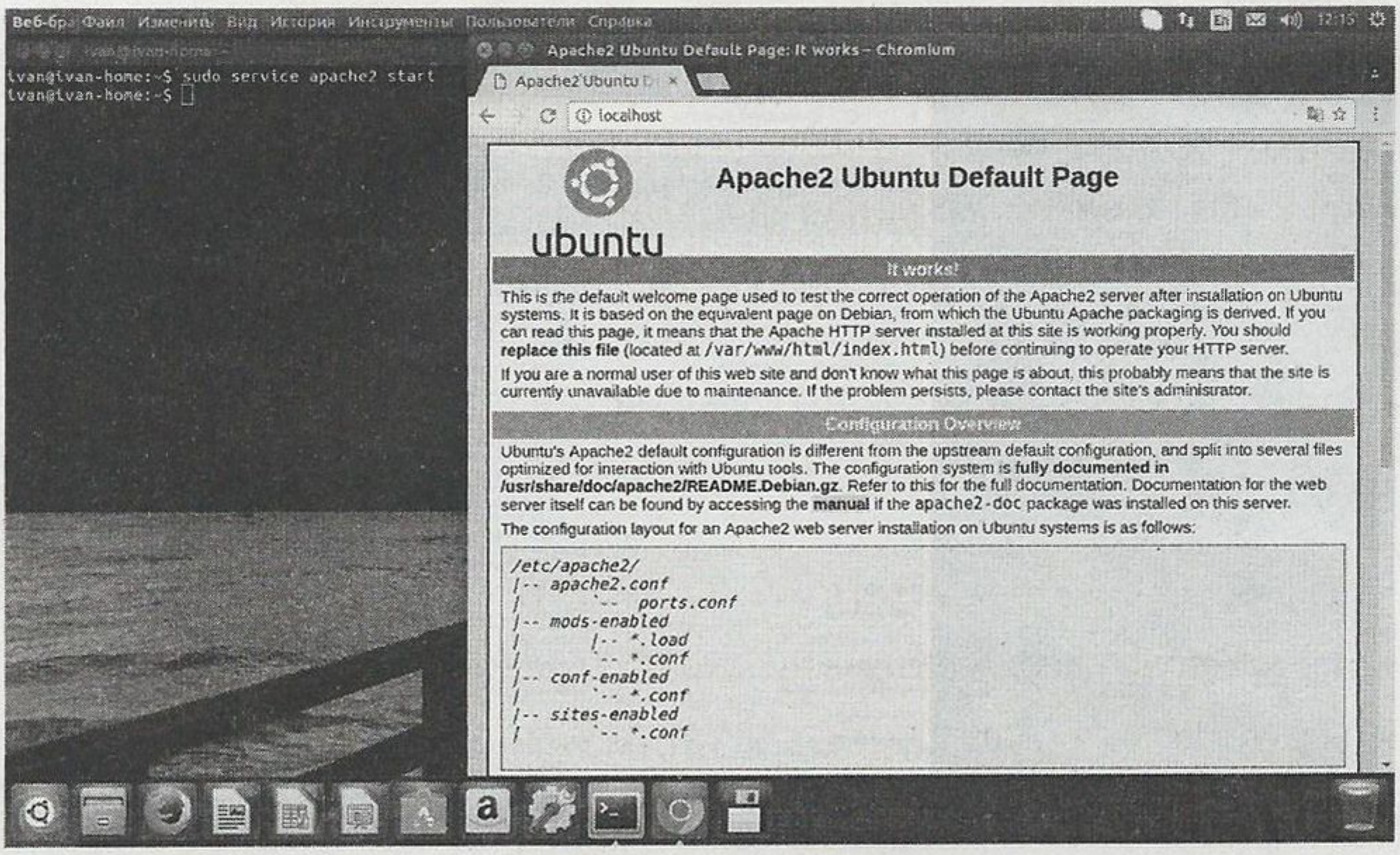


Рис. 6.34. Стандартная страница Apache

```
<?php
phpinfo();
?>
```

Сохраните файл и закройте редактор. Перейдите к браузеру и введите адрес http://localhost/test.php. Вы увидите страничку с информацией о версии PHP, установленных модулях PHP и других настройках (рис. 6.35). Эта страничка свидетельствует о том, что теперь у вас настроен веб-сервер с поддержкой PHP - на вашем локальном компьютере!

Можно также дополнить наш веб-сервер FTP-сервером. Для этого просто установите пакет proftpd:

```
sudo apt install proftpd
sudo service proftpd start
```

После этого вы сможете подключиться к своему компьютеру с любого другого компьютера вашей локальной сети, используя его IP-адрес, ваши имя пользователя и пароль. Посмотрите на рис. 6.36. На нем видно, что я с Windows-компьютера подключился к своему ноутбуку с IP-адресом 192.168.52.214, используя свое имя пользователя ivan и свой пароль. Видно и содержимое моего домашнего каталога, в который совсем недавно был загружен медиа-контент, о котором ранее шла речь.

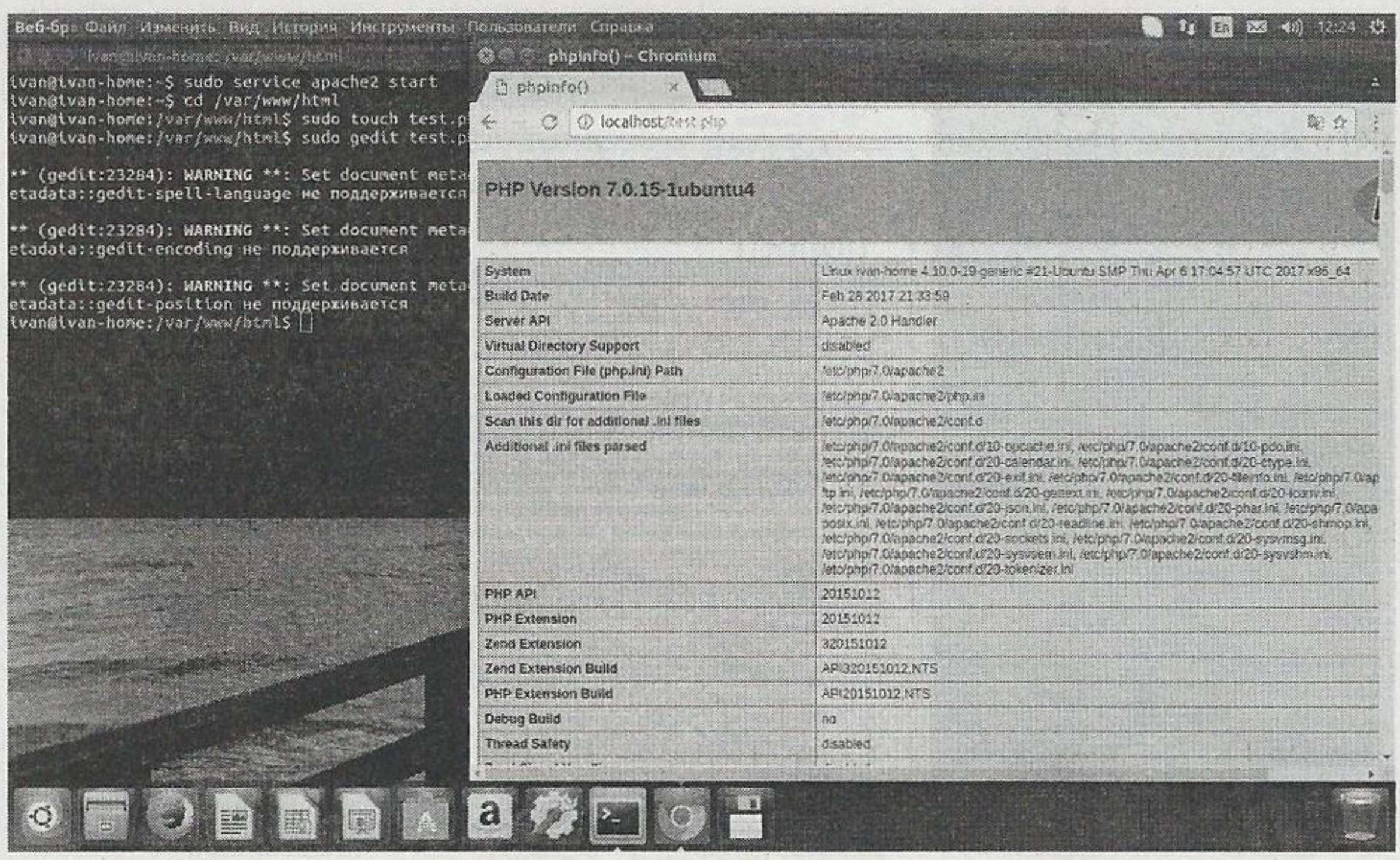


Рис. 6.35. Все готово!

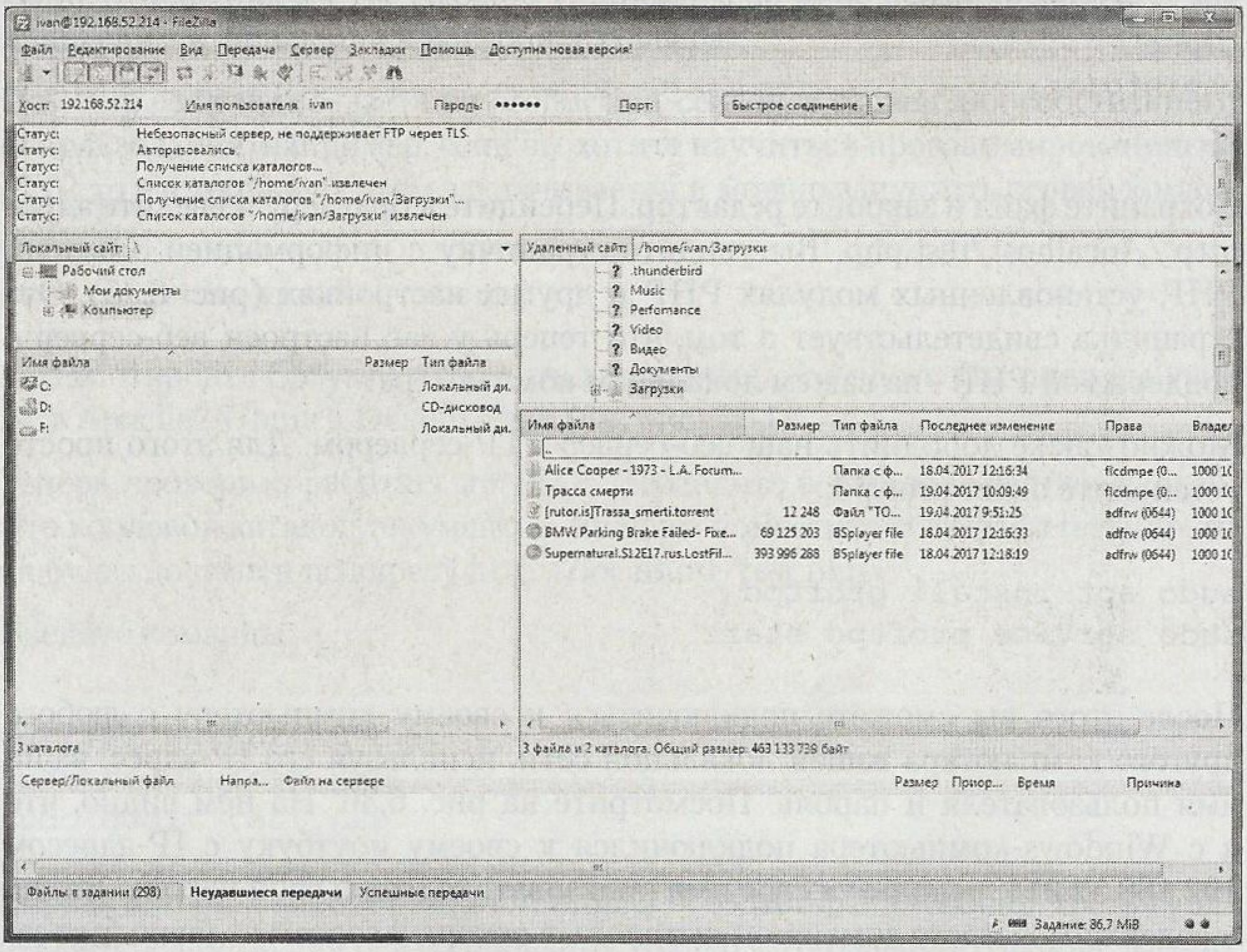


Рис. 6.36. FTP-сервер работает!

Имейте в виду, что конфигурация FTP-сервера по умолчанию очень опасна. Обычно пользователь не имеет право выходить за пределы своего домашнего каталога. Но конфигурация по умолчанию позволяет ему сделать это и просмотреть содержимое корневой файловой системы. Если ваш сервер будет использоваться только вами в локальной сети, проблем никаких нет. Но в реальных условиях нужно как минимум открыть файл /etc/proftpd/proftpd.conf и раскомментировать директиву:

```
DefaultRoot ~
```

После этого перезапустите сервер:

```
sudo service proftpd restart
```

Теперь пользователи не смогут выйти за пределы своего домашнего каталога, и конфигурационные файлы системы станут для них недоступными.

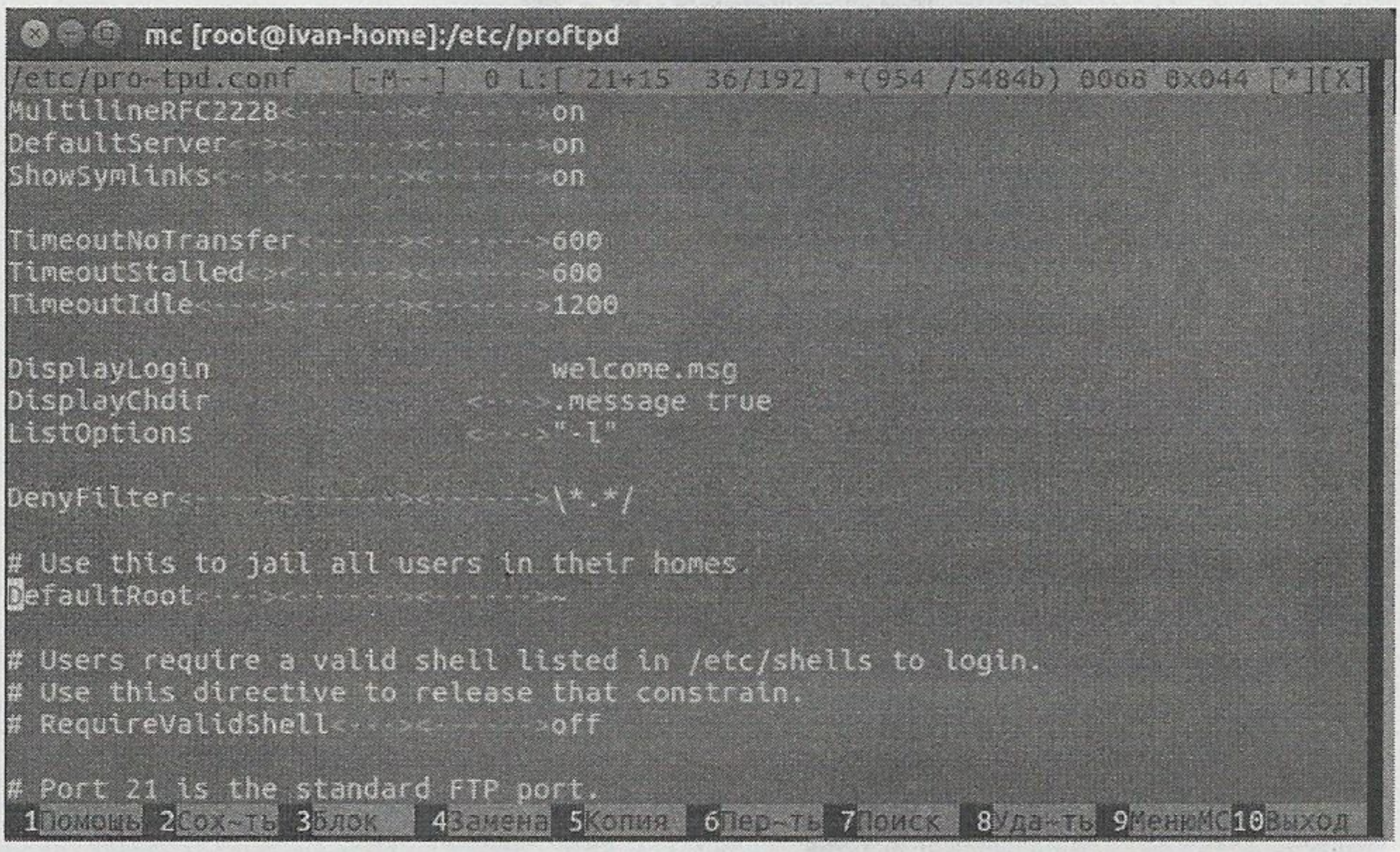
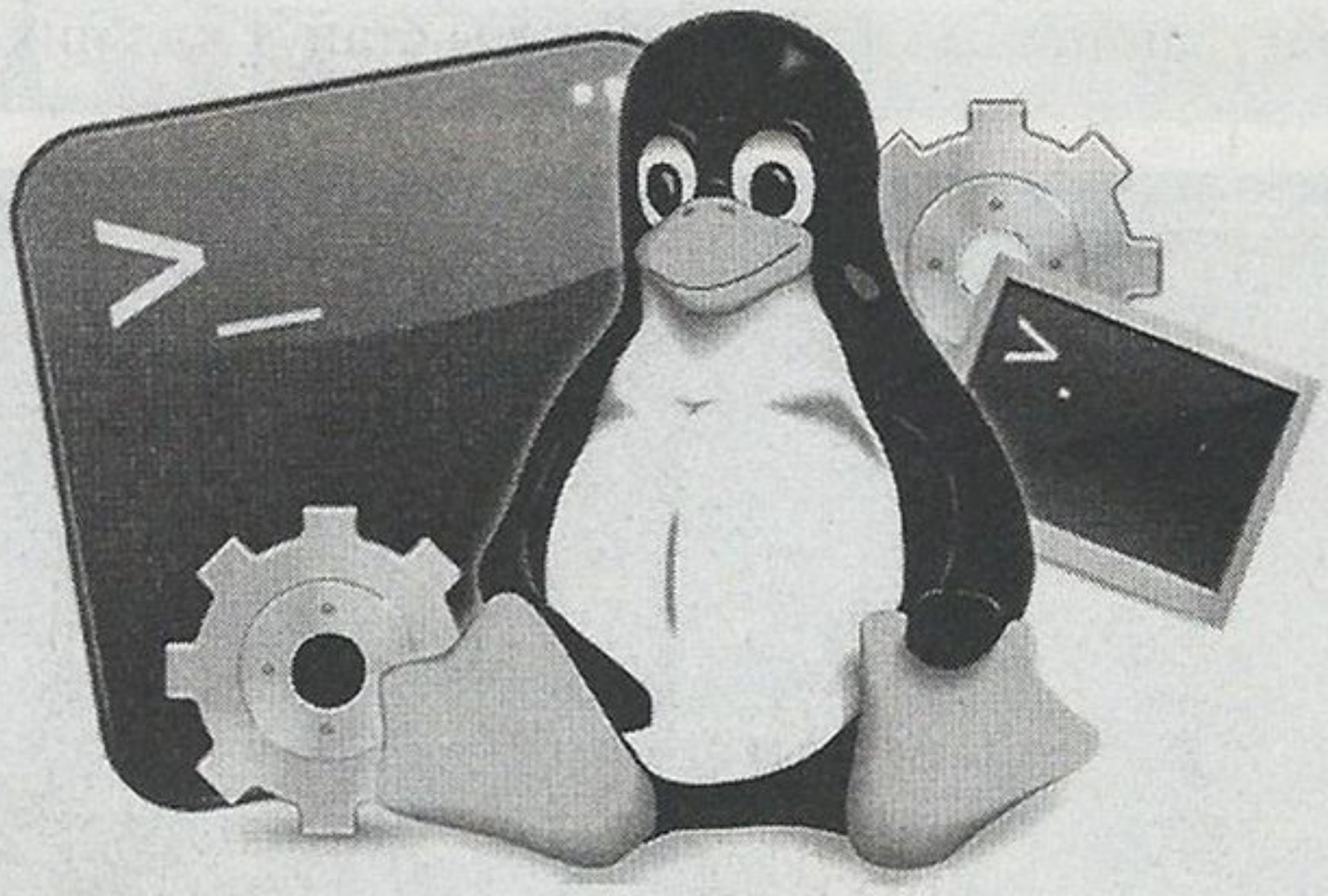
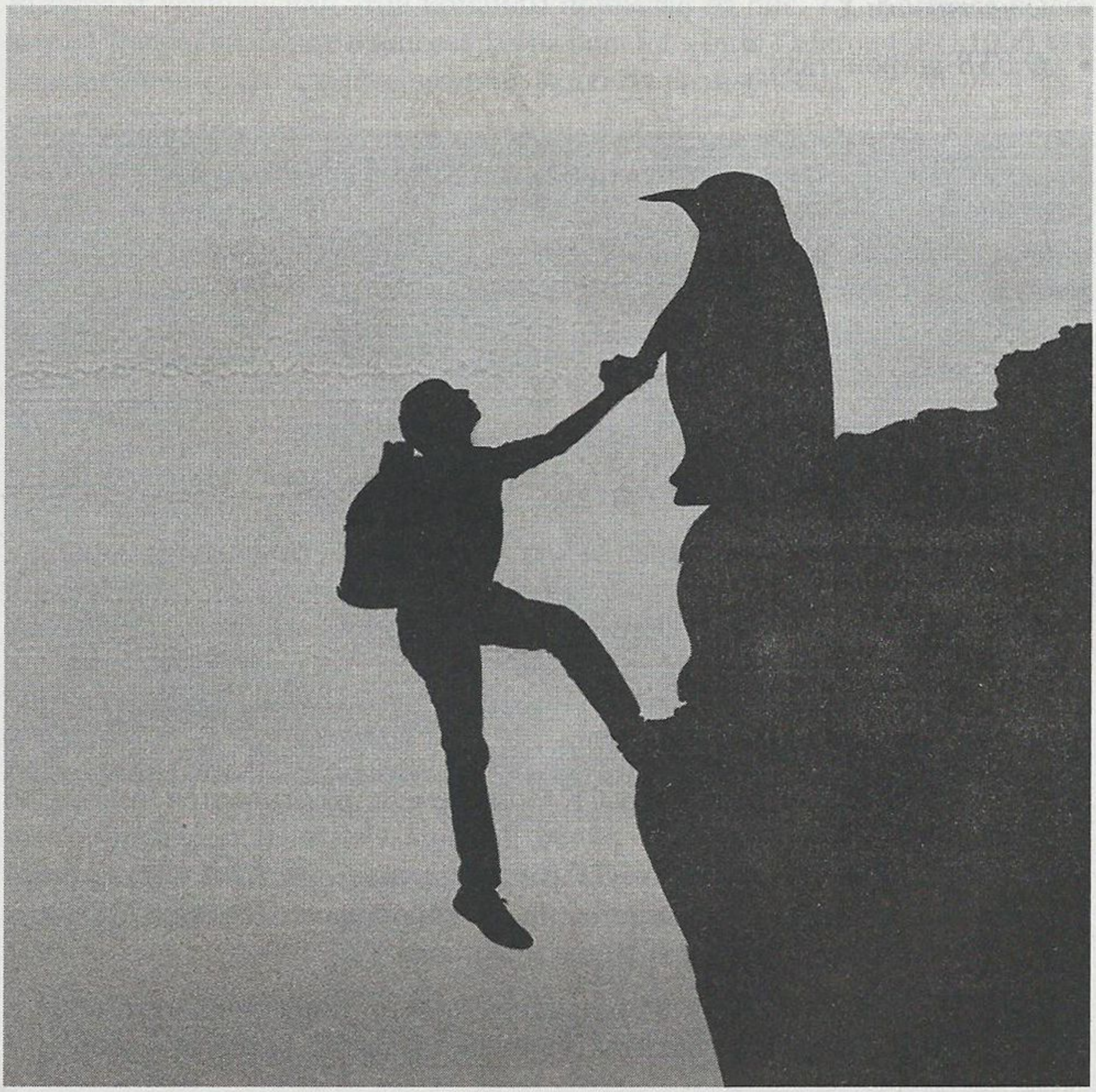


Рис. 6.36. Редактирование конфигурации ProFTPD



ГЛАВА 7.

Управление аппаратными средствами



В этой главе:

- *Определение архитектуры процессора*
- *Добавление в систему еще одного диска*
- *О настройке X*
- *О USB-устройствах*

Предмет данной главы - управление аппаратными средствами вашего компьютера средствами Linux. Мы рассмотрим процесс получения информации о процессоре и материнской плате, разберемся, как добавить еще один жесткий диск, как работать с USB-устройствами и т.д.

7.1. Процессор и материнская плата

Процессор (или же CPU) - мозг вашего компьютера. Все вычисления производит та самая небольшая микросхема, которая называется процессором. Процессоры постоянно совершенствуются, существуют самые различные семейства процессоров. На обычных настольных компьютерах (имеются в виду стационарные компьютеры, ноутбуки и нетбуки) используются процессоры следующих семейств:

- x86 - 32-разрядные процессоры. К этому семейству относится Intel 80386 и его более новые собраться - 80486, Pentium, Celeron и даже ранние версии процессоров серии Core. Да, ранние процессоры Intel Core i3 были 32-разрядными. Что же касается процессоров производства AMD, то к этому семейству относятся Athlon и Duron.
- x86-64 - сюда относятся 64-разрядные версии процессоров, Intel 64 и AMD64. AMD создала архитектуру x86_64 как 64-разрядное расшире-

ние архитектуры x86. Конечно, эти процессоры могут выполнять код, написанный для архитектуры x86, однако при выполнении 64-битного кода, который оптимизирован под эту архитектуру, производительность повышается. Поэтому нет смысла устанавливать 32-битные версии Linux на современные процессоры. Да и сейчас наблюдается тенденция отказа от поддержки 32-битных версий. Начиная с 2007 года, практически все процессоры принадлежат этому семейству. Все современные процессоры Intel - Core i3, Core i5, Core i7 - относится к семейству x86_x64.

Узнать, какой процессор у вас установлен можно командой:

```
cat /proc/cpuinfo | less
```

Результат выполнения этой команды приведен на рис. 7.1. Команда выводит не только название процессора (на рис. 7.1 - Intel Celeron CPU 1000 M), но и различные его характеристики, включая размер кэша.

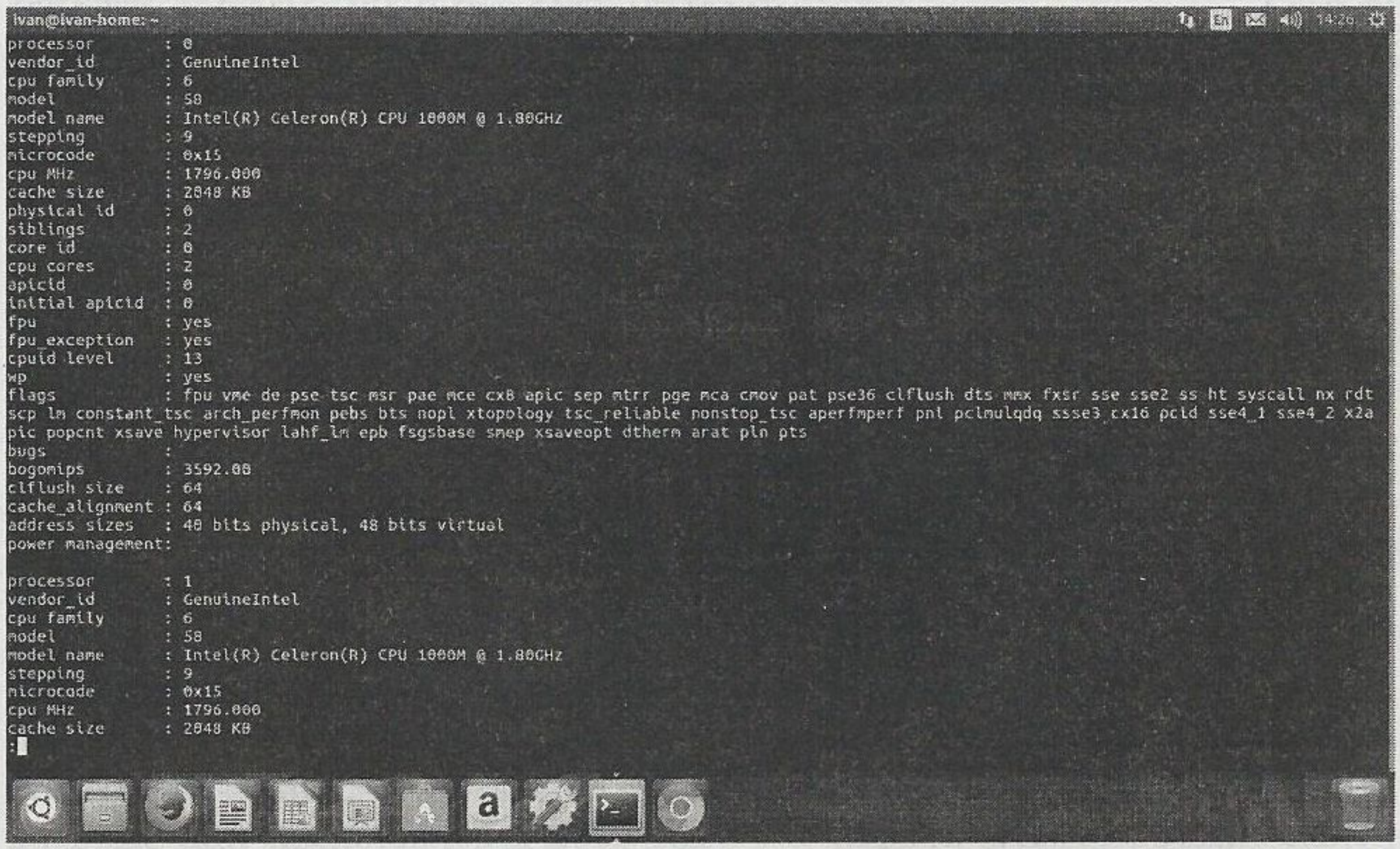


Рис. 7.1. Информация о процессоре

Архитектуру системы можно узнать с помощью команды uname -a. Ее вывод будет примерно таким:

```
Linux ivan-home 4.10.0-19-generic #21-Ubuntu SMP Thu Apr 6
17:04:57 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
```

Видно, что в данный момент используется 64-битная система. Наконец, команда lscpu выводит сокращенную версию информации из cpuinfo и архитектуру процессора, например:

```
Архитектура:x86_64
CPU op-mode(s):                  32-bit, 64-bit
Порядок байтов:                  Little Endian
CPU(s):                          2
On-line CPU(s) list:             0,1
Потоков на ядро:                 1
Ядер на сокет:                   2
Сокет(ы):                        1
NUMA node(s):                    1
Vendor ID:                       GenuineIntel
Семейство CPU:                   6
Модель:                          58
Модель:                          Intel(R) Celeron(R) CPU 1000M @ 1.80GHz
Stepping:                        9
CPU МГц:                         1796.000
BogoMIPS:                        3592.00
```

Теперь получим информацию о материнской плате. Для этого используется команда lspci:

```
00:00.0 Host bridge: Intel Corporation 2nd Generation Core Processor Family DRAM
Controller (rev 09)
00:02.0 VGA compatible controller: Intel Corporation 2nd Generation Core Processor
Family Integrated Graphics Controller (rev 09)
00:16.0 Communication controller: Intel Corporation 6 Series/C200 Series Chipset
Family MEI Controller #1 (rev 04)
00:1a.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB
Enhanced Host Controller #2 (rev 04)
00:1b.0 Audio device: Intel Corporation 6 Series/C200 Series Chipset Family High
Definition Audio Controller (rev 04)
00:1c.0 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI
Express Root Port 1 (rev b4)
00:1c.1 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI
Express Root Port 2 (rev b4)
00:1c.2 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI
Express Root Port 3 (rev b4)
00:1c.3 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI
Express Root Port 4 (rev b4)
```

```
00:1c.7 PCI bridge: Intel Corporation 6 Series/C200 Series Chipset Family PCI
Express Root Port 8 (rev b4)
00:1d.0 USB controller: Intel Corporation 6 Series/C200 Series Chipset Family USB
Enhanced Host Controller #1 (rev 04)
00:1f.0 ISA bridge: Intel Corporation HM65 Express Chipset Family LPC Controller
(rev 04)
00:1f.2 SATA controller: Intel Corporation 6 Series/C200 Series Chipset Family 6
port SATA AHCI Controller (rev 04)
00:1f.3 SMBus: Intel Corporation 6 Series/C200 Series Chipset Family SMBus
Controller (rev 04)
02:00.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL8111/8168/8411 PCI
Express Gigabit Ethernet Controller (rev 06)
03:00.0 System peripheral: Ricoh Co Ltd PCIe SDXC/MMC Host Controller (rev 07)
08:00.0 Network controller: Intel Corporation Centrino Wireless-N 1000 [Condor
Peak]
```

Что можно узнать из этого вывода? Во-первых, какой установлен чипсет, во-вторых, какие контроллеры аудио и видео установлены. Какой используется SATA-контроллер и какой контроллер сети Ethernet используется. Также видно, что установлен беспроводный сетевой адаптер Intel Corporation Centrino Wireless-N 1000.

7.2. Управление дисками

7.2.1. Различные интерфейсы

Сегодня на большинстве настольных компьютеров (серверы в расчет не берем - там совсем другая ситуация) используются два дисковых интерфейса или PATA (Parallel ATA) или SATA (Serial ATA).

PATA - это ни что иное как всем известный IDE (Integrated Drive Electronics). На данный момент он практически вытеснен интерфейсом SATA, который впервые был представлен в 2003 году. На данный момент сложно найти компьютер с интерфейсом PATA. Как правило, это старые компьютеры, выпущенные до 2007 года. С 2007 года компьютеры могут оснащаться или двумя интерфейсами - PATA и SATA или только SATA.

В старых дистрибутивах (когда еще интерфейс PATA был актуален) жестким дискам, подключенным к контроллеру PATA, назначались имена вида /dev/hdX, где X - буква, которая назначалась в зависимости от порядка под-

ключения диска к контроллеру, например, /dev/hda - диск, подключенный как Primary Master, /dev/hdb - Primary Slave и т.д. Дискам SATA, если такие были установлены, назначались немного другие имена - /dev/sdX, где X - это буква, означающая порядок подключения диска к SATA-контроллеру, например, если диск первый, то устройству назначалась буква a (/dev/sda), второму - b (/dev/sdb) и т.д.

В современных дистрибутивах любому диску назначаются имена вида /dev/sdX. Причем, если в компьютере есть PATA и SATA диск, то первым (то есть получит имя /dev/sda) будет PATA-диск, а потом уже - SATA (он получит имя /dev/sdb). Однако, учитывая, что на современных материнских платах больше нет контроллера IDE, то такой проблемы не будет - на современных компьютерах остались только более производительные SATA-контроллеры.

7.2.2. Добавление еще одного жесткого диска и его разметка

Представим, что вы решили купить еще один жесткий диск. Давайте рассмотрим все действия по его добавлению в систему:

- Выключите компьютер, отключите кабель питания - на всякий случай.
- Откройте корпус компьютера, подключите жесткий диск с помощью SATA-кабеля к свободному SATA-разъему. Подробную информацию вы найдете в документации по материнской плате. Подключите кабель питания для жесткого диска.
- Соберите компьютер, подключите кабель питания и включите компьютер.
- Дождитесь, пока компьютер загрузится.

Если вы сразу откроете Nautilus, то чуда не произойдет - вы не увидите еще один диск. А все потому, что нужно произвести его разметку (то есть разделение на разделы) и подмонтировать его к корневой файловой системе. Диски, которые уже были в использовании, нужно только подмонтировать - создавать разделы не нужно, поскольку они уже созданы.

Для создания разделов используются различные программы - fdisk, cfgisk, parted. Но все эти программы работают в текстовом режиме, а пользователю, перешедшему с Windows, хочется какой-то наглядной визуальной программы для разметки диска. И такая программа есть: она называется gparted. По умолчанию она не устанавливается, поэтому установим ее командой:

```
sudo apt install gparted
```

После этого запустите ее:

```
sudo gparted
```

После запуска программа отобразит разметку системного диска - на котором находится корневая файловая система. Как видно из рис. 7.2, на этом диске создан один раздел. Учитывая размер накопителя - 32 Гб - нет смысла создавать несколько разделов. В верхнем правом углу есть список выбора диска. Узнать какой из них какой можно по косвенным признакам - по размеру, например, или по разметке (на новом диске не будут созданы разделы).

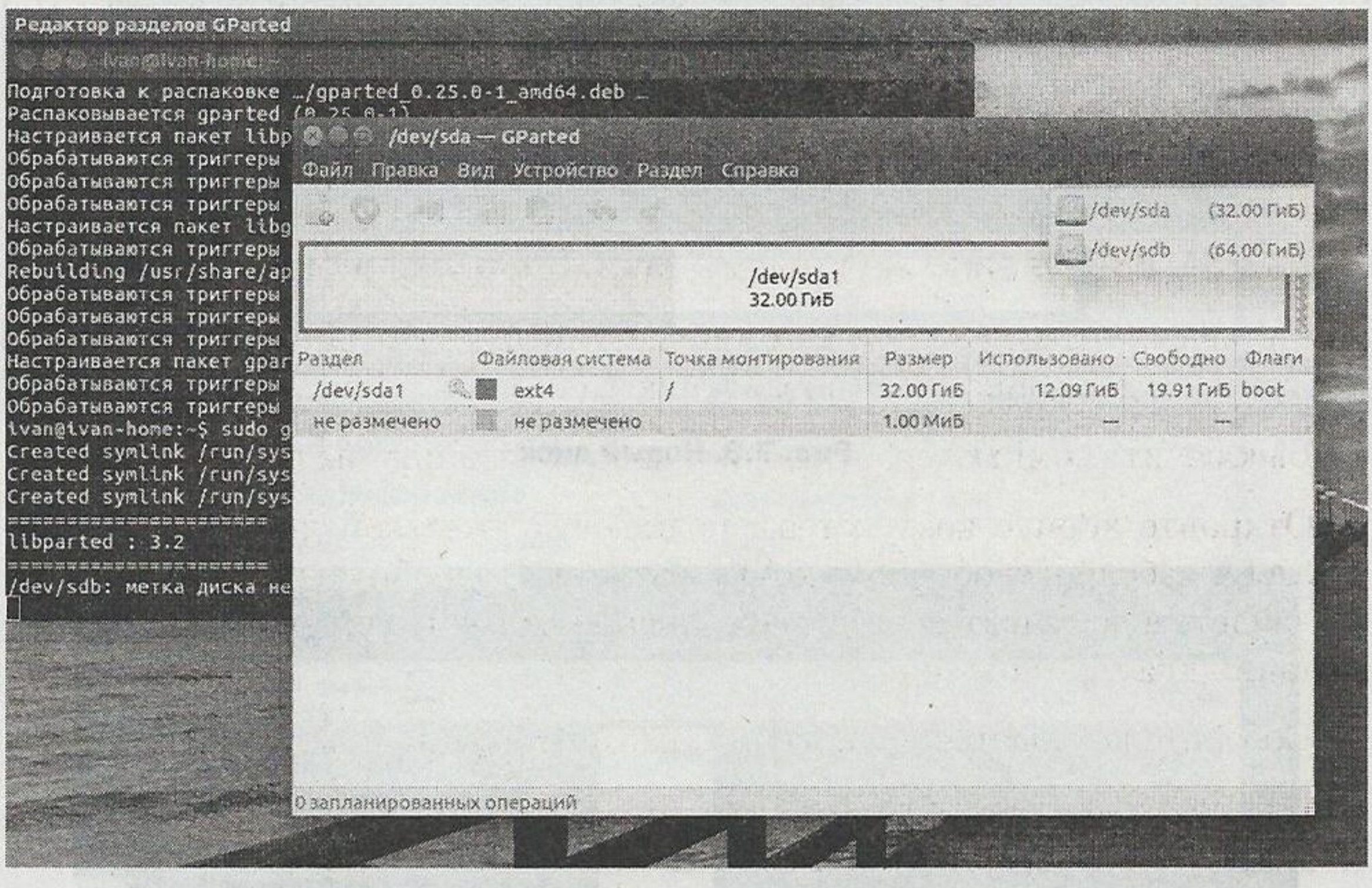


Рис. 7.2. Программа gparted

Итак, выберите диск, на котором нужно создать разделы. На рис. 7.3 показан, как отображается в программе новый диск - на нем не созданы разделы. Заметьте, я не произношу слова «жесткий» - диск может быть и SSD.

Разметка диска начинается с создания таблицы разделов. Выполните команду меню **Устройство, Создать таблицу разделов**. Программа позволяет выбрать тип таблицы разделов. Поддерживаются типы msdos, gpt, bsd и некоторые экзотические таблицы, с которыми вам не придется сталкиваться. Вам нужно выбрать или msdos или gpt, если размер диска превышает 2 ТБ. В нашем случае размер диска 64 Гб, поэтому можно смело выбрать msdos и нажать кнопку **Применить**.

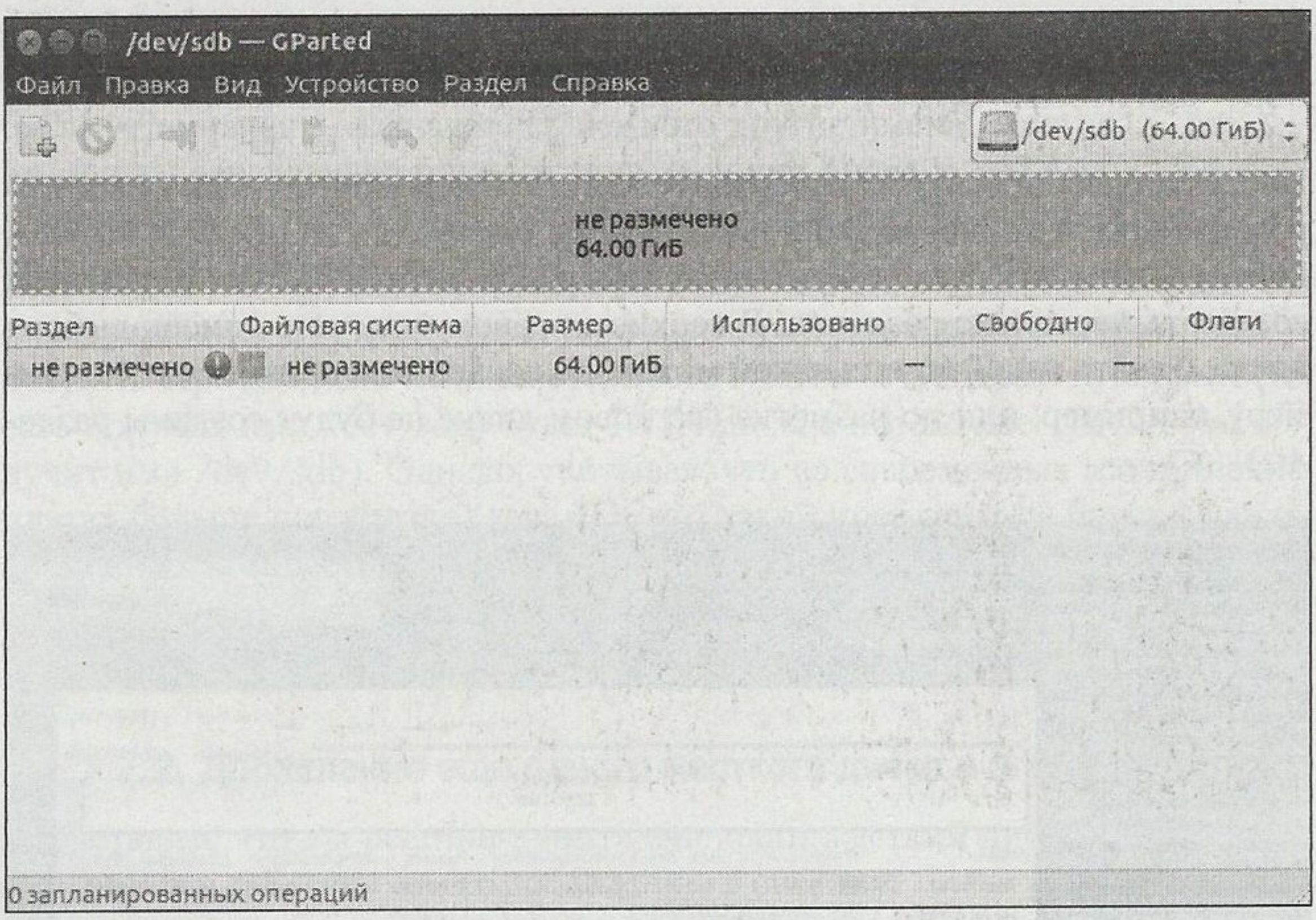


Рис. 7.3. Новый диск

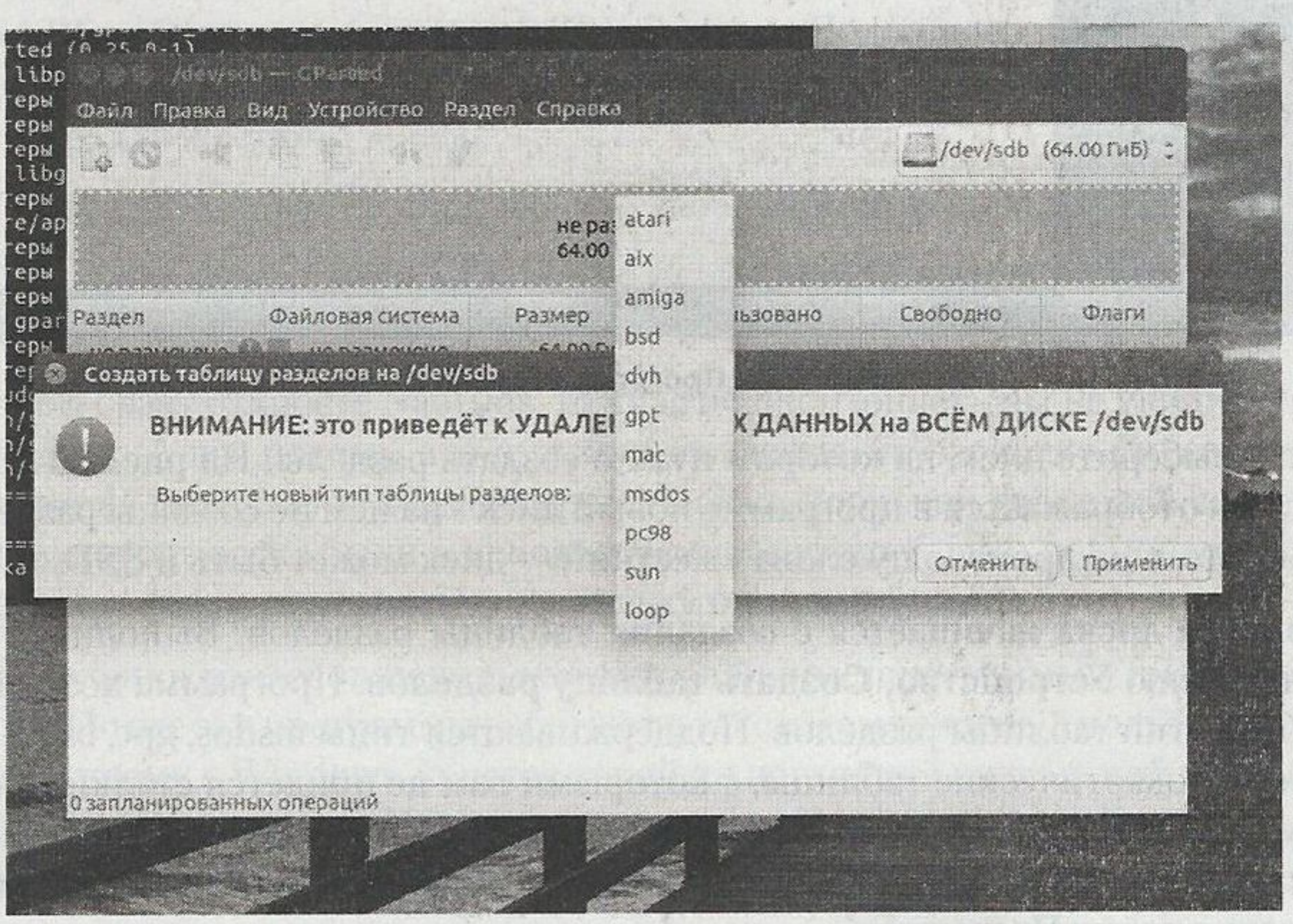


Рис. 7.4. Создание таблицы разделов

После этого выберите команду меню **Раздел, Новый** или нажмите кнопку с зеленым + на панели инструментов. Появится окно создания нового раздела. Установите раздел размера (если хотите создать раздел на весь диск, ничего изменять не нужно) и выберите файловую систему:

- ext4 - если планируется использование раздела только в Linux.
- ntfs - если планируете этот раздел совместно использовать в Windows.
- fat32 - если этот диск будет подключаться к различным устройствам вроде мультимедиа-проигрывателя или к автомобильной мультимедийной системе.

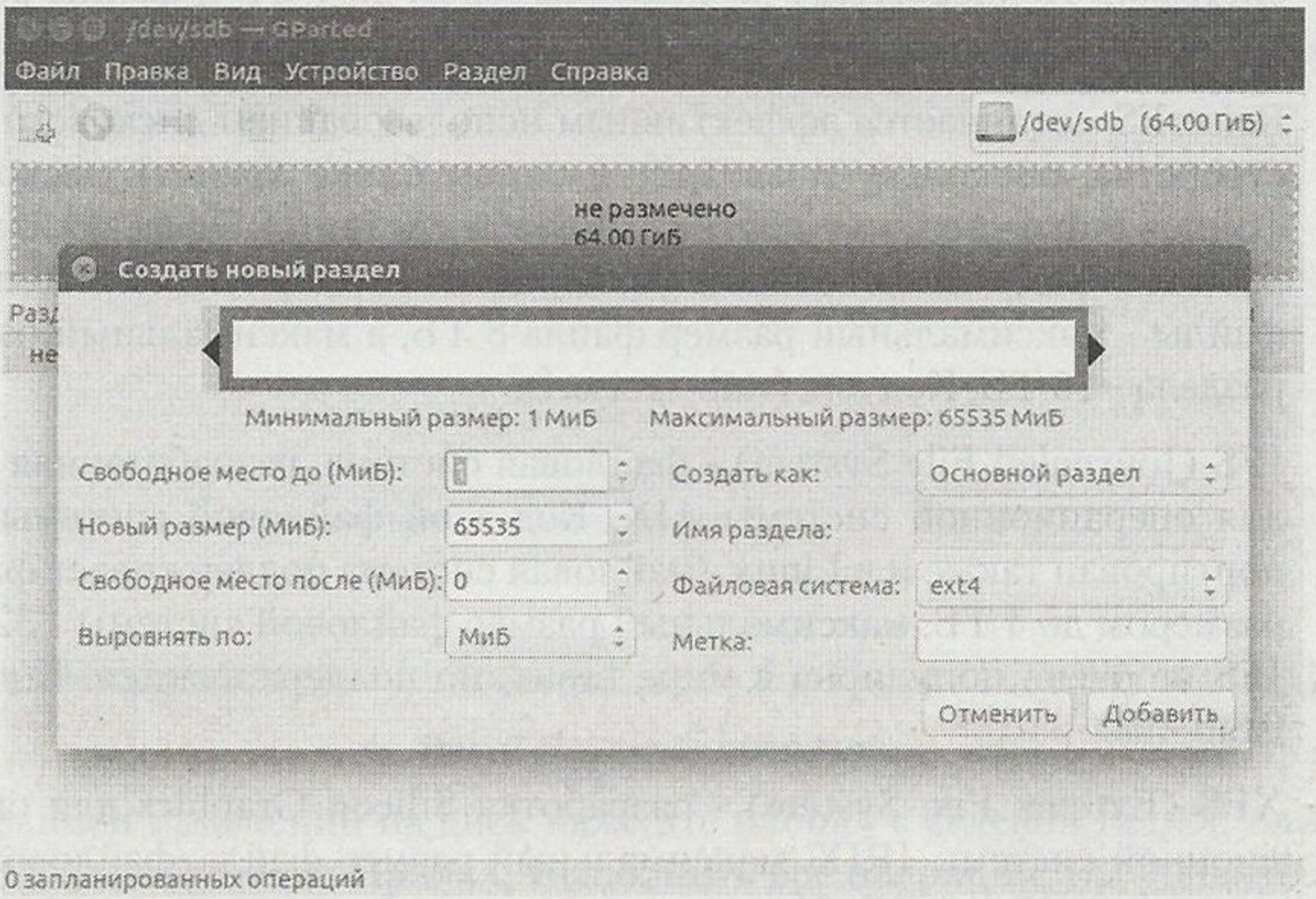


Рис. 7.5. Создание нового раздела

Примечание (оформить в виде врезки). **Краткая справка по файловым системам Linux**:

- ext2fs (The Second Extended File System) - была очень популярной (по сути основной) в 90-ые годы, сегодня практически не используется, но из соображений совместимости поддерживается современными дистрибутивами. Основной ее недостаток - отсутствия журнала, помогающего восстановить данные в случае системного сбоя, например, при отключении электричества. Однако, сам по себе журнал занимает много места, поэтому ext2 все еще может пригодиться - на небольших дисках, где размер важнее сохранности данных (например, на ноутбуках - жесткие диски ноутбуков могут быть не очень большими, а

ситуация с отключением электричества им не страшна). Код для fstab - ext2.

- ext3fs (The Third Extended File System) - по сути, то же самое, что и ext2, но с журналом. Была основной файловой системой Linux вплоть до 2010 года, пока не появилась ext4. Поддерживает размеры разделов до 16 ТБ и размеры файлов до 2 ТБ (у ext2 те же самые ограничения). Код для fstab - ext3.
- ext4fs (The Fourth Extended File System) - в отличие от ext3, которая является надстройкой над ext2, является абсолютно новой файловой системой. Поддерживает разделы размером до 1 ЭБ и файлы размером до 16 ТБ. Код для fstab - ext4.
- ReiserFS - отличается эффективным использованием дискового пространства, поскольку позволяет в одном блоке хранить несколько файлов (такое не могут другие файловые системы). Эффективна для работы с файлами небольших размеров, но поддерживает и большие файлы - максимальный размер файла 8 ТБ, а максимальный размер раздела - 16 ТБ. Код для fstab - reiserfs.
- JFS (Journaled File System) - файловая система, разработанная IBM для операционной системы AIX. Код этой файловой системы был портирован также и в Linux. Файловая система поддерживает файлы размером до 4 ТБ, максимальный размер файловой системы - 32 ПБ. JFS не очень популярна в мире Linux, но поддерживается. Код для fstab - jfs.
- XFS (Extents File System) - разработка Silicon Graphics для операционной системы IRIX, максимальный размер файла/файловой системы - 8 ЭБ/16 ЭБ. Неплохо показала себя при использовании в больших дисковых массивах. Хорошо работает с большими мультимедиа-файлами. Код для fstab - xfs.
- Btrfs - новая файловая система. Ей суждено стать следующей файловой системой после btrfs. Однако разработчики дистрибутивов, как показывает практика, не спешат переходить на эту файловую систему. Максимальный размер файла/файловой системы - 16 ЭБ/16 ЭБ.

Также нужно выбрать тип раздела. Таблица разделов MSDOS предполагает наличие на диске до 4 разделов. Вы можете создать или 4 основных раздела или 3 основных и один расширенный, а уже в расширенном создать дополнительные разделы, которые в терминологии gparted называются логическими дисками - это на случай, если 4 разделов окажется недостаточно. Как правило, сейчас мало кто без особой нужды разбивает жесткий диск на разные разделы, поэтому можно создать один основной раздел размером с

весь жесткий диск. Исключения составляют серверы, где желательно разделять разные данные - хранить на разных разделах.

Нажмите кнопку **Добавить**. В нижней части окна программы вы увидите список операций - в нашем случае будет одна операция по созданию основного раздела диска с файловой системой ext4 (рис. 7.6).

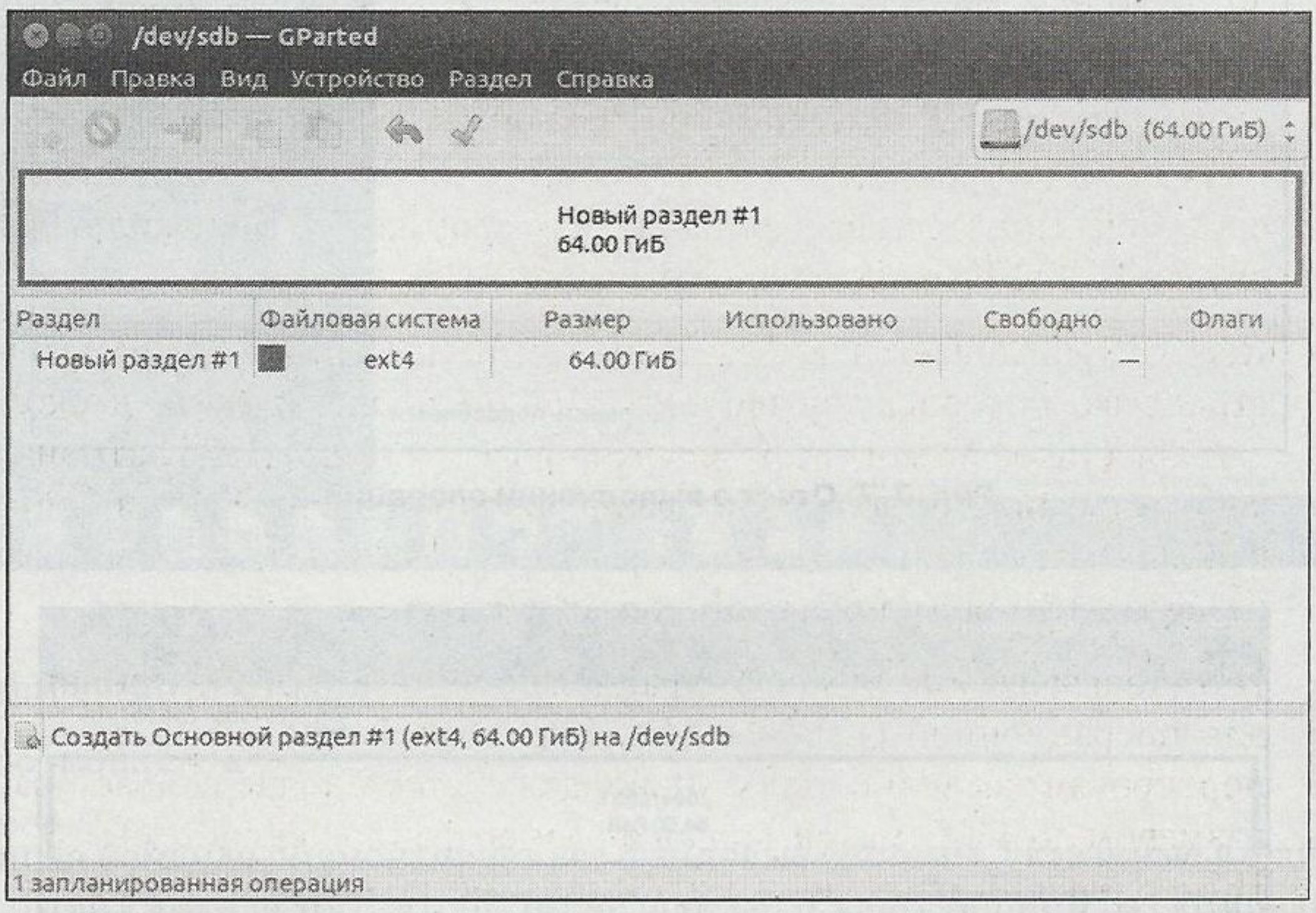


Рис. 7.6. Список операций

Для записи изменений на диск нажмите кнопку с зеленой галкой или выберите команду меню **Правка, Выполнить все операции**. Вы увидите окно с предупреждением, в котором нужно будет нажать кнопку **Применить**. После этого будет отображен отчет о выполнении операции (рис. 7.7). Окончательный результат наших стараний изображен на рис. 7.8 - созданная разметка.

Программу gparted можно закрывать. Осталось отредактировать /etc/fstab, чтобы созданный нами раздел /dev/sdb1 монтировался при загрузке системы. Но сначала нужно создать точку монтирования для него:

```
sudo mkdir /mnt/sdb1
```

При желании можете выбрать другую точку монтирования, например, /mnt/films или /mnt/music - разницы нет. После этого откройте текстовый редактор (с правами sudo) и добавьте строчку в /etc/fstab:

```
/dev/sdb1    /mnt/sdb1    ext4    rw    0    0
```

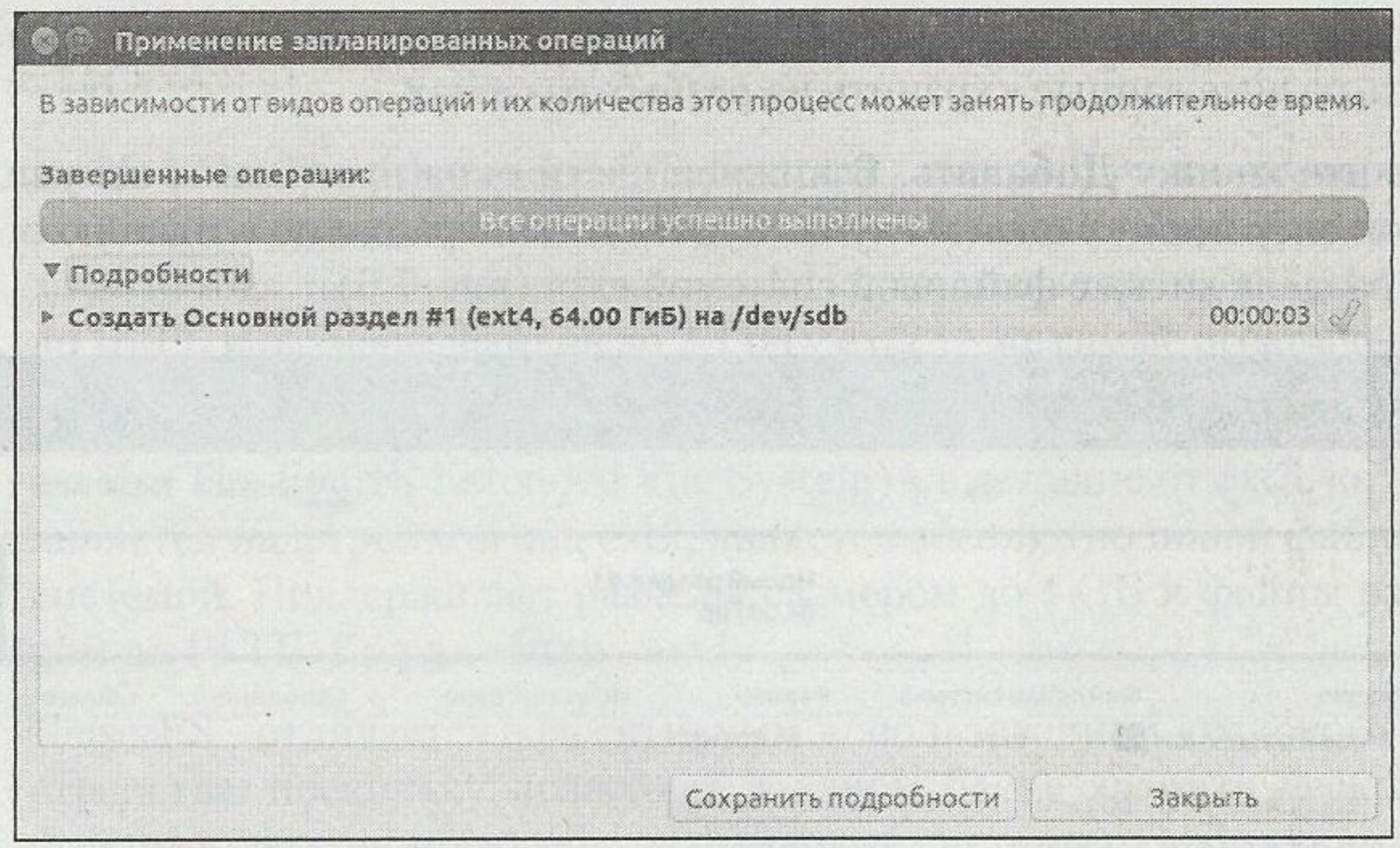


Рис. 7.7. Отчет о выполнении операций

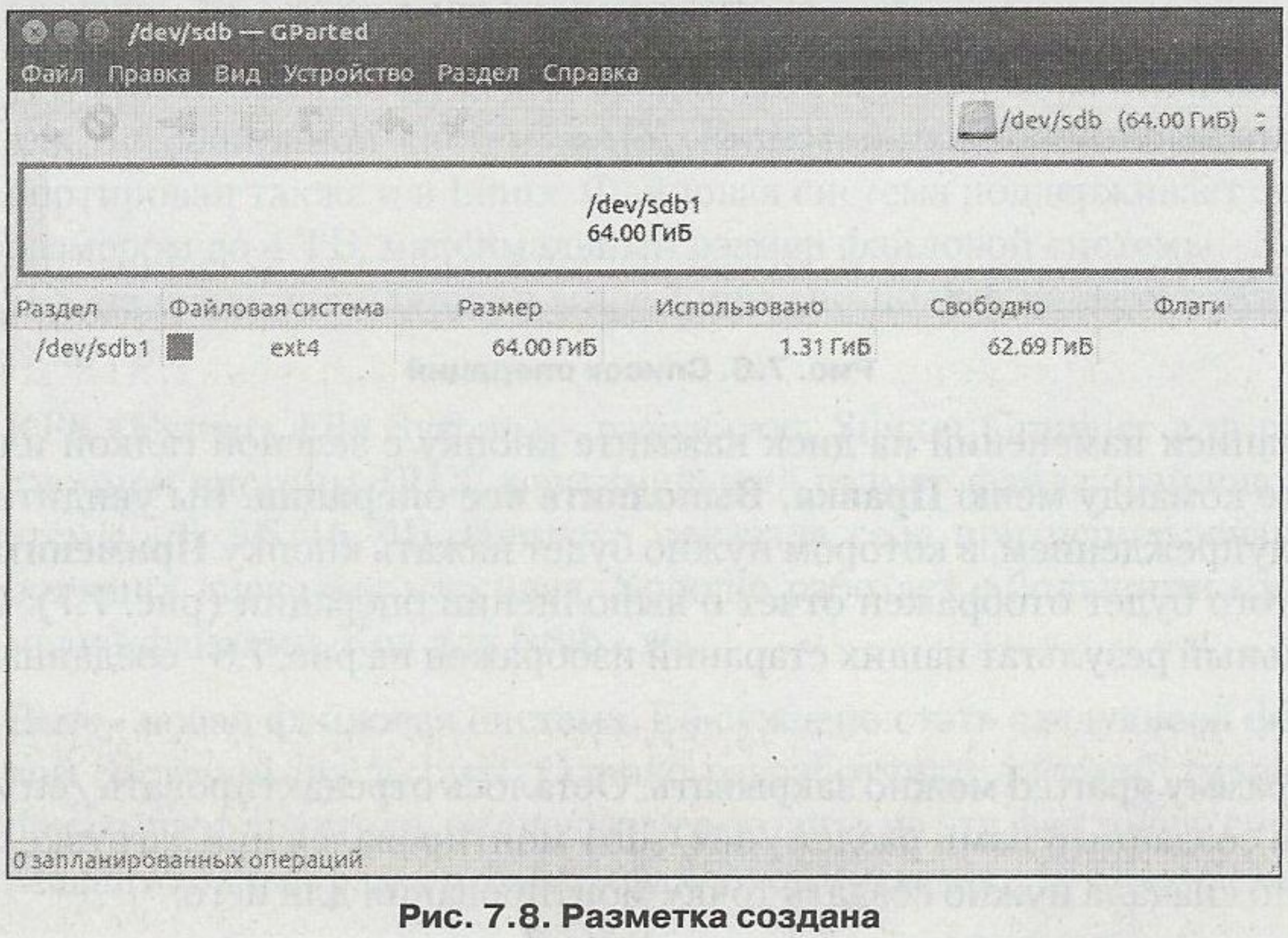


Рис. 7.8. Разметка создана

Если вы создали NTFS-раздел, то строка будет такой:

```
/dev/sdb1    /mnt/sdb1    ntfs-3g    rw,noexec,utf8    0    0
```

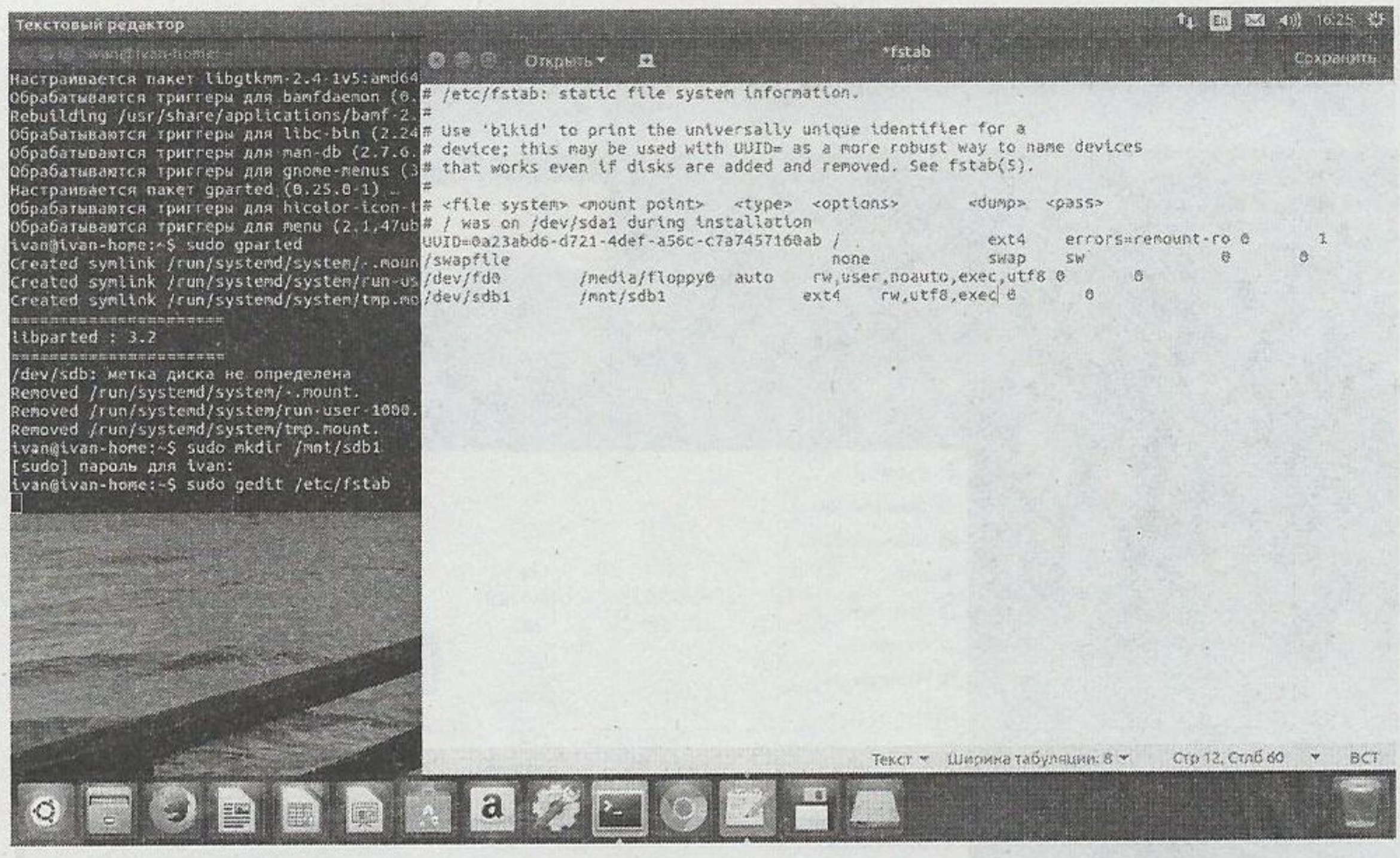


Рис. 7.9. Редактирование fstab

Затем введите команду:

```
sudo mount -a
```

Данная команда подмонтирует все файловые системы, указанные в файле /etc/fstab - это для того, чтобы вы не перезагружали лишний раз компьютер. Если вы все сделали правильно, то в процессе выполнения этой команды не будет выведено никаких сообщений об ошибках. Обычно ошибки могут быть связаны с неправильно указанными опциями файловых систем - не все файловые системы поддерживают те или иные опции. Чтобы продемонстрировать такое сообщение, я эмулировал ошибочную ситуацию и ее результат приведен на рис. 7.10.

```
ivan@ivan-home:~$ sudo mount -a
mount: wrong fs type, bad option, bad superblock on /dev/sdb1,
       missing codepage or helper program, or other error

       In some cases useful info is found in syslog - try
       dmesg | tail or so.
```

Рис. 7.10. Так выглядит ошибка при монтировании

Теперь нам осталось убедиться, что все сделано правильно. Команда mount выводит список смонтированных файловых систем. Посмотрим, смонтирован ли наш раздел:

```
mount | grep sdb1
```

Как видно из рис. 7.11, раздел смонтирован, можно приступать к его использованию - открыть файловый менеджер и перейти к /mnt/sdb1.

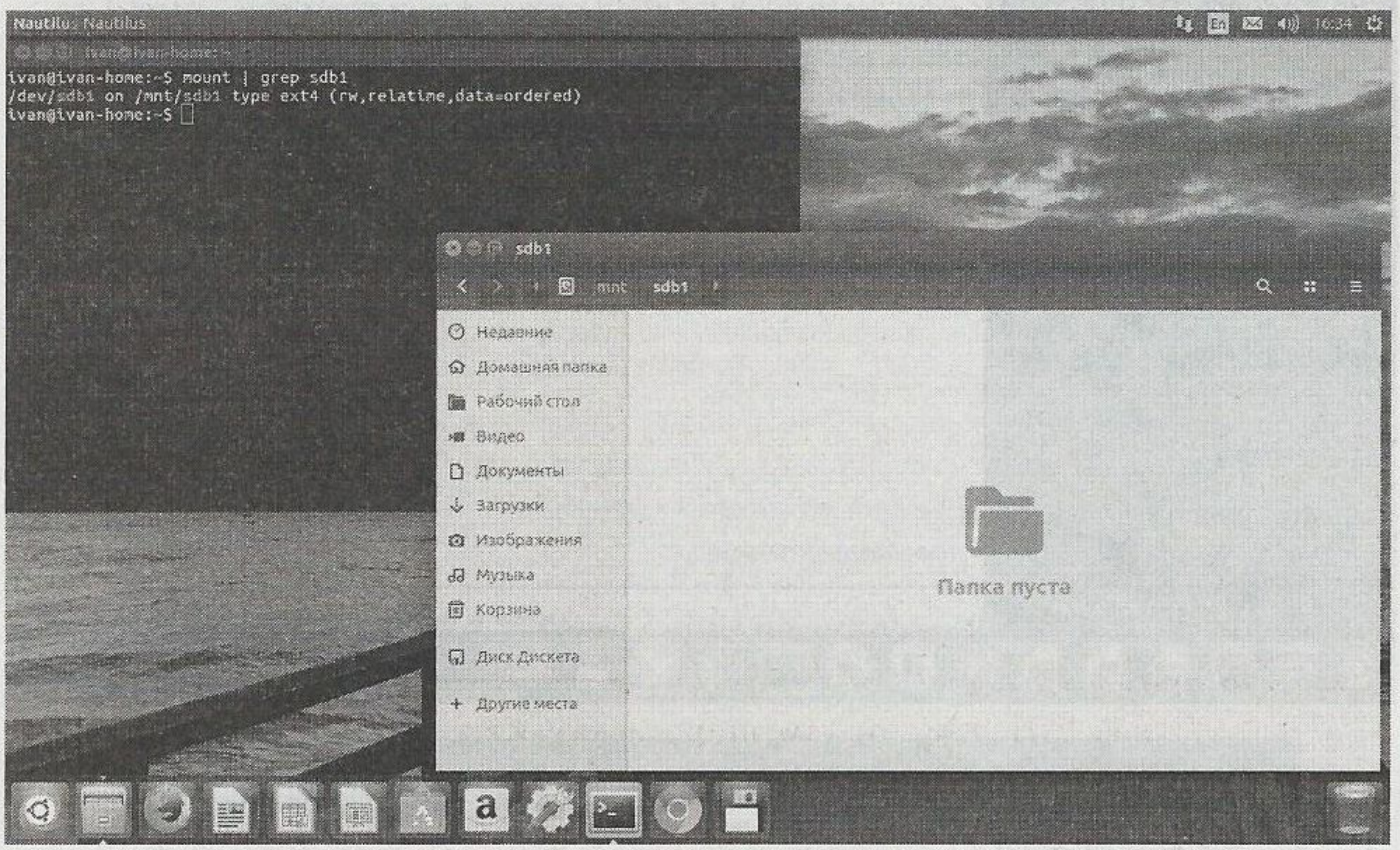


Рис. 7.11. Все нормально: раздел готов к работе

7.2.3. Сменные и оптические диски

Сменные и оптические диски, как уже было показано ранее, монтируются автоматически. Монтирование происходит к каталогу /media/<UUID-устройства>.

При работе с внешними жесткими дисками и флешками важно размонтировать носитель перед его физическим отключением. Для этого нажмите кнопку извлечения напротив значка этого устройства (см. рис. 7.12).

Игнорирование этого требования может привести к повреждению данных на сменном носителе!

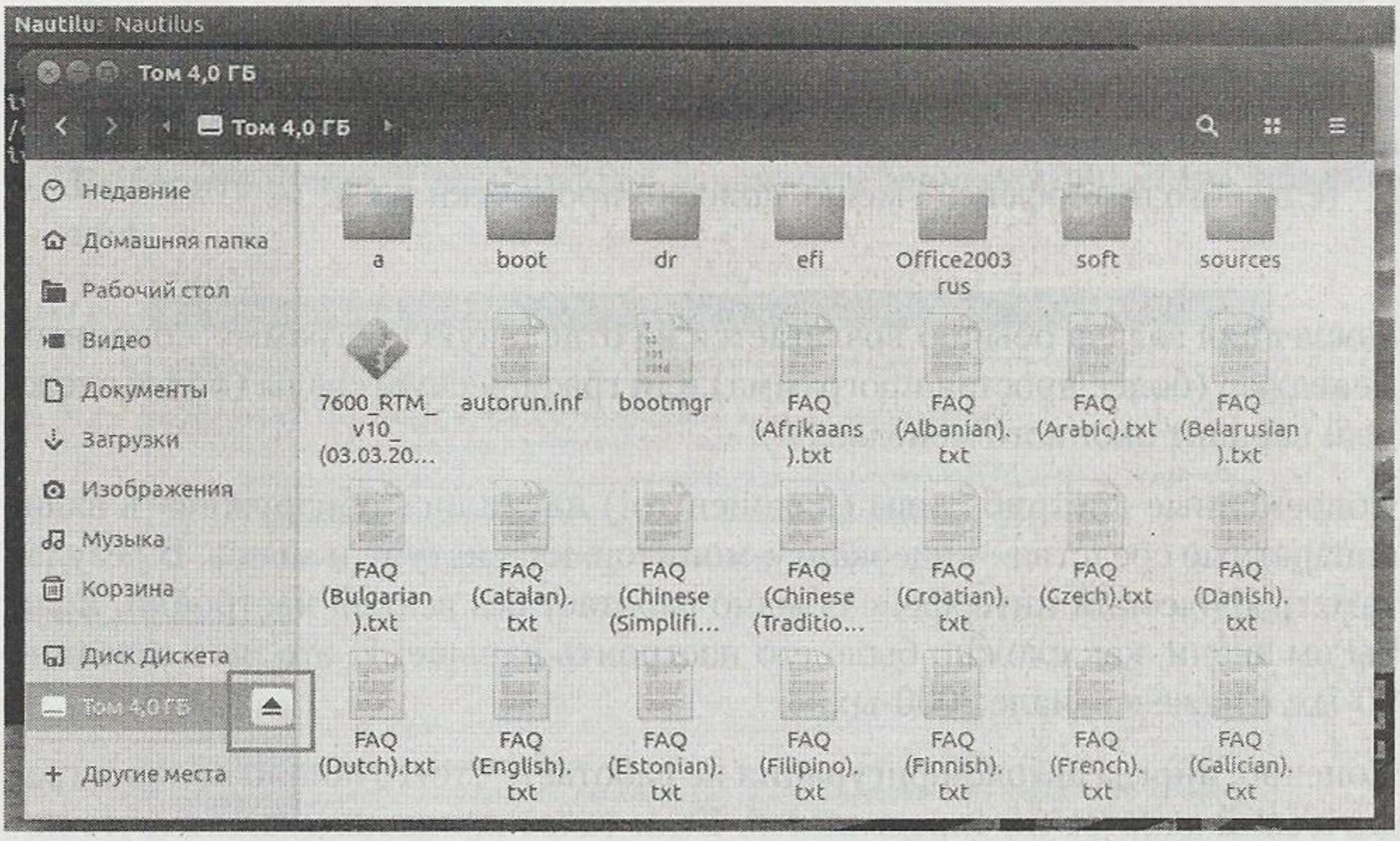


Рис. 7.12. Кнопка размонтирования сменного носителя

7.3. Мониторы и графические адаптеры

Как вы знаете, у Linux есть два режима экрана - текстовый и графический. Текстовый, как правило, не требует какой-либо настройки. Он просто работает. Настройка GUI более сложна. В последнее время графический интерфейс также практически не требует настройки - разве что можно выбрать более удобное разрешение экрана.

Но все же мы поговорим о настройке графического режима подробнее. В Linux экраном управляет X Window System или просто X. Давайте же разберемся, для чего нужна система X и как она взаимодействует с аппаратными средствами.

В большинстве случаев пользователи не понимают, как работает X. Работает себе, да и работает. В обязанности X Window System входит:

- Инициализация видеокарты, установка разрешения.
- Выделение фрагментов экрана, которые используются как окна определенных приложений.
- Управление окнами: переключение между ними, изменение размера, сворачивание и т.д.
- Управление указателем мыши или аналогичного устройства.

- Обработка ввода с клавиатуры и передача ввода приложению из активного окна.
- Управление внешним видом окон, отображение элементов пользовательского интерфейса - меню, полоски прокрутки и т.д.

Последняя задача обычно возлагается на отдельную программу - оконный менеджер (более простая программа) или графическую среды (более сложный вариант оконного менеджера).

Современные дистрибутивы (наконец-то!) научились обнаруживать ваши аппаратные средства - видеокарту, монитор, клавиатуру и мышь. В результате графический интерфейс обычно работает без всякой настройки. Если бы вы знали, как сложно было его настроить раньше! И это не в середине 90-ых, а даже в начале 2000-ых!

Конечно, иногда автоконфигурация не работает. Что-то пошло не так и графический адаптер не определяется. Если такое произошло, вы можете попытаться настроить X вручную. Его конфигурационный файл называется /etc/X11/xorg.conf. Если он отсутствует, первым делом нужно его сгенерировать. Убедитесь, что X не запущен и от пользователя root введите команду:

```
# Xorg -configure
```

Результат конфигурации будет помещен в файл /root/xorg.conf.new. Вы можете скопировать этот файл в /etc/X11/xorg.conf, отредактировать и попытаться запустить X. Проблема может быть в том, что в этом файле неправильно указан драйвер видеокарты. В большинстве случаев нужно использовать один из пересиленных ниже драйверов:

- nv, nvidia - для видеокарт производства nVidia.
- ati, fglrx - для видеокарт AMD/ATI.
- intel - для видеокарт от Intel.
- fbdev, vesa - стандартные драйверы, которые должны работать с самыми разными видеокартами.

Драйверы nvidia и fglrx являются проприетарными и предоставляются производителями (нужно загрузить с сайта производителя). В случае с Ubuntu эти драйверы можно установить, используя средства самой системы:

- Откройте окно **Параметры системы**

- Запустите приложение **Программы и обновления**
- Перейдите на вкладку **Дополнительные драйверы** (рис. 7.13)
- Выберите нужный вам драйвер и нажмите кнопку **Применить изменения**.

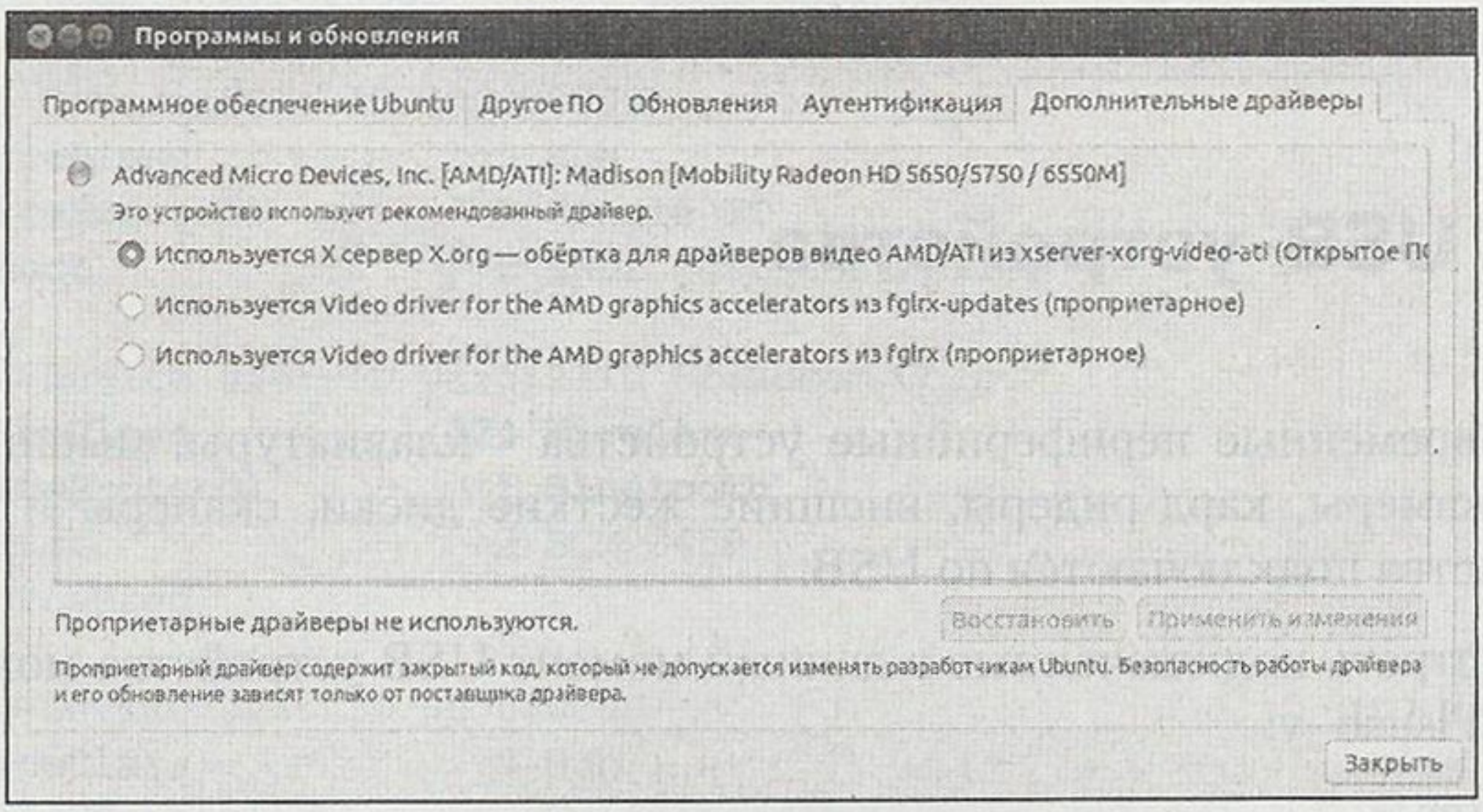


Рис. 7.13. Дополнительные драйверы

Для установки разрешения монитора используйте апплет **Настройка экранов** программы **Параметры системы** (рис. 7.14). Просто выберите нужное вам разрешение и нажмите кнопку **Применить**.

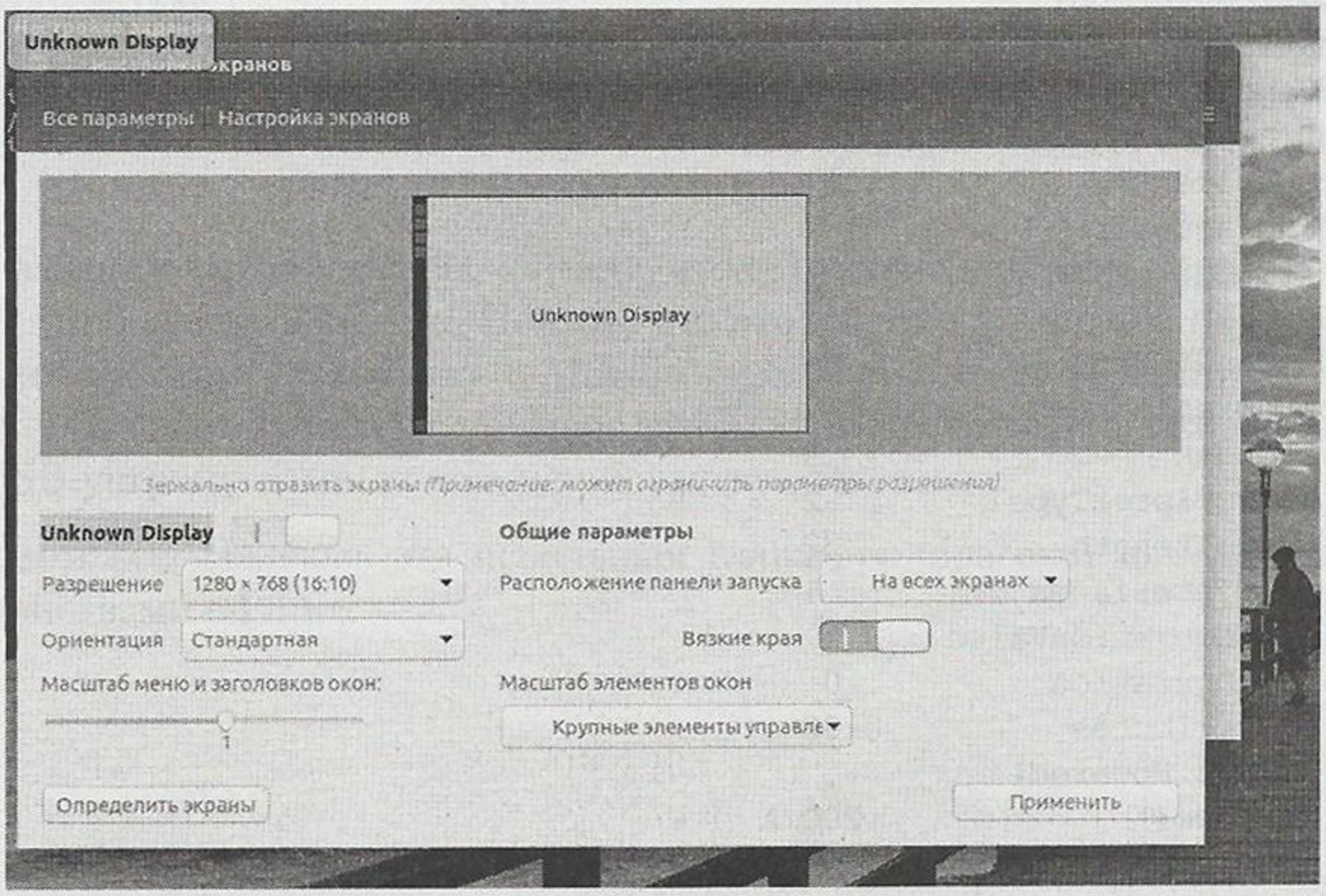


Рис. 7.14. Изменение разрешения экрана

Изменить разрешение можно и в консоли. Так узнать доступное разрешение и частоту обновления экрана можно с помощью команды:

```
xrandr
```

Изменить разрешение можно так:

```
xrandr --size <разрешение> --refresh <частота>
```

7.4. USB-устройства

Все современные периферийные устройства - клавиатуры, мыши, принтеры, камеры, кард-ридеры, внешние жесткие диски, сканеры и прочие устройства подключаются по USB.

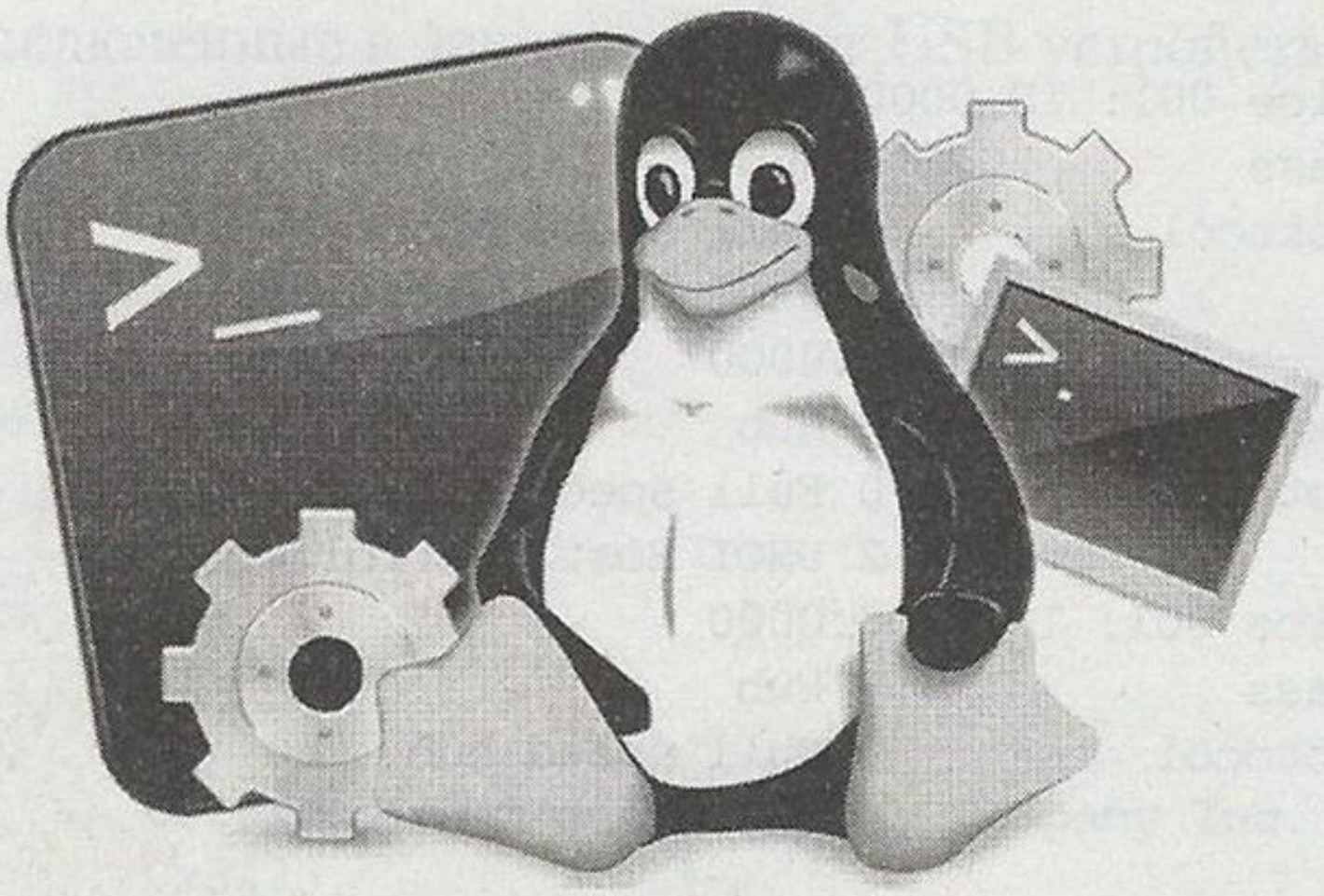
Просмотреть подключенные в данный момент USB устройства можно командой lsusb -v:

```
2>/dev/null
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0        64
  idVendor           0x13fe Kingston Technology Company Inc.
  idProduct          0x3223
  bcdDevice            1.10
  iManufacturer           1
  iProduct                2
  iSerial                 3
  bNumConfigurations      1
  Configuration Descriptor:
    bLength                 9
    bDescriptorType         2
    wTotalLength           32
    bNumInterfaces          1
    bConfigurationValue     1
    iConfiguration          0
    bmAttributes         0x80
      (Bus Powered)
    MaxPower              200mA
    Interface Descriptor:
      bLength                 9
```

```
        bDescriptorType         4
        bInterfaceNumber        0
        bAlternateSetting       0
        bNumEndpoints           2
        bInterfaceClass         8 Mass Storage
        bInterfaceSubClass      6 SCSI
        bInterfaceProtocol     80 Bulk-Only
...
Bus 005 Device 001: ID 0000:0000
  bDeviceClass                9 Hub
  bDeviceProtocol             1 Single TT
  iProduct                    2 EHCI Host Controller
...
Bus 004 Device 006: ID 0a5c:2110 Broadcom Corp.
  bDeviceClass              224 Wireless
  bDeviceProtocol             1 Bluetooth
  iProduct                    2 BCM2045B
  (Bus Powered)
...
Bus 004 Device 001: ID 0000:0000
  bDeviceClass                9 Hub
  bDeviceProtocol             0 Full speed hub
  iProduct                    2 UHCI Host Controller
Bus 001 Device 001: ID 0000:0000
  bDeviceClass                9 Hub
  bDeviceProtocol             0 Full speed hub
  iProduct                    2 UHCI Host Controller
Bus 002 Device 001: ID 0000:0000
  bDeviceClass                9 Hub
  bDeviceProtocol             0 Full speed hub
  iProduct                    2 UHCI Host Controller
...
```

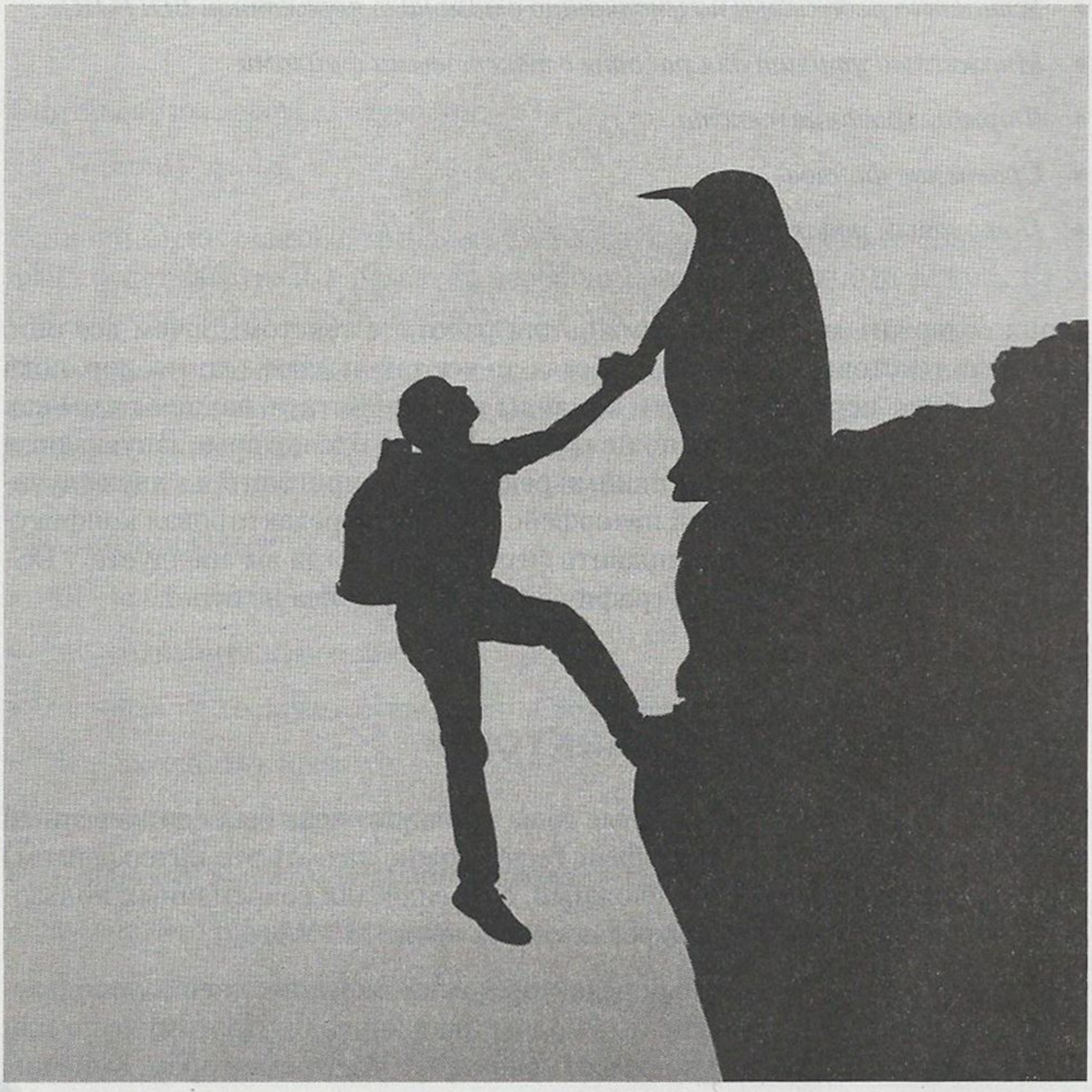
По названию устройства вы поймете, что есть что. Например, первая в списке - это флешка Kingston. Понять, что это именно флешка, а не принтер можно по типу устройства - MassStorage. Далее следует беспроводный Bluetooth-адаптер, также в примере есть USB-хабы - они ведь тоже являются USB-устройствами, поэтому присутствуют в списке.

На этом все. Надеюсь эта небольшая глава пролила свет на управление устройствами в Linux.



ГЛАВА 8.

Работа с текстом



В этой главе:

- *Самый неудобный текстовый редактор всех времен и народов*
- *Более удобные текстовые редакторы*
- *Установка редактора по умолчанию с помощью переменной EDITOR*
- *Множество утилит для работы с текстовыми файлами*
- *Форматирование текста*
- *Сравнение файлов*
- *Потоковый редактор*

Linux содержит множество команд для работы с текстом. Зачем все они, если есть текстовые редакторы? Когда-то все так думают - до тех пор, пока не попробуют использовать эти команды на деле. Кстати, начнем то мы как раз с текстовых редакторов, но не графических, а тех, которые запускаются в консоли. Умение работать с такими редакторами пригодится в двух случаях: если «упал» графический интерфейс и нужно отредактировал конфигурационные файлы, чтобы исправить ситуацию, и когда вы арендуете VDS-сервер для своего сайта - там графического интерфейса не будет.

8.1. Текстовые редакторы

Когда-то очень давно, еще в 70-ые годы прошлого века был единственный текстовый редактор - vi. Интерфейс (если можно назвать это интерфейсом) данного редактора довольно сложный, особенно для современных пользователей, которые привыкли к редакторами вроде MS Word.

Самое интересное, что данный редактор дожил до наших дней и используется до сих пор, как редактор по умолчанию в консоли. Поэтому рано или поздно у вас появится необходимость работать с этим редактором. Конечно,

можно переопределить переменную окружения EDITOR, указав в ней другой, более удобный текстовый редактор. Но это уже дело вкуса.

Редактор vi может работать в трех режимах:

- основной (визуальный) режим — в нем и осуществляется редактирование текста;
- командный режим — в нем выполняется ввод специальных команд для работы
- с текстом (если сравнивать vi с нормальным редактором, то этот режим ассоциируется с меню редактора, где есть команды вроде «сохранить», «выйти» и т. д.);
- режим просмотра — предназначен только для просмотра файла (если надумаете использовать этот режим, вспомните про команду less).

Синтаксис вызова vi следующий:

```
vi [опции] файл
```

Если опции не указаны, то файл будет загружен. Если указать опцию -e, то файл будет загружен, а редактор перейдет сразу в командный режим. Если файл не существует, то будет создан пустой файл.

После запуска vi главное знать, как из него выйти. В нем не будет знакомой строчки меню, также редактор не будет реагировать на привычные комбинации клавиш вроде <Alt>+<X>, <Ctrl>+<C>. На рис. 8.1 представлен редактор vi, в который загружен файл /etc/fstab.

Рассмотрим команды редактора vi:

- :q! - выход без сохранения.
- :w - сохранить изменения.
- :w <файл> - сохранить изменения под именем <файл>.
- :wq - сохранить и выйти.
- :q - выйти, если нет изменений.
- i - перейти в режим вставки символов в позицию курсора.
- a - перейти в режим вставки символов в позицию после курсора.
- o - вставить строку после текущей.
- O - вставить строку над текущей.

- x - удалить символ в позицию курсора.
- dd - удалить текущую строку.
- u - отменить последнее действие.

```
ivan@ivan-home: ~
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=0a23abd6-d721-4def-a56c-c7a7457160ab /               ext4    errors=remount
-ro 0       1
/swapfile                                 none            swap    sw
  0       0
/dev/fd0        /media/floppy0  auto    rw,user,noauto,exec,utf8 0       0
/dev/sdb1       /mnt/sdb1               ext4    rw,exec 0       0
~
~
~
~
~
~
~
~
~
~
"/etc/fstab" [readonly] 12 lines, 690 characters
```

Рис. 8.1. Редактор vi

Команды, которые начинаются с двоеточия, будут отображены в нижней строке, остальные просто выполняются, но не отображаются. Как уже было отмечено, у редактора vi есть два основных режима (режим просмотра не считается) — режим команд и режим редактирования (визуальный). Переключение в режим команд осуществляется нажатием клавиши <Esc>. Нажатие клавиш <i>, <a> и др. переключает редактор в режим вставки, когда набираемые символы трактуются именно как символы, а не как команды.

Чтобы переключиться обратно в командный режим используется клавиша <Esc>. В некоторых случаях (например, когда вы пытаетесь передвинуть курсор левее первого символа в строке) переход в командный режим осуществляется автоматически.

Теперь немного практики, введите команду:

```
$ vi test.txt
```

Далее нажмите клавишу <i>, чтобы переключиться в режим вставки. Введите любой текст, но постарайтесь не ошибаться, поскольку исправление ошибок в vi — дело, требующее отдельного разговора.

Затем нажмите клавишу <Esc> и введите :wq. После выхода из редактора введите команду:

```
cat test.txt
```

Убедитесь, что файл создан и в нем сохранен введенный вами текст.

Если ввести не команду i, а команду a, то вы тоже перейдете в режим вставки, но с одним отличием. Введенный текст будет вставляться не перед символом, в котором находится курсор, а после него. Также в режим вставки можно перейти командами **o** и **O**. В первом случае будет добавлена пустая строка после текущей строки, а во втором — перед текущей строкой, а весь дальнейший ввод будет восприниматься именно как ввод текста, а не команд.

Для удаления символа нужно перейти в режим команд и над удаляемым символом нажать <x>. Клавиши <Backspace> и <Delete> тут не работают. Точнее, <Backspace> работает, но для удаления последней непрерывно введенной последовательности символов.

Для удаления строки, в которой находится курсор, нужно использовать команду dd. Помните, что vi считает строкой не то, что вы видите на экране, а последовательность символов до первого символа новой строки (\n). Если строка длиннее 80 символов, то она переносится на две экранных строки и визуально выглядит как две строки, а не как одна.

Чтобы перейти в конец строки (клавиши <Home> и <End> тоже не работают, как вы успели заметить, если уже запускали vi), нужно ввести команду $. При этом курсор перемещается не по экранным линиям, а как раз по строкам текста.

Для отмены последней операции используется команда u. Вот только истории изменений нет, да и по команде u отменяется вся предыдущая команда целиком. Например, вы создали файл, перешли в режим вставки (команда i) и ввели огромный фрагмент текста. Если вы введете команду u, то она отменит всю предыдущую команду, т. е. удалит весь введенный вами текст. Поэтому будьте осторожны.

Думаю, вы убедились, что vi - самый неудобный редактор. Но не беда. Есть много других замечательных и очень удобных редакторов. Например, редактор mcedit, входящий в состав файлового менеджера mc. Именно он запускается, когда вы редактируете файл в mc (при нажатии F4), см. рис. 8.2.

Еще один очень удобный редактор - joe. Он не устанавливается по умолчанию, поэтому для его установки нужно установить одноименный пакет. Сам редактор отображен на рис. 8.3.

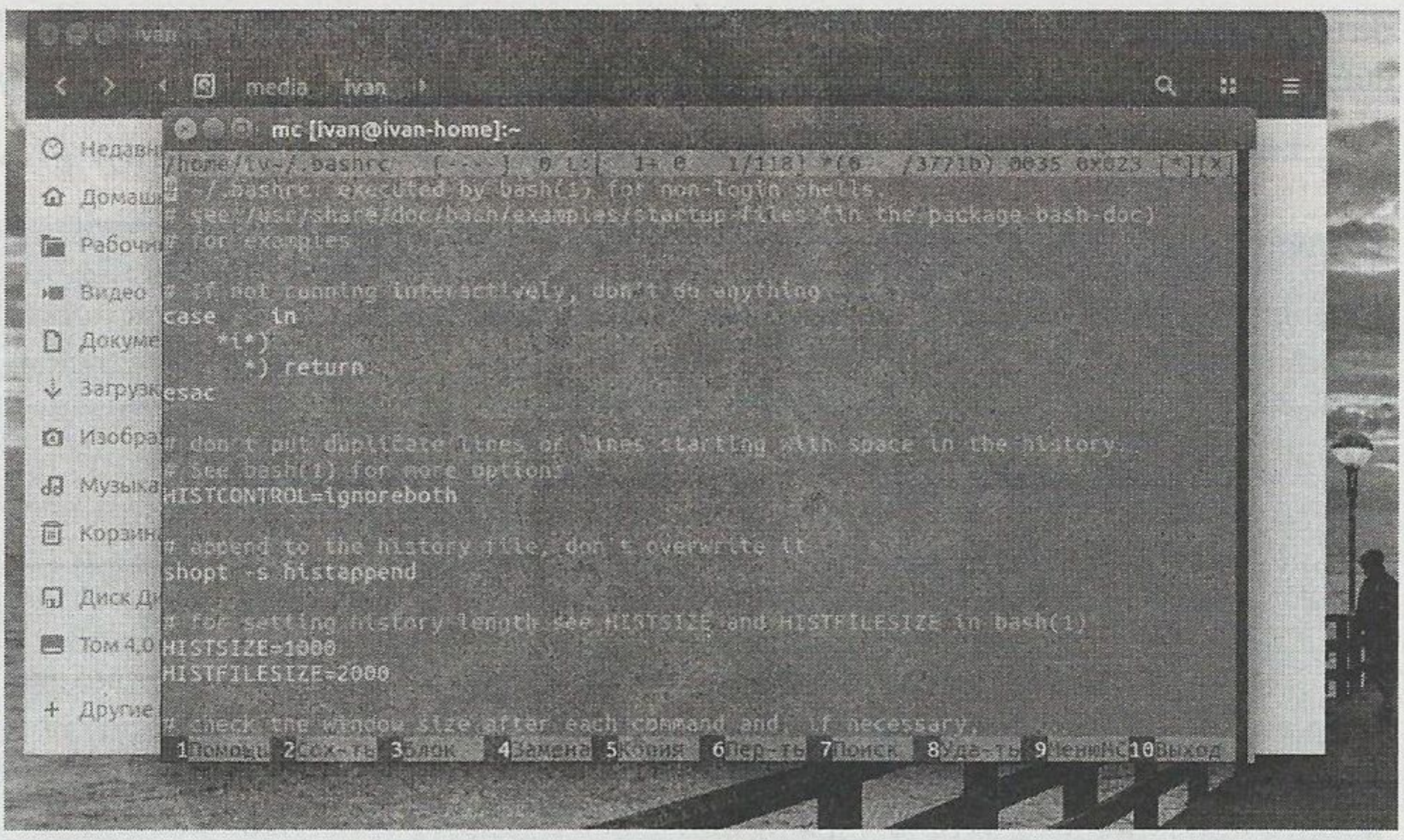


Рис. 8.2. Текстовый редактор файлового менеджера mc

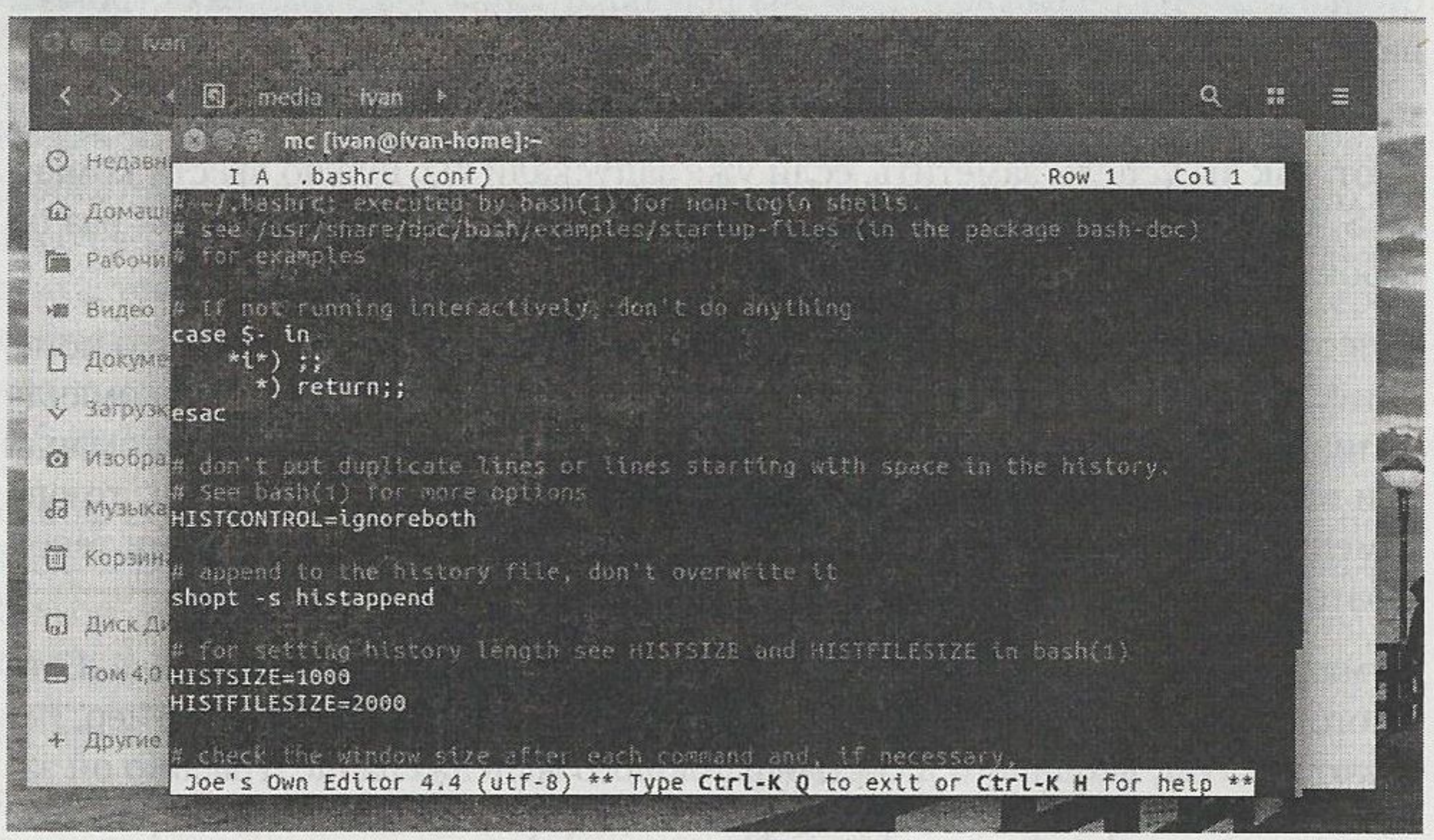


Рис. 8.3. Текстовый редактор joe

Существуют и другие редакторы - nano, pico, ee. Все они будут гораздо удобнее, чем vi. Попробуйте установить их и выбрать лучший для себя. Потом не забудьте прописать выбранный вами редактор в переменной окружения EDITOR.

Установить переменную editor можно так:

```
export EDITOR=joe
```

Вместо joe, разумеется, можно указать любой текстовый редактор. Данную строку нужно добавить в ваш файл ~/.bashrc. Затем выйти из системы (выполнить команду exit) и снова зайти (или закрыть и открыть окно терминала).

8.2. Статистика: подсчет символов, слов, строк

Команда wc используется:

для подсчета слов в текстовом файле:

```
wc file.txt
```

для подсчета количества строк (если задан параметр -l):

```
wc -l file.txt
```

для подсчета количества символов (параметр -c):

```
wc -c file.txt
```

8.3. Потоковый текстовый редактор

Программа sed — мощный потоковый редактор, и его описанию можно было бы посвятить целую книгу. Пусть маленькую, но книгу. Подробно мы не будем рассматривать эту команду - при желании информацию можно найти на страницах руководства или в Интернете (если в дистрибутиве man на английском). Рассмотрим несколько примеров:

- заменить строку string1 на string2 в файле file.txt, результат вывести на стандартное устройство вывода (экран):

```
sed 's/string1/string2/g' file.txt
```

То же самое, но перенаправить вывод в file2.txt:

```
sed 's/string1/string2/g' file.txt > file2.txt
```

- вывести третью строку файла file.txt:

```
sed -n '3p;3q' file.txt
```

- удалить пустые строки из файла:

```
sed '/^$/d' file.txt
```

- удалить строку string1 из текста, не изменяя всего остального:

```
sed -e 's/string1//g' file.txt
```

- удалить пустые символы в конце каждой строки:

```
sed -e 's/ *$//' file.txt
```

- удалить пустые строки и комментарии из файла:

```
sed '/ *#/d; /^$/d' file.txt
```

- удалить первую строку из файла:

```
sed -e '1d' result.txt
```

8.4. Сравнение двух файлов

Команда cmp используется для сравнения двух файлов. Если файлы идентичны, то cmp вообще никак не реагирует. А если файлы отличаются, то cmp выводит номер строки и номер символа в строке, откуда начинается различие.

Команда cmp более универсальна, поскольку она может использоваться как для сравнения текстовых, так и двоичных файлов. А вот команда diff и ее аналоги умеют сравнивать только текстовые файлы.

Формат вызова команды следующий:

```
cmp [параметры] файл1 файл2
```

Параметр -c обеспечит вывод всех отличающихся символов. Параметр -i n позволит проигнорировать первые n символов. Параметр -l выведет все отличия, а не только первое.

Параметр -s удобно использовать в сценариях. На экран информация не будет выводиться, зато будет возвращен код возврата: 0, если файлы одинаковые, 1, если файлы отличаются, и 2, если произошла ошибка при открытии одного из файлов.

Команда comm сравнивает содержимое двух файлов, которые были перед этим отсортированы командой sort. Вывод программы располагается в три столбца. В первом выводятся строки из файла1, во втором — из файла2, а в третьем — строки, которые имеются в обоих файлах.

Формат вызова программы следующий:

```
comm [параметры] файл1 файл2
```

Описание параметров программы может быть найдено в справочной системе (man comm).

Команда diff используется для сравнения двух файлов. Формат вызова программы diff:

```
diff [параметры] файл1 файл2
```

В выводе программы отличающиеся строки помечаются символами > и <:

- строка из первого файла помечается символом <;
- строка из второго файла — символом >.

Самые полезные параметры программы diff приведены в табл. 8.1.

Таблица 8.1. Параметры программы diff

Параметр	Описание
-a	Сравнение всех файлов, в том числе бинарных
-b	Программа будет игнорировать пробельные символы в конце строки
-B	Игнорирует пустые строки
-e	Применяется для создания сценария для редактора ed, который будет использоваться для превращения первого файла во второй
-w	Игнорирует пробельные символы

-y	Вывод в два столбца
-r	Используется для сравнения файлов в подкаталогах. Вместо первого файла указывается первый каталог, вместо второго файла — соответственно второй каталог

8.5. Текстовый фильтр

Команда grep производит поиск строки в одном или нескольких файлах. Но обычно ее используют в другом контексте, например, у нас есть большой файл или вывод какой-то программы и нам нужно найти какую-то строку. Пример:

```
dmesg | grep
```

Команда egrep похожа на команду grep (с которой вы уже знакомы), но считается более быстрой и более функциональной. Если файлы не заданы, то программа читает текст из стандартного ввода.

Формат вызова программы:

```
egrep [параметры] строка файлы
```

Параметры команды egrep приведены в табл. 8.2.

Таблица 8.2. Параметры команды egrep

Параметр	Описание
-A n	Выводит n строк после строки, в которой есть искомая строка
-B n	Выводит n строк перед строкой, содержащей искомую строку
-b	Выводит для каждой строки файла, где есть искомая строка, ее положение в файле
-c	Вывод количества совпадений, но не выводит сами совпадения
-C	Возвращает две строки до и две строки после строки, которая содержит искомую строку

-f файл	Поиск искомых строк, которые имеются в указанном файле
-h	Выводит строки, содержащие искомую строку, но не выводит имена содержащих их файлов
-i	Игнорировать регистр букв
-n	Выводит номера строк (и сами строки), содержащих искомую строку
-s	Не выводить сообщения об ошибке, если некоторые файлы не могут быть открыты
-w	Поиск совпадения целого слова с искомой строкой
-x	Поиск совпадения целой строки с искомой строкой

8.6. Форматирование текста

Команда fmt форматирует текст, выравнивает его по правой границе и удаляет символы новой строки. Синтаксис вызова команды:

```
fmt [параметры] файлы
```

Параметр -t заставит программу начинать параграф с красной строки. Параметр -w n задает максимальную ширину (длину) строки в n символов. Значение по умолчанию - 72.

Параметр -c запрещает форматировать первые строки, а параметр -p <префикс> отключает форматирование строк, которые начинаются с указанного префикса.

8.7. Замена символов табуляции пробелами

Команда expand заменяет в указанных файлах символы табуляции на соответствующее количество пробелов. Команде можно передать лишь один параметр -i, означающий, что замена должна быть только в начале строки.

Формат вызова команды:

```
expand [-i] файлы
```

Есть и обратная команда - unexpand. Она заменяет пробелы символами табуляции. Количество пробелов можно задать с помощью параметра -t n (где n — количество пробелов).

Синтаксис вызова:

```
unexpand [параметры] файл
```

8.8. Разбивка текста на столбцы

Команда column используется для разбивки текста на несколько столбцов. Текст может быть прочитан как из файла, так и со стандартного ввода, если файл не указан.

Формат вызова команды:

```
column [параметры] [файл]
```

Параметр -c n позволяет задать количество столбцов (n - это то самое количество). Параметр -s <символ> позволяет задать разделитель столбцов.

8.9. Выравнивание текста по правому краю

Команда fold выравнивает текст по правому краю. Производит разрыв строк, если это необходимо. Максимальная длина строки — 80 символов. Установить другую длину строки можно с помощью параметра -w, как и в случае с командой fmt. Кроме того, можно указать параметр -s, разрешающий разрыв строки только на пробеле.

Формат вызова команды:

```
fold [параметры] файлы
```

8.10. Вывод начальных и последних строк файла

Команды head и tail позволяют вывести, соответственно, начальные и последние строки файла. При этом параметр -n задает количество строк. Пример:

```
head -n 10 file.txt
tail -n 10 file.txt
```

8.11. Сортировка файлов

Команда sort сортирует все указанные файлы, результат сортировки всех указанных файлов отправляется на стандартный вывод.

Формат вызова команды:

```
sort [параметр]... [файлы]
```

Параметр -f позволяет игнорировать регистр букв. Параметр -r обеспечивает сортировку в обратном порядке. Параметр -o <файл> позволяет вывести результат сортировки в файл.

8.12. Разбивка текстовых файлов на несколько частей

С помощью команды split можно разбить файл на несколько частей. По умолчанию команда создает части размером в 1000 строк, но вы можете изменить это количество, например:

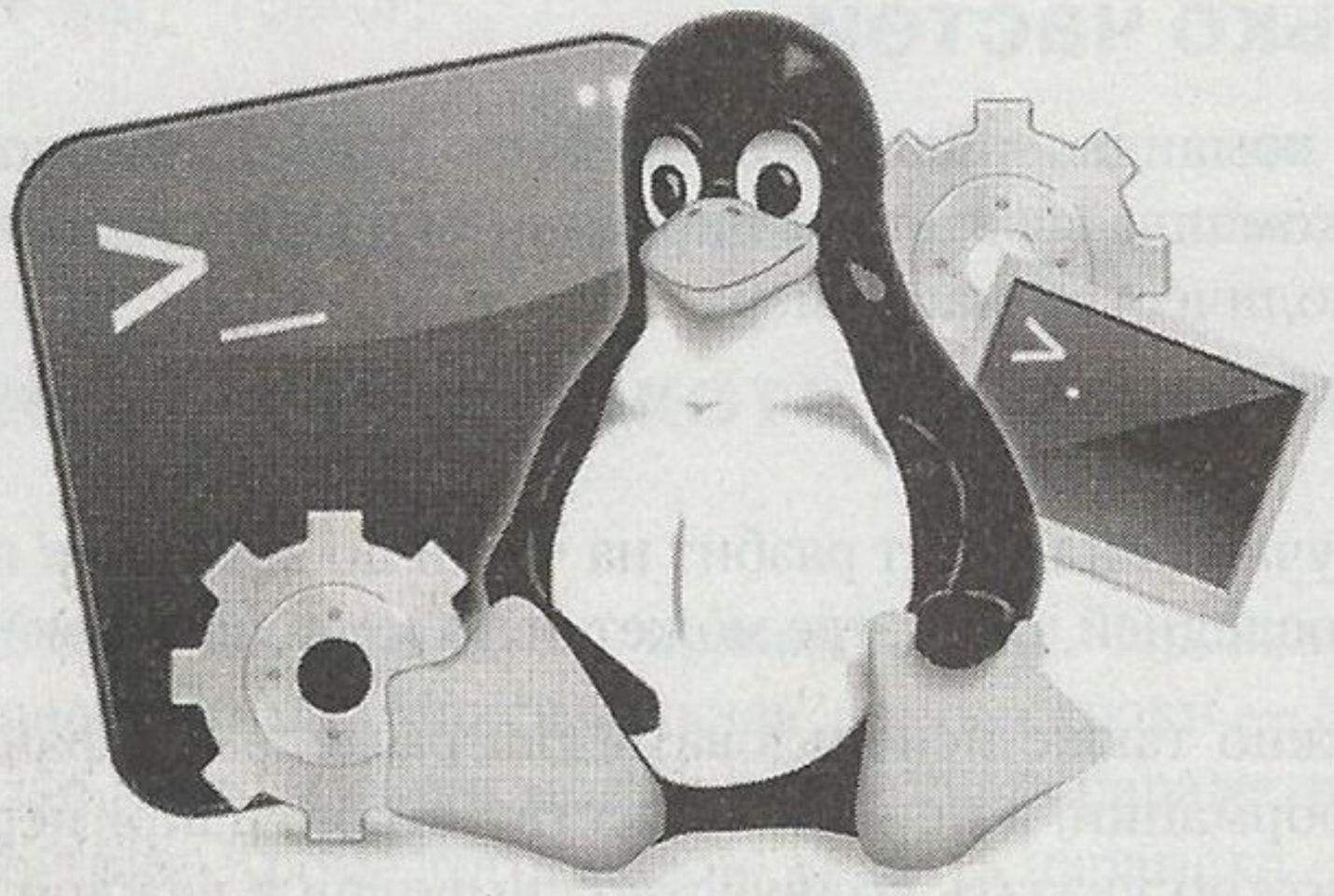
```
split -200 book.txt
```

В данном случае файл будет разбит на части по 200 строк в каждой (кроме, возможно, последней части, где может быть меньше строк).

Команду можно также использовать для разделения файлов на части по размеру информации, а не по количеству строк, например с помощью параметра -b можно указать количество символов в каждой части. Примеры вызова команды:

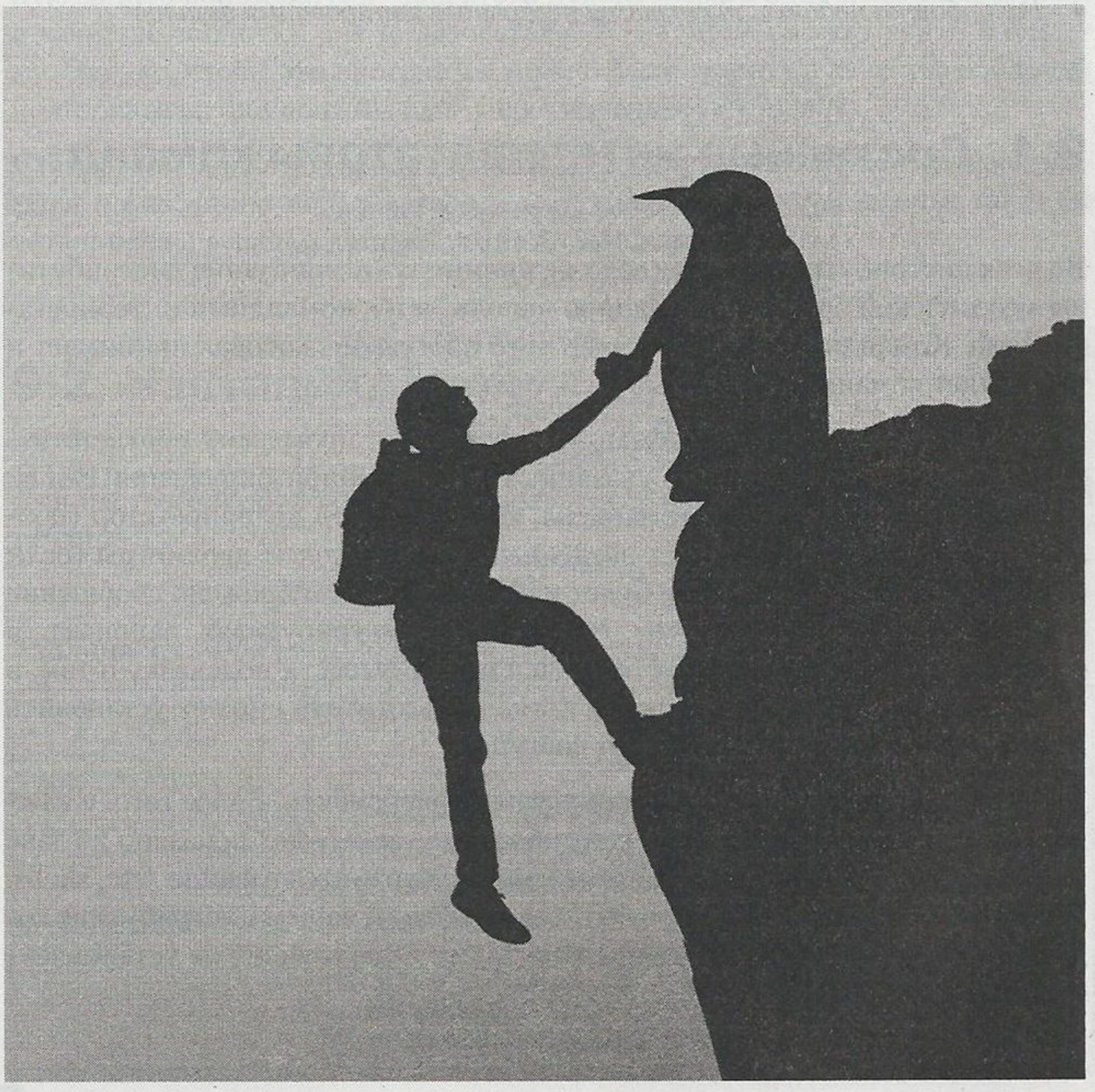
```
split -b100b book.txt
split -b100k book.txt
split -b100m book.txt
```

Первая команда разделит файл на части по 100 байтов каждая, вторая — на части по 100 Кбайт каждая, третья — по 100 Мбайт каждая.



ГЛАВА 9.

Сценарии командной строки



В этой главе:

- *Автоматизация задач с помощью сценариев оболочки*
- *Примеры реальных сценариев*
- *Разработка небольшой системы резервного копирования файлов*

9.1. Доступные интерпретаторы команд

Во всех дистрибутивах Linux (если не принимать во внимание дистрибутивы мезозойской эры) по умолчанию используется командный интерпретатор bash. Командный интерпретатор - это программа, которая принимает и выполняет команды, введенные пользователем.

Когда пользователь вводит команду, то bash ищет программу, соответствующую команде, в каталогах, указанных в переменной окружения PATH. Если такая программа будет найдена, то командный интерпретатор (оболочка) запустит ее и передаст введенные пользователем параметры (если они есть). Если программа не будет найдена, будет отображено сообщение об отсутствии этой программы. В некоторых дистрибутивах, например, в современных версиях Ubuntu, оболочка может вывести подсказку о том, в каком пакете находится команда - тогда пользователь сможет установить нужный пакет и программа станет доступной.

В системе может быть установлено несколько оболочек. Более того, у каждого пользователя может быть своя оболочка - как будет показано в главе 11. Список установленных в системе оболочек хранится в файле /etc/shells. В листинге 9.1 приводится пример этого файла. В вашем дистрибутиве содержимое файла может отличаться, поскольку оно зависит от установленных в системе оболочек.

Листинг 9.1. Файл /etc/shells

```
/bin/ash
/bin/bash
/bin/csh
/bin/false
/bin/ksh
/bin/sh
/bin/tcsh
/bin/true
/bin/zsh
```

Программы /bin/true и /bin/false не являются оболочками. Это своеобразные «болванки», с помощью которых можно отключить ту или иную учетную запись пользователя - для этого нужно указать одну из этих программ в качестве оболочки для пользователя. Тогда он не сможет войти в систему. Однако, чтобы эти программы можно было указать в качестве оболочки пользователя, они должны быть перечислены в /etc/shells.

Стандартом является оболочка /bin/bash. Другие оболочки устанавливаются по желанию пользователя и могут быть, как установлены, так и не установлены в вашем дистрибутиве по умолчанию.

9.2. Настройка bash

Для настройки оболочки bash используется ряд файлов, перечисленных в таблице 9.1.

Таблица 9.1. Файлы конфигурации оболочки bash

Файл	Описание
/etc/profile	Содержит глобальные настройки, влияющие на всю систему. Как правило, этот файл не редактируется, а вместо этого редактированию подвергаются файлы, находящиеся в домашнем каталоге пользователя ~. Но при желании вы можете отредактировать этот файл - тогда настройки повлияют на всех пользователей в системе.

~/.bash_profile	Обрабатывается при каждом входе в систему. В этот файл можно добавить команды, которые будут выполнены при каждом входе в систему конкретного пользователя. Сейчас этот файл практически не используется, а вместо него настройки предлагается хранить в ./bashrc.
~/.bashrc	Обрабатывается при каждом запуске дочерней оболочки. Как правило, это происходит один только раз - при входе в систему, но пользователь может запустить дочернюю оболочку.
~/.bash_logout	Обрабатывается при выходе пользователя из системы.

Теперь давайте разберемся, какие настройки могут быть в этих файлах. Можно, например, добавить вызов какой-то программы, например, mc. Но если вам нужно запускать файловый менеджер при каждом входе в систему, проще его установить как оболочку пользователя. Так даже будет правильнее. Обычно файлы конфигурации используются для определения псевдонимов команд и задания значений переменных окружения.

Задать псевдоним команд можно с помощью команды alias, например:

```
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
```

Псевдонимы позволяют сократить основную команду. Например, в этом случае при вводе команды l будет выполнена команда ls -CF.

Переменные окружения устанавливаются так:

```
export Имя_переменной = Значение
```

Например:

```
export GCC_COLORS='error=01;31:warning=01;35:note=01;36:car
et=01;32:locus=01'
```

Также в этом файле можно задать внешний вид приглашения командной строки. Переменная PS1 отвечает за внешний вид командной строки. По умолчанию командная строка имеет формат:

```
пользователь@компьютер:рабочий_каталог
```

Значение PS1 при этом будет:

```
PS1='\u@\h:\w$'
```

Допустимые значения, которые можно использовать для задания внешнего вида командной строки, приведены в таблице 9.2.

Таблица 9.2. Модификаторы командной строки

Модификатор	Описание
\a	ASCII-символ звонка (код 07).
\d	Дата в формате «день недели, месяц, число»
\h	Имя компьютера до первой точки
\H	Полное имя компьютера
\j	Количество задач, запущенных в оболочке в данное время
\l	Название терминала
\n	Символ новой строки
\r	Возврат каретки
\s	Название оболочки
\t	Время в 24-часовом формате (ЧЧ: ММ: СС)
\T	Время в 12-часовом формате (ЧЧ: ММ: СС)
\@	Время в 12-часовом формате (AM/PM)
\u	Имя пользователя
\v	Версия bash (сокращенный вариант)

\V	Версия bash (полная версия: номер релиза, номер патча)
\w	Текущий каталог (полный путь)
\W	Текущий каталог (только название каталога, без пути)
\!	Номер команды в истории
\#	Системный номер команды
\$	Если UID пользователя равен 0, будет выведен символ #, иначе — символ $
\\	Обратный слэш
$ ( )	Подстановка внешней команды

9.3. Введение в сценарии bash

Командная оболочка позволяет писать сценарии - небольшие программы, которые будут выполняться самим командным интерпретатором. Синтаксис таких сценариев отличается для каждой оболочки. Далее в этой главе мы будем рассматривать только синтаксис оболочки bash.

Сценарии позволяют автоматизировать выполнение рутинных задач. Чтобы не вводить множество команд, их объединяют в сценарий. Потом запускают сценарий (вводят одну команду), который выполняет все указанные в нем команды.

Кроме того, синтаксис поддерживает базовые структуры вроде циклов, условных операторов. Можно написать более или менее полноценную программу, не прибегая к использованию «взрослых» языков программирования вроде C. Синтаксис bash с одной стороны чем-то напоминает C, а с другой стороны проще, поскольку нет указателей, структур и тому подобных вещей.

Напишем первый сценарий. Для этого создайте файл first и отредактируйте его:

```
touch first
gedit first
```

Содержимое этого файла приведено в листинге 9.2.

Листинг 9.2. Сценарий first

```
#!/bin/bash
echo «Мой первый сценарий»
```

Первая строка нашего сценария — это указание, что он должен быть обработан программой /bin/bash. Обратите внимание: если между # и ! окажется пробел, то данная директива не сработает, поскольку будет воспринята как обычный комментарий. Комментарии начинаются с решетки, после которой следует пробел:

```
# Комментарий
```

Сделаем файл исполнимым (чтобы его можно было запускать):

```
chmod +x first
```

А теперь запустим:

```
./first
```

Результат выполнения сценария приведен на рис. 9.1. Как и ожидалось, он просто вывел строку «Мой первый сценарий».

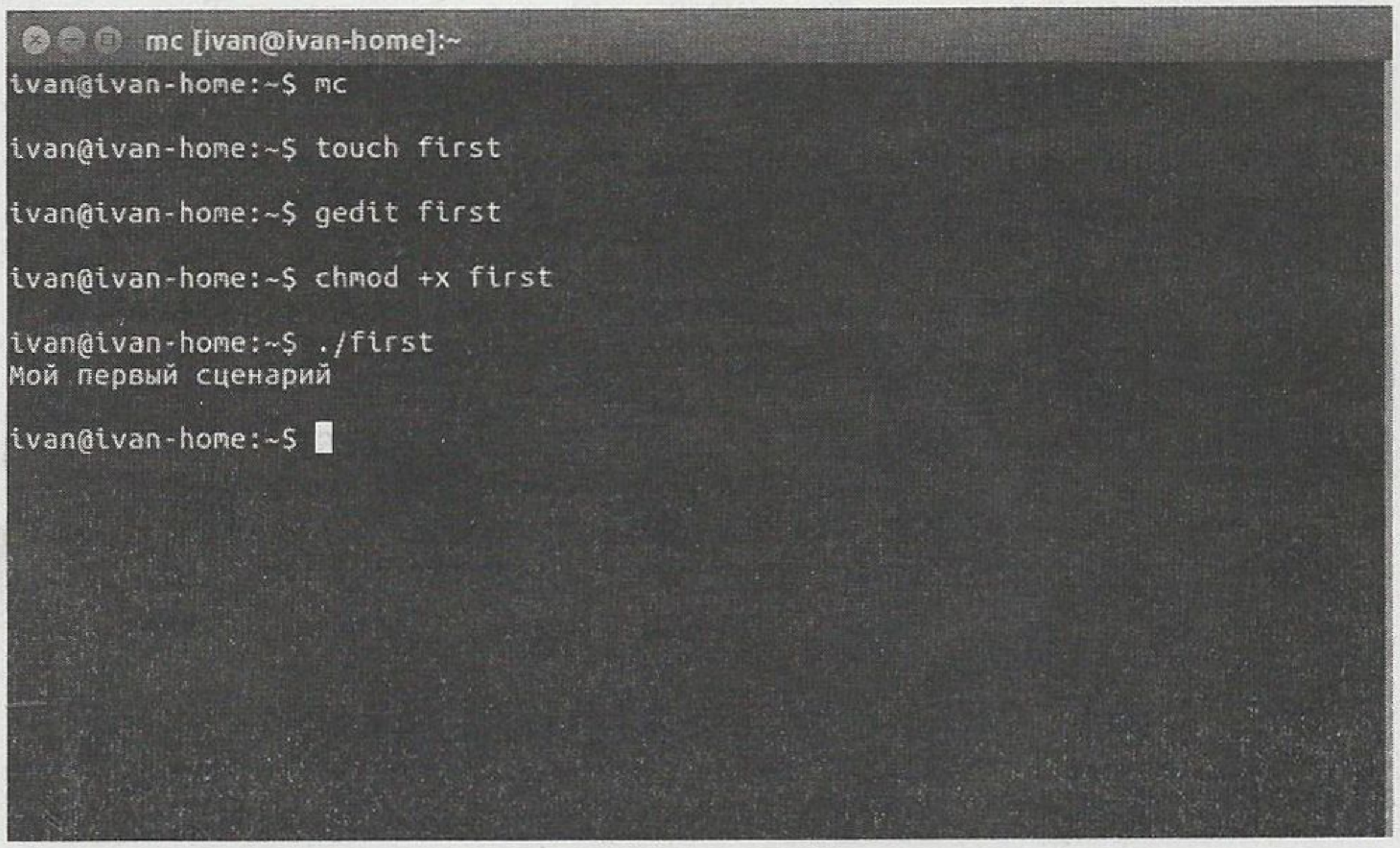


Рис. 9.1. Результат выполнения сценария

Чтобы вводить для запуска сценария просто first (без ./), сценарий нужно скопировать в каталог /usr/bin (точнее, в любой каталог из переменной окружения PATH):

```
sudo cp ./first /usr/bin
```

Я рекомендую не захламлять /usr/bin вашими сценариями, а создать для них отдельный каталог (например, /opt/my) и добавить его в список каталогов PATH:

```
PATH="/opt/my:$PATH"
export PATH
```

Эти строки нужно добавить в ваш файл ~/.bashrc.

9.4. Переменные окружения

В Linux часто используются переменные окружения. Это специальные переменные, содержащие служебные данные, например, переменная PATH содержит список каталогов, в которых командный интерпретатор производит поиск введенной команды.

Примеры некоторых часто используемых переменных окружения:

- BASH – полный путь до исполняемого файла командной оболочки bssh;
- BASH_VERSION – версия bash;
- HOME – домашний каталог пользователя, который запустил сценарий;
- HOSTNAME – имя компьютера;
- RANDOM – случайное число в диапазоне от 0 до 32 767;
- OSTYPE – тип операционной системы;
- PWD – текущий каталог;
- PS1 – строка приглашения;
- UID – ID пользователя, который запустил сценарий;
- USER – имя пользователя.

Для установки собственной переменной окружения, как уже отмечалось, используется команда export:

```
export PATH=<значение>
```

9.5. Собственные переменные

В любой программе есть переменные. Сценарии bash - не исключение. Переменные можно объявлять в любом месте сценария, но до места их первого использования. Рекомендуется объявлять переменные в самом начале сценария, чтобы потом не искать, где вы объявили ту или иную переменную.

Для объявления переменной используется следующая конструкция:

```
переменная=значение
```

Пример объявления переменной:

```
NAME="ivan"
echo $NAME
```

Обратите внимание, что не должно быть никаких пробелов до и после знака равенства. Также при объявлении переменной знак доллара не используется, зато используется при обращении к значению переменной. Это вносит некоторую путаницу - лучше бы знак доллара использовался всегда.

Вы можете объявить даже массивы, например:

```
ARR [0]=1
ARR [1]=2

echo $ARR [0]
```

9.6. Передача аргументов сценарию

Сценариям можно передавать аргументы - параметры. Обратиться к переданному аргументу можно с помощью конструкции $n, где n - номер параметра. Нумерация начинается с 1, поскольку 0 - это имя самого сценария.

Количество переданных аргументов содержится в $#. $@ содержит все аргументы, переданные скрипту, разделенные пробелами.

Вывести все переданные аргументы можно так:

```
for n in $@
do
  echo "$n"
done
```

А вот как можно проанализировать первый параметр с помощью конструкции case-esac:

```
# Анализируем первый параметр
case «$1» in
  help)
   # Действия при получении параметра help
   echo «Выводим справку»
   ;;
  post)
   # Действия при получении параметра post
   echo «Отправляем e-mail сообщение»
   ;;
*)
   # Действия в остальных случаях
   # Выводим подсказку об использовании сценария, и
   # завершаем работу
      echo "Usage: $0 {help|post }"
   exit 1
   ;;
esac
```

Если вы программировали ранее, то принцип работы конструкции case-esac должен быть вам знаком. Если же нет, то, смысл должен быть интуитивно понятен.

9.7. Условные операторы

В bash доступно два условных оператора, if и case-esac. Синтаксис оператора if следующий:

```
if условие_1 then
   команды_1
elif условие_2 then
   команды_2
...
elif условие_N then
   команды_N
else
   команды_N+1
fi
```

Оператор if в bash работает аналогично оператору if в других языках программирования. Если истинно первое условие, то выполняется первый список команд, иначе — проверяется второе условие и т. д. Количество блоков elif, понятно, не ограничено.

Самая ответственная задача — это правильно составить условие. Условия записываются в квадратных скобках. Вот пример записи условий:

```
# Переменная N = 5
[ N==5 ]

# Переменная N не равна 5
[ N!=5 ]
```

Операции сравнения указываются не с помощью привычных знаков > или <, а с помощью следующих выражений:

- -lt — меньше;
- -gt — больше;
- -le — меньше или равно;
- -ge — больше или равно;
- -eq — равно (используется вместо ==).

Применять данные выражения нужно следующим образом:

```
[ переменная выражение значение|переменная ]
```

Например:

```
# N меньше 10
[ $N -lt 10 ]
# N меньше A
[ $N -lt $A ]
```

В квадратных скобках вы также можете задать выражения для проверки существования файла и каталога:

- -e файл — условие истинно, если файл существует;
- -d каталог — условие истинно, если каталог существует;
- -x файл — условие истинно, если файл является исполнимым.

С оператором case мы уже немного знакомы, но сейчас рассмотрим его синтаксис подробнее:

```
case переменная in
значение_1) команды_1 ;;
...
значение_N) команды_N ;;
*) команды_по_умолчанию;;
esac
```

Значение указанной переменной по очереди сравнивается с приведенными значениями (значение_1 ... значение_N). Если есть совпадение, то будут выполнены

команды, соответствующие значению. Если совпадений нет, то будут выполнены команды по умолчанию. Один из примеров использования case был приведен ранее. Рассмотрим еще один - обработка пользовательского ввода (лист. 9.3).

Листинг 9.3. Обработка ввода пользователя

```
#!/bin/bash

echo -n «Продолжить? (y/n) «

read ans
case "$ans" in
    y|Y) echo "Ввели «y», продолжаем..."
        ;;
```

```
    n|N) echo «Ввели «n», завершаем...»
         exit 0
         ;;
    *) echo «Выполняем действие по умолчанию...»
         ;;
esac
```

Мы читаем ввод (с помощью инструкции read) пользователя в переменную ans, а затем выполняем действия в зависимости от ее значения.

9.8. Циклы

В bash можно использовать циклы. Мы рассмотрим циклы for и while, хотя вообще в bash доступны также циклы until и select, но они применяются довольно редко.

Синтаксис цикла for выглядит так:

```
for переменная in список
do
команды
done
```

В цикле при каждой итерации переменной будет присвоен очередной элемент списка, над которым будут выполнены указанные команды. Чтобы было понятнее, рассмотрим небольшой пример:

```
for n in 1 2 3;
do
echo $n;
done
```

Обратите внимание: список значений и список команд должны заканчиваться точкой с запятой.

Как и следовало ожидать, наш сценарий выведет на экран следующее:

```
1
2
3
```

Вот еще пример использования цикла for. Он выводит все файлы *.txt, находящиеся в текущем каталоге:

Листинг 9.4. Вывести все файлы *.txt из текущего каталога

```
#!/bin/bash
for d in `ls *.txt`; do
  echo $d
done
```

Особого смысла в этом сценарии нет, но вместо echo можно передать файл на обработку, например, передать команде sort, которая отсортирует файл. А если у нас есть мультимедиа файлы, то можно передать их программе, которая будет воспроизводить их, например:

```
for d in `ls *.wav`; do
  aplay $d
done
```

Синтаксис цикла while выглядит немного иначе:

```
while условие
do
команды
done
```

Цикл while выполняется до тех пор, пока истинно заданное условие. Выведем 1 2 3 с помощью цикла while:

```
n=1
while [ $n -lt  4 ]
do
  echo "$n "
  n=$(( $n+1 ));
done
```

9.9. Пользовательские функции

В bash можно использовать функции. Синтаксис объявления функции такой:

```
имя() { список; }
```

Вот пример объявления и использования функции:

```
list_txt()
{
echo «Выводим текстовые файлы»
ls *.txt
}
```

9.10. Практические примеры

В этом разделе мы рассмотрим несколько практических примеров сценариев. Сценарии будут иметь практическую ценность, а не будут созданы ради демонстрации возможностей bash.

9.10.1. Переименование файлов

Начнем со сценария переименования файлов. Данный сценарий ищет в текущем каталоге файлы, в именах которых есть пробелы. Все найденные пробелы он заменяет символами подчеркивания.

Листинг 9.5. Сценарий remove_spaces

```
#!/bin/bash

num=0                  # Сколько файлов было переименовано

for filename in *  # Перебираем все файлы в текущем каталоге
do
# Передаем имя файла фильтру grep
# Если имя файла содержит пробел, то код завершения
# последней операции равен 0
    echo «$filename» | grep -q « «
```

```
    if [ $? -eq 0 ]
    then
# Если код завершения равен 0, переименовываем
# файл
      fname=$filename
      n=`echo $fname | sed -e "s/ /_/g"`
      mv "$fname" "$n"
      let "num += 1"
    fi
done

echo "Переименовано файлов: $num"

exit 0
```

Использовать сценарий нужно так. Первым делом сделайте его исполнимым и переместите в один из каталогов пути PATH. Далее перейдите в каталог, файлы которого нужно переименовать и запустите сценарий:

```
cd ~/Music
remove_spaces
```

Результат выполнения сценария показан на рис. 9.2. На нем показано содержимое каталога Music до переименования и после.

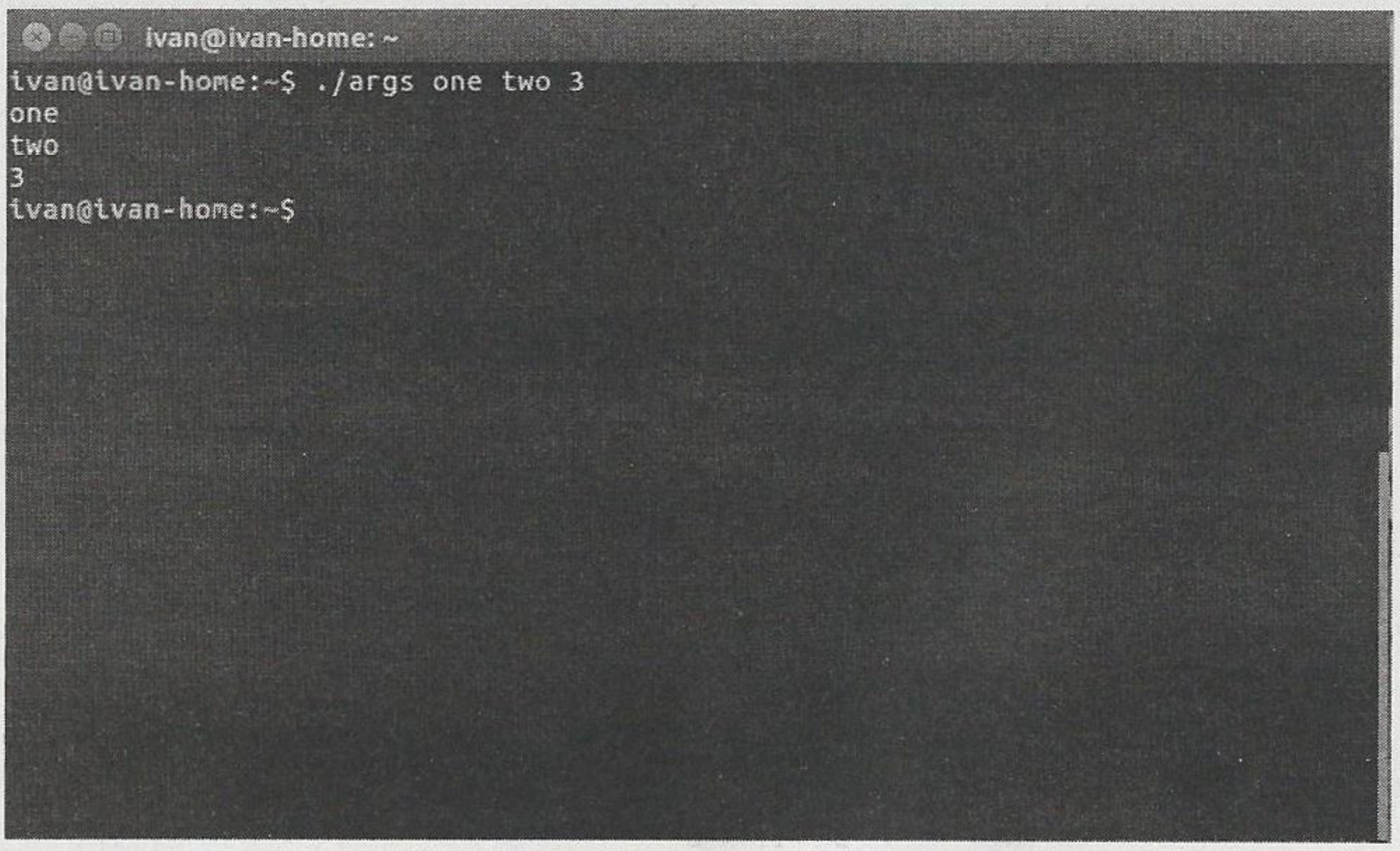


Рис. 9.2. Сценарий переименования файлов

9.10.2. Мониторинг системы

Напишем еще один сценарий, выполняющий мониторинг журнала syslog. Наш сценарий будет каждые 5 секунд выводить на экран последние 5 записей этого журнала. Для выхода из сценария нажмите Ctrl + C.

Листинг 9.6. Сценарий show_syslog

```
#!/bin/bash
# Интервал обновления, в секундах
INT=5

while [ true ]
do
# Выводим последние 5 строк журнала
tail -n 10 /var/log/syslog
# Ждем
sleep $INT
# Две пустые строки
echo; echo
done
```

Наш третий сценарий будет выполнять резервное копирование файлов журналов. В имена резервных копий сценарий вставляет дату создания копии. Сами же резервные копии сжимаются с помощью bzip и перемещаются в каталог, заданный переменной BKP_DIR.

Листинг 9.7. Сценарий backup_logs

```
#!/bin/bash

PROBLEM=66

# Здесь укажите ваш каталог для архивации.
BKP_DIR=/opt/backups/weblogs

# Настройки Apache2 по-умолчанию
LOG_DAYS="4 3 2 1"
LOG_DIR=/var/log/apache2
LOG_FILES="access_log error_log"

# Расположение программ, проверьте в своем дистрибутиве
LS=/bin/ls
MV=/bin/mv
```

```
ID=/usr/bin/id
CUT=/bin/cut
COL=/usr/bin/column
BZ2=/usr/bin/bzip2

# Проверка прав пользователя
USER=`$ID -u`
if [ «X$USER» != «X0» ]; then
 echo «PANIC: Только root может запускать этот сценарий!»
 exit $PROBLEM
fi

# Каталог для резервной копии существует и доступен на запись?
if [ ! -x $BKP_DIR ]; then
 echo «PANIC: Каталог $BKP_DIR не найден или не доступен для
записи!»
 exit $PROBLEM
fi

# Переместить, переименовать и сжать
for logday in $LOG_DAYS; do
 for logfile in $LOG_FILES; do
 MYFILE="$LOG_DIR/$logfile.$logday"
 if [ -w $MYFILE ]; then
 DTS=`$LS -lgo --time-style=+%Y%m%d $MYFILE | $COL -t |
$CUT -d ' ' -f7`
 $MV $MYFILE $BKP_DIR/$logfile.$DTS
 $BZ2 $BKP_DIR/$logfile.$DTS
 else
 # Выдать сообщение об ошибке, если файл недоступен на
запись.
 if [ -f $MYFILE ]; then
 echo «ОШИБКА: $MYFILE недоступен для записи. Пропуск."
 fi
 fi
 done

exit 0
```

9.10.3. Резервное копирование

Представим, что у нас есть веб-сервер хостинг-провайдера и нужно обеспечить его резервное копирование. Есть множество уже готовых систем резервного копирования, решающих поставленную задачу. Но в общеоб-

разовательных целях интересно все реализовать самостоятельно, поэтому будем использовать только стандартные средства, которые есть в любом дистрибутиве Linux. Что же касается уже готовых комплексов резервного копирования, то у каждого из них есть своя документация и множество статей в Интернете на русском языке. Поэтому не вижу смысла переписывать руководство еще раз другими словами.

Первым делом нужно настроить сервер, чтобы он работал в штатном режиме. Убедитесь, что настроили все и вам больше не придется вносить изменения в конфигурацию сервера. Будем считать, что пользовательские данные будут, как обычно, храниться в каталоге /home/<имя пользователя>. Базы данных пользователей, как обычно, хранятся в /var/lib/mysql/<имя>. Для упрощения нашей конфигурации будем считать, что одному пользователю будет принадлежать только одна база данных, которая совпадает с именем пользователя. Например, есть сайт example.com, он принадлежит пользователю examplecom, его HTML-код хранится в каталоге /home/examplecom/public_html, а база данных - в каталоге /var/lib/mysql/examplecom.

Сразу рекомендую настраивать RAID-массивы, чтобы была возможность горячей замены и вы могли на лету восстанавливать работоспособность сервера в случае выхода из строя одного из жестких дисков.

Позаботимся о создании резервной копии данных. Нам нужно копировать подкаталоги каталога /home и /var/lib/mysql. В первом хранятся HTML-код (и вспомогательные файлы) сайтов пользователей, а во втором - база данных. Разработаем первую версию нашего сценария backup. Сначала нужно создать сам сценарий:

```
# cd /bin
# touch backup
# chmod +x backup
# nano backup
```

Затем введите следующий код:

```
#!/bin/bash

rm -r /mnt/backup/*
cd /home

for hd in `ls /home`; do
  echo "Creating backup for $dh"
  tar -czf /mnt/backup/$hd.tar.gz $hd
done
```

Разберемся, что и к чему. Первым делом мы удаляем старые резервные копии (команда rm). Программа tar с параметром c создает новый архив, но если файл архива существует, он не будет перезаписан.

Чтобы добавить файлы в уже существующий архив можно использовать параметр u, но его нельзя использовать вместе с параметром c, поэтому нужно проверять существует ли файл архива. Если нет, тогда нужно создать архив (параметр c), если да, тогда нужно модифицировать архив (параметр u).

Далее мы создаем архив для каждого домашнего каталога пользователя. Например, если в /home у вас есть два подкаталога - ivan и eva, то будут созданы архивы ivan.tar.gz и eva.tar.gz.

Прошу обратить внимание: мы не делаем резервную копию всего /home. Тогда мы получим один большой архив, с которым будет очень неудобно работать (да и можем превысить ограничение на максимальный размер файла - все зависит от объемов данных, с которыми работают ваши пользователи), поэтому проще и правильнее создать несколько меньших архивов. Да и предоставить резервную копию по запросу пользователя будет проще - нужно передать ему архив с резервной копией, а не заниматься извлечением его файлов вручную.

Цикл for проходится по всем элементам вывода команды ls /home. В этом каталоге (/home) обычно находятся только каталоги (файлов в нем нет, только в его подкаталогах), то нам достаточно просто вывести его содержимое командой ls, а затем передать каждую строку вывода команде tar.

Опции czf означают, что нам нужно создать архив (c) в файле (параметр f) и при этом его сжать. Команда tar может создавать архивы и на ленточных устройствах, поэтому важно задать параметр f. Второй параметр - это имя архива, в нашем случае это будет /mnt/backup/<имя_пользователя>.tar.gz. Третий параметр - это имя каталога, который мы будем архивировать.

Каталог /mnt/backup должен существовать и к нему должен быть подмонтирован внешний жесткий диск. Об этом наш сценарий не заботиться, но при желании никто не помешает вам добавить строку вроде этой:

```
mount /dev/sdb1 /mnt/backup
```

Эта строку нужно поместить до команды rm, а в самой команде нужно изменить имя устройства (в нашем случае это /dev/sdb1).

Если вы решились управлять монтированием через сценарий, то в конце сценария нужно добавить строку размонтирования диска:

```
umount /mnt/backup
```

Также мы можем скопировать по SSH созданные архивы на удаленную машину. Тогда у вас будет две резервных копии, что повышает надежность нашего плана резервного копирования. После команды tar добавьте команду ($hd - сокращение от home directory):

```
scp $hd.tar.gz user@examle.com:/backups
```

Данная команда скопирует файл <имя_пользователя>.tar.gz на SSH-сервер example.com в каталог /backups. Вот только есть один момент: программа scp запрашивает пароль пользователя для копирования файлов на удаленный сервер. Чтобы этого не происходило (тогда вам нужно присутствовать перед компьютером и вводить пароль), нужно настроить аутентификацию с помощью ключей. Пока не будем на этом останавливаться, а рассмотрим измененную версию нашего сценария:

```
#!/bin/bash

mount /dev/sdc1 /mnt/backup
rm -r /mnt/backup/*
cd /home

for hd in `ls /home`; do
   echo "Creating backup for $hd"
   tar -czf /mnt/backup/$hd.tar.gz $hd
scp $hd.tar.gz user@examle.com:/backups
done

# Сбрасываем буферы на диск (не обязательно)
sync
# Размонтируем файловую систему
umount /mnt/backup
```

Если вам нужно копировать только содержимое каталога /home, то у вас есть уже рабочая версия сценария. Однако у нас еще есть каталог /var/lib/mysql. Однако код цикла для создания резервной копии будет отличен, поскольку в каталоге /var/lib/mysql кроме подкаталогов с файлами баз данных есть еще и обычные файлы, которые нам не нужно никуда копировать. Поэтому в цикле нужно проверить, является ли элемент, который вывела программа ls, каталогом и заархивировать только его:

```
echo "Database backup..."
cd /var/lib/mysql
```

```
for db in `ls /var/lib/mysql`; do
  test -d && "$db" && tar -czf /mnt/backup/db/$db.tar.gz
$db
done
```

Обратите внимание: архивы создаются в каталоге /mnt/backup/db: если имя базы данных у нас совпадает с именем пользователя, то ничего хорошего из затеи хранить архив в каталоге /mnt/backup у нас не выйдет. Поэтому мы должны создать каталог db заранее и хранить в нем резервные копии баз данных.

Кроме домашних каталогов и базы данных нужно еще сделать резервную копию почтовых ящиков пользователей, которые обычно хранятся в каталоге /var/mail. Здесь все просто - каждому пользователю соответствует файл почтового ящика, поэтому можем использовать код, подобный созданию резервной копии домашнего каталога, только резервные копии будем хранить в каталоге /mnt/backup/mail:

```
cd /var/mail

for dn in `ls /var/mail`; do
  echo "Creating backup for $dn [mail]"
  tar -czf /mnt/backup/mail/$dn.tar.gz $dn
# Если нужно, раскомментируйте строку
# scp $dn.tar.gz user@examle.com:/backups/mail
done
```

Теперь интегрируем наши два сценария в один общий (листинг 9.8)

Листинг 9.8. Сценарий /bin/backup

```
#!/bin/bash
# Измените имя устройства или закомментируйте вообще эту
строку
# если /mnt/backup монтируется по умолчанию
mount /dev/sdb1 /mnt/backup

# Рекурсивное удаление всего из /mnt/backup
rm -r /mnt/backup/*
# Создаем каталог для хранения базы данных
mkdir /mnt/backup/db
# Каталог для почтовых ящиков
mkdir /mnt/backup/mail

cd /home
```

```
# Копируем домашние каталоги
for hd in `ls /home`; do
  echo "Creating backup for $hd"
  tar -czf /mnt/backup/$hd.tar.gz $hd
done

# Копируем базы данных
echo "Database backup..."
cd /var/lib/mysql
for db in `ls /var/lib/mysql`; do
  test -d && "$db" && tar -czf /mnt/backup/db/$db.tar.gz $db
done

cd /var/mail
echo "Mail backup..."
for dn in `ls /var/mail`; do
  echo "Creating backup for $dn [mail]"
  tar -czf /mnt/backup/mail/$dn.tar.gz $dn
done

# Если нужно хранить резервные копии на удаленной машине
# scp -r /mnt/backup user@example.com:/backups

# Сбрасываем буферы на диск (не обязательно)
sync
# Размонтируем файловую систему
umount /mnt/backup
```

Наш сценарий готов полностью. Осталось только настроить SSH-сервер на аутентификацию без пароля. Для этого на клиенте (то есть на нашем сервере, резервное копирование которого вы будете производить, далее SSH-клиент) введите команду:

```
# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
...
```

Эта команда создает пару приватного/публичного ключей. Обратите внимание, что когда программа попросит вас ввести ключевую фразу, вводить ничего не нужно - вы же не хотите вводить пароль?

После этого на SSH-сервере (на компьютере, на который вы будете отправлять резервные копии) откройте файл конфигурации sshd. Обычно это /etc/ssh/sshd_config. Добавьте в него строки:

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile %h/.ssh/authorized_keys
```

Обычно они в нем уже есть, но их нужно расскомментировать. Строка «%h/.ssh/authorized_keys» означает, что ключи будут храниться в домашнем каталоге пользователя (%h), подкаталог .ssh, файл authorized_keys. Если вы этого файла там нет, вы можете его создать самостоятельно.

Скопируем наш файл публичный ключ на SSH-сервер (команду вводим на SSH-клиенте):

```
scp /root/.ssh/id_rsa.pub root@example.com:/root/
```

На SSH-сервере вводим следующую команду, помещающую ваш ключ в файл authorized_keys:

```
cat /root/id_rsa.pub >> /root/.ssh/authorized_keys
```

Перезапустите SSH-сервер:

```
sudo service ssh restart
```

Теперь переходим на SSH-клиент и попробуем скопировать файл на сервер. На этот раз команда ssh не должна запрашивать пароль:

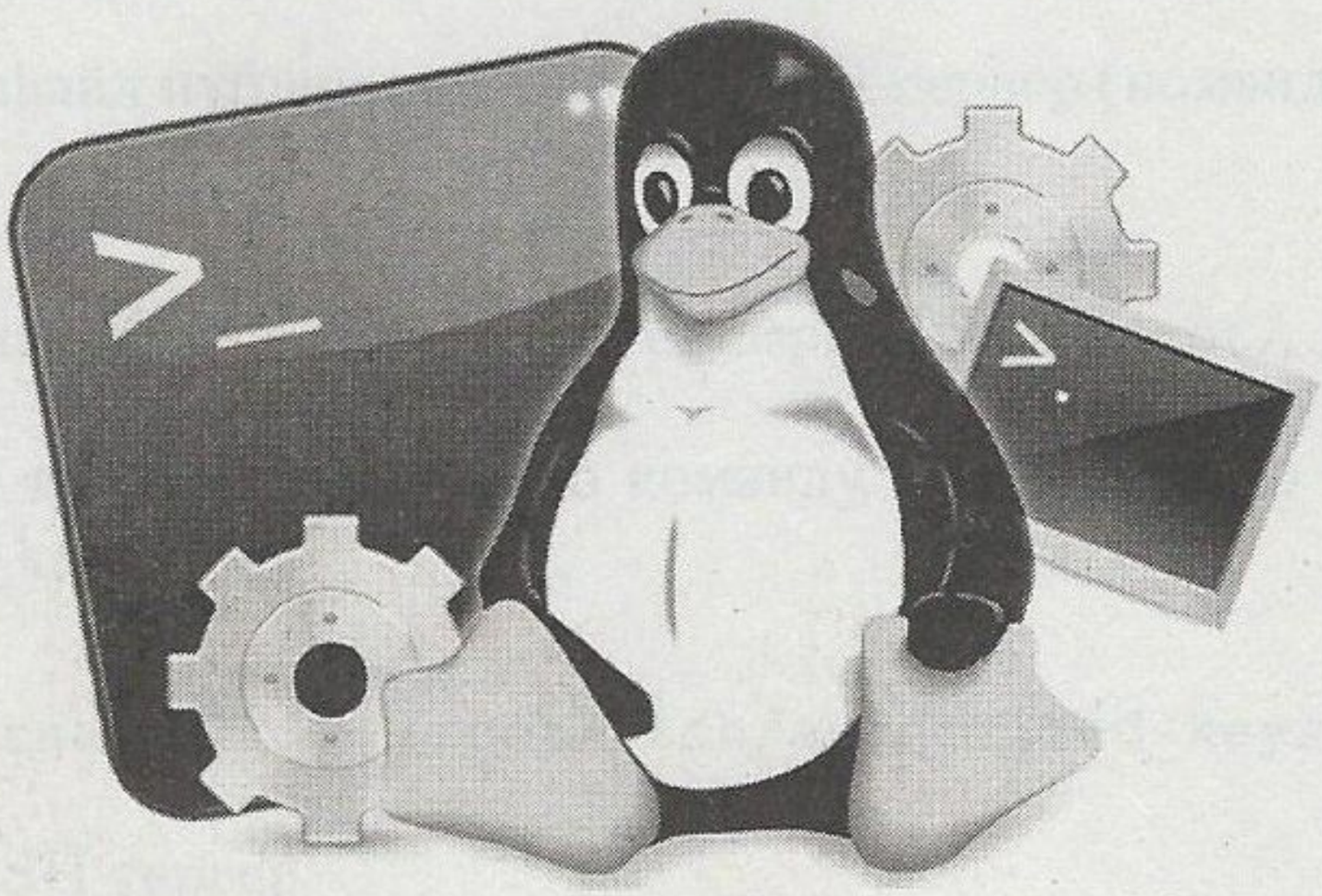
```
scp /root/.ssh/id_rsa.pub root@example.com:/root/
```

Все, наш SSH-сервер настроен. Не забудьте на нем создать каталог /backups и можно приступать к тестированию сценария резервного копирования.

Когда вы убедитесь, что все работает, как нужно, добавьте вызов сценария /bin/backup на вашем веб-сервере в таблицу расписания планировщика.

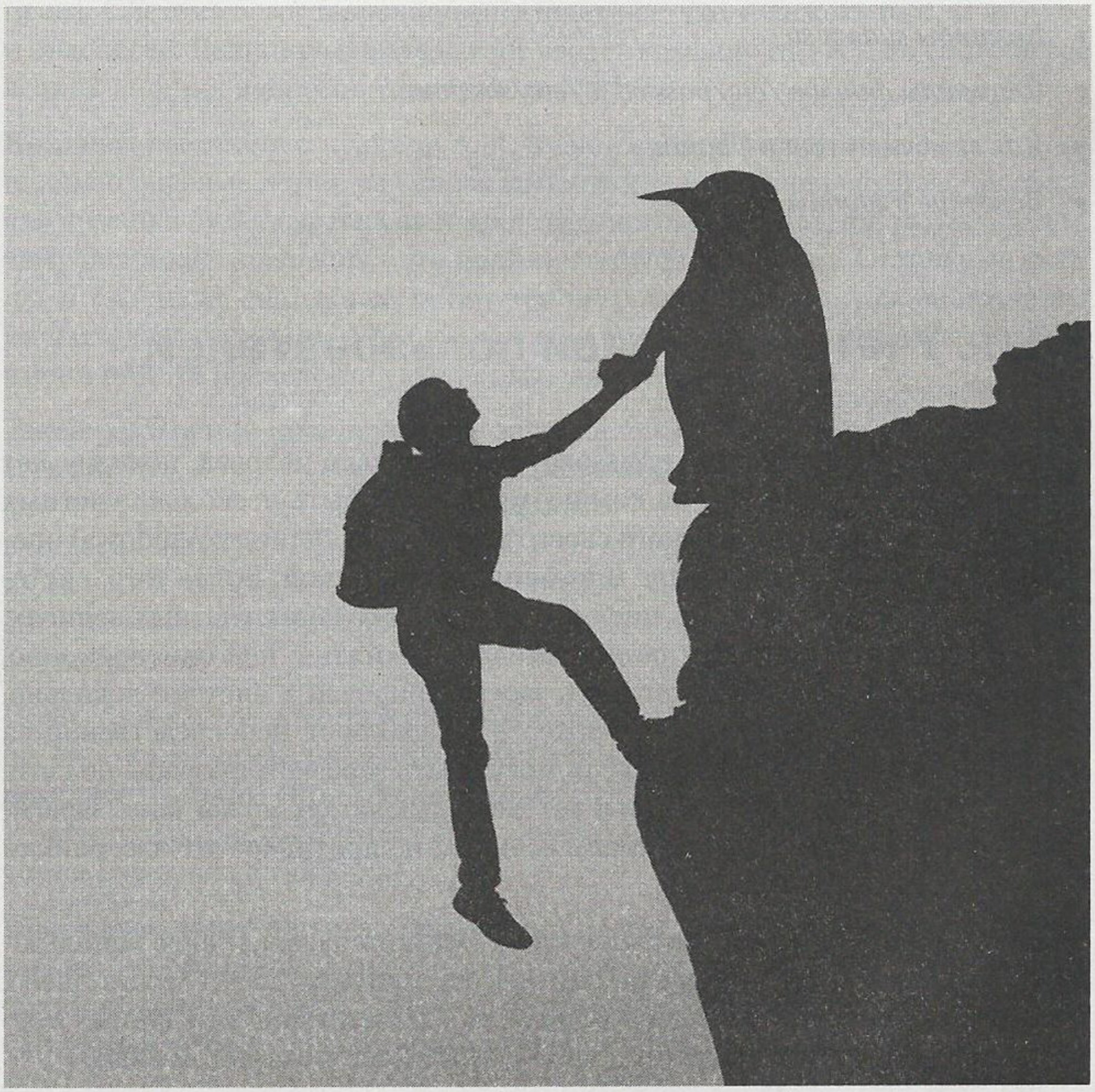
За недолгое время мы создали систему резервного копирования с ротацией резервных копий и возможностью хранения резервных копий на удаленной машине. И все это - стандартными средствами.

Продолжить писать сценарии можно было бы до бесконечности, но эта книга посвящена Linux в целом, а не только bash. Если вы заинтересовались, рекомендую прочитать следующую статью **https://habrahabr.ru/post/126701/**, где вы найдете дополнительные полезные примеры.



ГЛАВА 10.

Пользователи и группы. Основы безопасности



В этой главе:

- *Обычные учетные записи и root*
- *Команды sudo и su*
- *Форматы файлов /etc/passwd и /etc/shadow*
- *Как включить root в Ubuntu*
- *Введение в группы*

10.1. Учетные записи пользователей

Одним компьютером могут пользоваться несколько человек, поэтому нет ничего удивительного, что на компьютере может быть несколько учетных записей пользователей. С самого своего начала (в отличие от Windows) операционная система Linux была многопользовательской. Более того, система может не просто хранить информацию о нескольких учетных записях, но и позволяет нескольким пользователям работать с ней одновременно. Так, один пользователь может быть зарегистрирован в системе локально, скажем, через графический менеджер - он работает непосредственно за компьютером. Другие пользователи могут быть зарегистрированы по сети, например, по ssh или FTP. Один и тот же пользователь может быть зарегистрирован в системе несколько раз, если это не противоречит настройкам безопасности.

Удаленный доступ к системе был в Linux с самого начала. Ранее использовался протокол telnet, сейчас он заменен на протокол SSH (Secure Shell), поскольку он обеспечивает шифрование передаваемых по сети данных в отличие от telnet, где все, включая пароль, передается в открытом виде.

Среди учетных записей пользователей можно выделить одну особенную - это пользователь root, обладающий максимальными правами в системе. О нем мы и поговорим в следующем разделе.

10.2. Пользователь root

10.2.1. Максимальные права - это хорошо или плохо?

Как уже было только что отмечено, у пользователя root есть максимальные права. Система безоговорочно выполняет все, что «скажет» root. В этом то и опасность. Если пользователь root введет команду rm -rf /, то система без всяких лишних вопросов удалит сама себя.

Когда вы работаете с правами root, будьте крайне осторожны. Есть потенциально опасные команды - та же команда rm, удаляющая файлы. Если такая команда будет введена от имени обычного пользователя, то, максимум, чему она может навредит - это пользовательским файлам. Остальные файлы, в том числе системные, останутся нетронутыми - система не позволит ничего сделать с ними. А вот когда такая опасная команда будет введена от имени root, то последствия могут быть непоправимы.

Такое разделение прав доступа - залог безопасности Linux. Да, для Linux есть вирусы, но их не так много, как для Windows. Даже если вы запустите вирус в Linux, то он максимум, что сможет сделать - это повредить пользовательские файлы. Получить права root он вряд ли сможет, если вы сами их ему не предоставите. А поэтому прежде, чем запустить загруженные из Интернета сценарии и программы, дважды подумайте, стоит ли это делать от имени root.

Права root настолько огромны, что появились системы ограничения доступа, способные ограничить права даже самого root, если заподозрят что-то неладное. К таким системам относятся SELinux, LIDS и некоторые другие. В некоторых дистрибутивах они активны по умолчанию, в некоторых - установлены, но не активны.

Некоторые дистрибутивы, например, Ubuntu, отключают учетную запись root из соображений безопасности. Создаваемый при установке системы пользователь добавляется в файл /etc/sudoers, что позволяет ему использовать команду sudo для кратковременного получения прав root.

Если вы хотите, чтобы с вашей системой все было хорошо, придерживайтесь следующих рекомендаций:

- Старайтесь не работать под именем root.
- Если вам нужны права root, например, для установки программного обеспечения, используйте команды sudo и su. Подробно эти команды будут рассмотрены далее.
- Создайте обычного пользователя, под которым вы будете выполнять обычные ежедневные операции - серфинг в Интернете, работа с документами.
- Если вы устанавливается программное обеспечение не из репозитария, а из стороннего источника, то требование прав root должно, как минимум, насторожить. По возможности, откажитесь от такого приложения или проведите дополнительную проверку.

10.2.2. Команды sudo и su

Для выполнения некоторых операций, например, для изменения конфигурационных файлов, управления пакетами, нужны права root. Чтобы их получить, необязательно входить, как root. Тем более, что в некоторых дистрибутивах вход, как root запрещен.

Нужно использовать команды sudo или su. Давайте разберемся, какая между ними разница. Начнем с команды su:

```
su
```

После ввода команды система запросит пароль пользователя root. Введите его и вы получите права root. Пароль root, как правило, устанавливается при инсталляции Linux. Команду удобно использовать истинному администратору - тому, кто является root, но из соображений безопасности использует другую учетную запись. Поскольку только он знает пароль root. Вы же не будете предоставлять свой пароль кому угодно, правильно?

После ввода команды su (и правильного ввода пароля) открывается полноценная root-сессия на неограниченное время. Поэтому не забудьте или ввести команду exit или закрыть окно терминала, когда права root станут вам не нужны.

Теперь поговорим о команде sudo. Она предназначена для выполнения одной команды с правами root:

```
sudo <команда>
```

После ввода этой команды система запросит пароль пользователя, который вводит эту команду. Но это не означает, что права root будут предоставлены любому пользователю в системе. Разрешение использовать эту команду имеют только пользователи, указанные в файле /etc/sudoers. Для редактирования этого файла нужно использовать команду visudo:

```
su
visudo
```

Также желательно установить переменную окружения EDITOR, чтобы система запустила какой-то удобный редактор, иначе будет запущен редактор vi, который крайне неудобный, о чем было сказано в главе 8.

Использовать команду sudo предпочтительнее по следующим причинам:

- Команду sudo нужно вводить при вводе каждой команды, которая требует права root. Вы уже точно не забудете, что работаете с root правами.
- При вводе первой команды sudo запрашивается пароль. При вводе второй, если прошло немного времени, пароль не запрашивается, но если пройдет чуть больше времени (зависит от настроек системы), то sudo опять будет запрашивать пароль. Сессию с su легко оставить открытой - ввел команду su, поработал, как root, забыл ввести exit и отошел на обеденный перерыв. С sudo такое не пройдет.
- Каждая команда, введенная с помощью sudo, записывается в журнал /var/log/auth.log, поэтому в случае чего вы хотя бы будете знать, что случилось, прочитав этот журнал. А также у вас будет храниться история введенных команд с полномочиями root, а если вы будете работать под именем root, никакой журнал вестись не будет.
- Администратор может разрешать и запрещать другим пользователям использовать полномочия root, не предоставляя пароль root, то есть практически нет риска скомпрометировать учетную запись root.
- Если кто-то захочет взломать вашу систему, то у него ничего не выйдет поскольку учетная запись root отключена, а названия других учетных записей еще раздобыть нужно.

Да, sudo неудобно использовать при вводе множества команд. Но есть одно ухищрение. Можно ввести команду sudo bash, запустится командная оболочка с правами root и все запущенные в ней программы, будут обладать правами root. По сути, такая сессия будет аналогична той, которая вызывается командой su, поэтому нужно быть предельно осторожными.

Однако команда sudo bash будет полезна, когда у вас есть необходимость вводить сложные команды после sudo. Так, sudo не очень дружит с перенаправлением ввода вывода. Поэтому проще ввести sudo bash и просто работать, не задумываясь об использовании sudo.

10.2.3. Как включить root в Ubuntu

Включать учетную запись root крайне не рекомендуется, но если вам нужно это сделать, то делается это достаточно просто. Для этого достаточно задать пароль для пользователя root:

```
sudo passwd root
```

Сначала программа запросит ваш пользовательский пароль, затем новый пароль root и его подтверждение:

```
Enter your existing password
Enter password for root
Confirm password for root
```

После этого вы сможете войти в систему под учетной записью root.

Для отключения учетной записи root используется команда:

```
sudo passwd -l root
```

> **Внимание!** После закрытия учетной записи root у вас могут быть проблемы с входом в систему в режиме восстановления, поскольку пароль root уже установлен (то есть он не пустой, как по умолчанию), но в то же время учетная запись закрыта. Поэтому лучше использовать sudo bash и не включать учетную запись root вовсе.

10.3. Файл /etc/passwd

Информация об имеющихся учетных записях хранится в файле /etc/passwd. Фрагмент этого файла с реальной машины приведен в листинге 10.1.

Листинг 10.1. Фрагмент файла /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

```
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
…
ivan:x:1000:1000:ivan,,,:/home/ivan:/bin/bash
proftpd:x:123:65534::/run/proftpd:/bin/false
ftp:x:124:65534::/srv/ftp:/bin/false
```

Как видите, кроме учетных записей обычных пользователей в этот файл добавляются и служебные учетные записи. Как правило, служебные учетные записи нужны, чтобы от их имени запускался тот или иной сервис. Раньше сервисы запускались от имени root, что создавало огромную брешь в системе безопасности. Сейчас же можно назначить отдельные права для каждого пользователя, чем ограничить права того или иного сервиса.

Рассмотрим формат файла /etc/passwd:

```
имя_пользователя:пароль:UID:GID:полное_имя:домашний_
каталог:оболочка
```

- Имя пользователя - это логин пользователя, который он вводит для регистрации в системе.
- Пароль в современных системах в этом файле не указывается, а второе поле осталось просто для совместимости со старыми системами. Сейчас пароли хранятся в файле /etc/shadow.
- UID (User ID) и GID (Group ID) — идентификаторы пользователя и группы соответственно. Идентификатор пользователя root всегда равен 0, как и идентификатор группы root. Список групп вы найдете в файле /etc/groups. Обратите внимание, что UID обычных пользователей по умолчанию всегда > 1000, а UID специальных пользователей (служебных пользователей) - до 1000.
- Полное имя - имя пользователя в реальной жизни, может не заполняться, а может содержать реальные данные, по которым администратор сможет понять, кто есть кто. Все зависит от использования компьютера. Если это домашний компьютер с двумя-тремя учетными записями, то и так понятно, кто есть кто. А вот если учетных записей, скажем, 500, то запомнить все довольно сложно.
- Домашний каталог - имя домашнего каталога - каталога, который содержит файлы пользователя. Обычно домашний каталог называется /home/<логин>

- Оболочка - оболочка, которая будет запущена при входе пользователя в консоль или при запуске терминала. У служебных аккаунтов, как правило, используются программы вроде nologin, false, true - это программы-заглушки, которые завершают работу сразу после запуска. Использование этих программы отключает вход учетной записи в систему.

Раньше пароли, хоть и в зашифрованном виде, хранились в /etc/passwd. Затем они были вынесены в файл /etc/shadow, доступ к которому есть только у пользователя root. Сделано такое разделение для того, чтобы обычные пользователи (или различные программы, которым нужно обращаться к списку учетных записей) не видели пароль, пусть даже и зашифрованный.

В файле /etc/shadow полей больше, чем в /etc/passwd. Как и в случае с /etc/passwd, поля разделяются двоеточиями:

1. Имя пользователя. Совпадает с именем пользователя в файле /etc/passwd

2. Зашифрованный пароль. Позже мы поговорим о том, как распознать алгоритм шифрования, которым был зашифрован пароль.

3. Количество дней (с 1 января 1970 года), когда пароль был сменен в последний раз.

4. Число дней до смены пароля. Если в этом поле 0, то пароль может быть сменен в любой момент.

5. Количество дней, после которых пароль должен быть сменен. Обычно здесь значение 999999, которое показывает, что пользователь может никогда не менять свой пароль.

6. Число дней, в течение которых пользователь получает предупреждение о необходимости изменить пароль. Обычно такие предупреждения пользователь получает за неделю (7 дней) до часа «X».

7. Число дней после окончания действия пароля, когда еще пользователь может работать со старым паролем. Если после этого срока пользователь не сменит пароль, учетная запись будет заблокирована.

8. Число дней, начиная с 1 января 1970, после которых пароль будет заблокирован

9. Не используется.

Обычно последние три поля не используются. Поле с зашифрованным паролем позволяет понять, какой алгоритм система использовала для его шифрования. Посмотрите на начало зашифрованного пароля:

- 1 – MD5. Ранее часто использовался, сейчас чаще используется SHA-512, поскольку в MD5 обнаружились математические уязвимости.
- 2, $2a$ – Blowfish. Чаще используется в FreeBSD/OpenBSD, чем в Linux;
- 5 – SHA-256;
- 6 – SHA-512. Используется в современных дистрибутивах.

Форматы файлов /etc/passwd и /etc/shadow были приведены, чтобы вы понимали, что происходит. Модифицировать учетную запись пользователя правильнее с помощью команды usermod, а не с помощью редактирования файла /etc/passwd. Конечно, можно внести небольшие изменения, например, изменить полное имя пользователя. А вот для изменения остальных параметров, например, домашнего каталога, правильнее использовать usermod, чтобы потом не делать много ручной работы. Команды управления учетными записями будут рассмотрены в следующей главе.

10.4. Введение в группы

В некоторых случаях пользователей объединяют в группы для более эффективного управления ними. Представим, что у нас есть три пользователя: ivan, mark, eva, которые должны совместно работать над проектом. Чтобы пользователи получили доступ к домашним каталогам друг друга их достаточно объединить в одну группу. По умолчанию пользователь не имеет доступ к домашнему каталогу другого пользователя, поскольку пользователи находятся в разных группах. Также можно предоставить доступ к какому-то объекту всей группе сразу, тогда каждый ее член сможет обратиться к объекту и не нужно будет указывать права отдельно для каждого пользователя.

Формат этого файла следующий:

```
имя_группы:пароль:GID:список_пользователей
```

- Имя_группы - имя группы, вы увидите его в выводе команды ls -l.
- Пароль - обычно пароль для группы не используется, поэтому это поле будет пустым.
- GID - данный номер указывается в /etc/passwd для задания принадлежности пользователя к группе.
- Список пользователей - содержит список членов группы. Имена пользователей разделяются запятыми

Фрагмент файла /etc/group приведен в листинге 10.2.

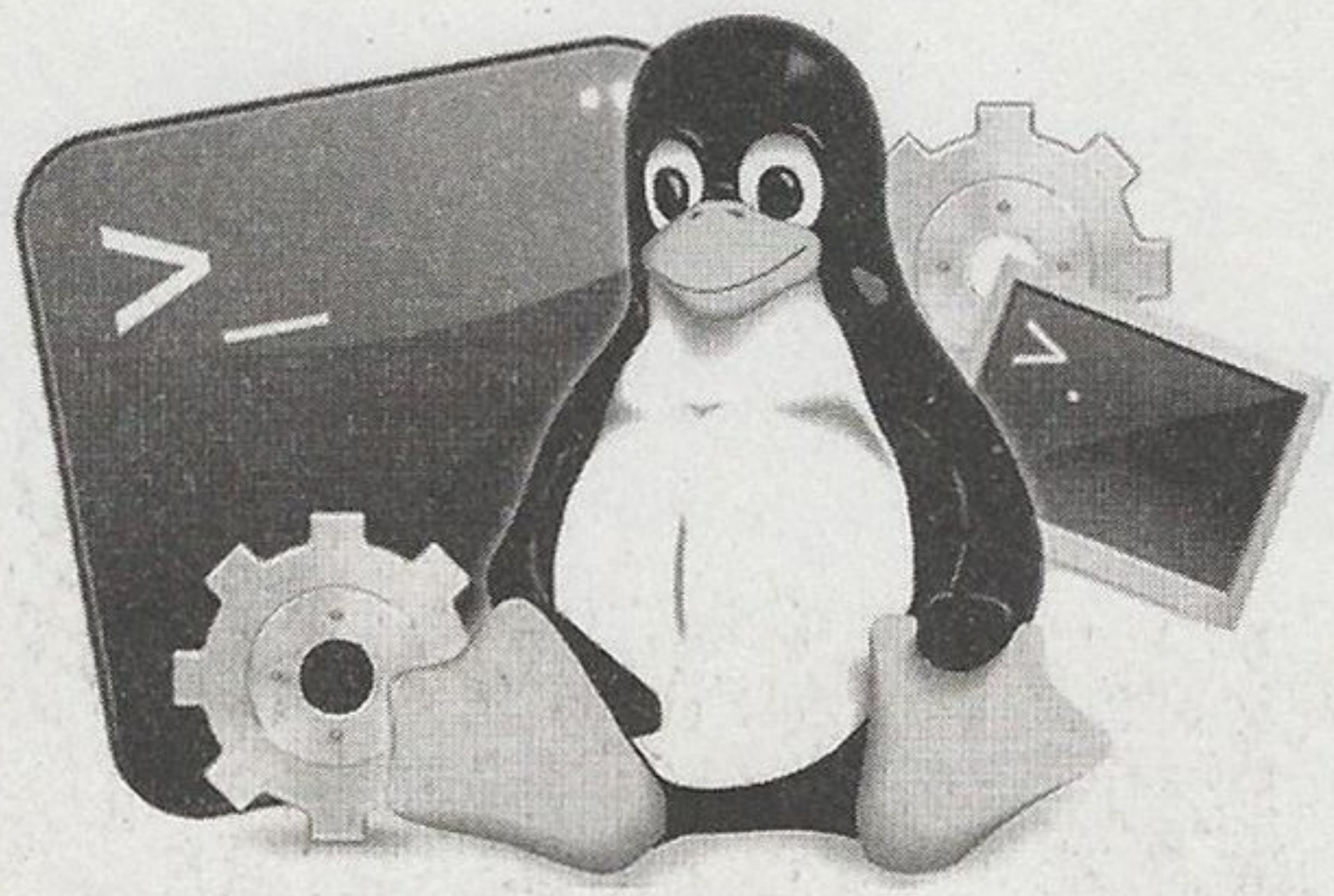
Листинг 10.2. Фрагмент файла /etc/group

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ivan
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:ivan
…
colord:x:129:
saned:x:130:
ivan:x:1000:
sambashare:x:131:ivan
```

Создать группу, а также поместить пользователя в группу позволяют графические конфигураторы. Вы можете использовать их — они очень удобные, но если вы хотите стать настоящим профи, тогда вы должны знать, что доступные в системе группы указываются в файле /etc/group. Добавить новую группу в систему можно с помощью команды groupadd, но, как пра-

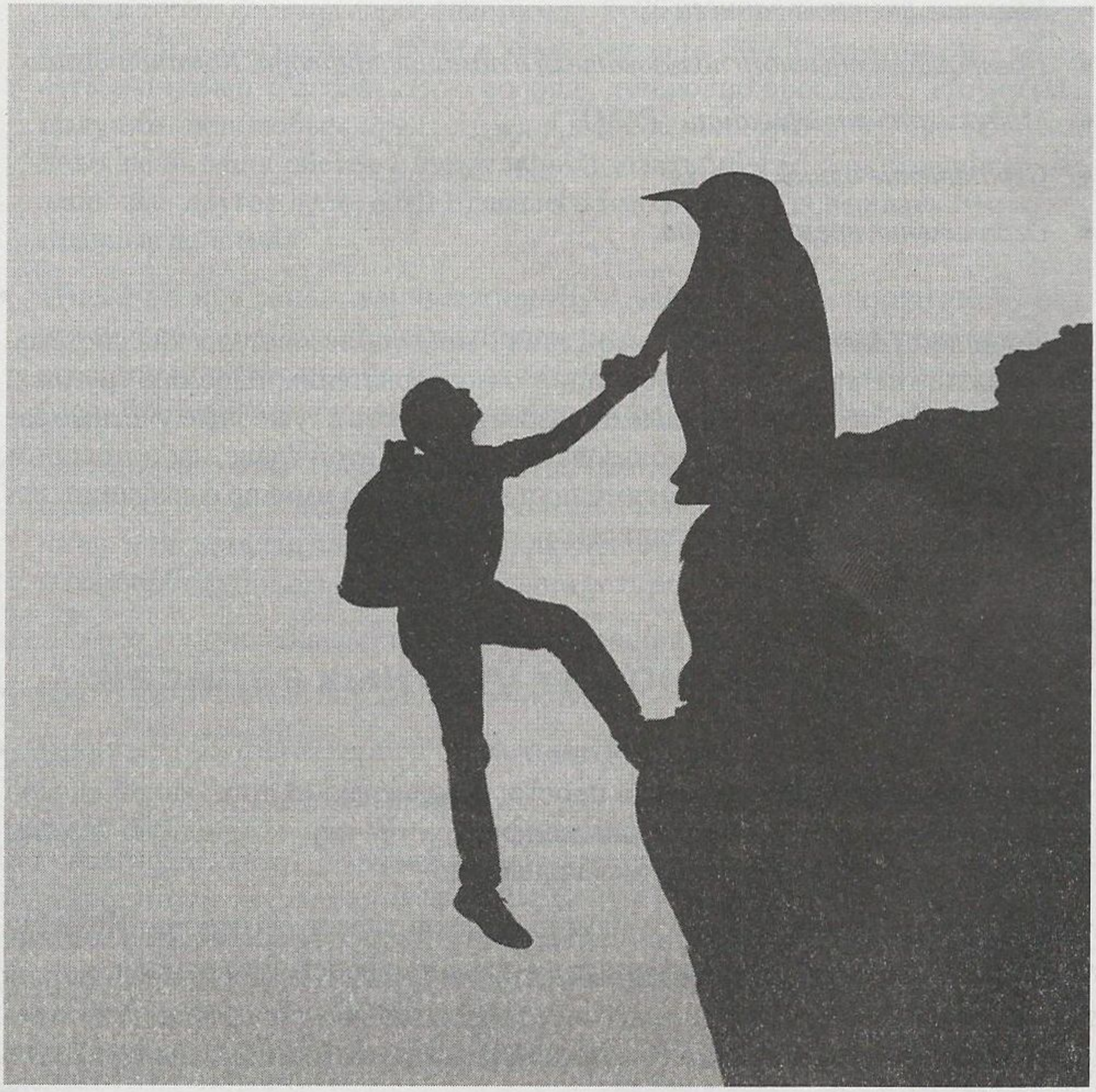
вило, проще добавить в текстовом редакторе еще одну запись в файл /etc/group, а изменить группу пользователя еще проще — для этого достаточно отредактировать /etc/passwd.

В следующей главе будет рассмотрено управление учетными записями пользователей, а управление правами доступа будет рассмотрено в главе 12.



ГЛАВА 11.

Управление учетными записями пользователей



В этой главе:

- *Создание новых учетных записей в консоли и с помощью GUI*
- *Редактирование учетных записей пользователей*
- *Удаление учетных записей*
- *Превращаем обычного пользователя в администратора. Команда visudo*
- *Модули аутентификации (PAM)*
- *Ограничение длины пароля*
- *Ограничение времени входа*

Из прошлой главы вы узнали, что Linux - многопользовательская система. Вы узнали, где хранится информация о пользователях, паролях, группах. Было сказано, что есть обычные пользователи, есть служебные учетные записи и есть учетная запись пользователя root. В этой главе мы поговорим об управлении учетными записями пользователей, а именно о создании, изменении, удалении учетных записей.

11.1. Создание новых учетных записей

Создать учетную запись - очень просто, независимо от того, какой способ вы предпочитаете - графический конфигуратор или командную строку. Важно создать эту учетную запись правильно.

При создании учетной записи администратор задает ее пароль. Поэтому первое, о чем нам нужно поговорить - о создании надежного пароля.

11.1.1. Выбор надежного пароля

Первым делом нужно отметить, что Linux всегда подскажет вам, если ваш пароль ненадежен, а некоторые дистрибутивы даже не позволят установить такой пароль. Все зависит от настроек модулей аутентификации (PAM), которые будут также рассмотрены в этой главе. Поэтому не нужно волноваться - Linux позволит установить простой пароль, только если вы сами этого захотите.

Давайте разберемся, что такое простой, а что такое сложный пароль. Не рекомендуется в паролях использовать следующую информацию:

- Номер телефона, ваш адрес, номер социального страхования - такая информация, как правило, общедоступна и легко подбирается. Неоднократно видел, как сотрудники провайдера устанавливают в качестве имени точки доступа адрес (например, kv26), а в качестве пароля - номер телефона пользователя. Если я знаю соседа из 26-ой квартиры и у меня есть его номер телефона (что вообще, думаю, не проблема - его можно получить под любым предлогом), я могу использовать его Wi-Fi. Одно дело, если вашу систему будут использовать просто для серфинга - на здоровье, другое дело, если от вашего имени будут совершать противоправные действия.
- Имена членов семьи, клички животных, имена друзей, номер авто - вся эта информация также легко получается, поэтому избегайте ее использования. Если уж сильно хочется использовать кличку животного, то добавляйте в пароль случайную информацию, например, Bar_31_siK.
- Словарные слова - такие слова легко подбираются с помощью атаки по словарю, причем независимо, какие слова вы используете - на русском, на английском, английские при включенной русской раскладке (ыускуе) и наоборот (gfhjkm). У хакеров давно есть все словари.
- Любую другую общедоступную информацию о вас.
- Простые числовые пароли вроде 123456.
- Простые текстовые пароли вроде qwerty, password.

Как действует хакер? Сначала он пытается угадать ваш пароль, используя общедоступную информацию о вас и словарь популярных паролей. В большинстве случаев ему это удается на первом этапе. Если у него не получается, он подключает общие словари и пытается атаковать систему. Он знает, что после нескольких неудачных попыток входа в систему за короткое время учетная запись может быть заблокирована, поэтому он будет делать паузы,

а все это увеличивает время подбора пароля. Поэтому хакеры и стараются сначала использовать общедоступную информацию в надежде на простую победу. Последний этап - это атака грубой силы: программа генерирует пароль путем перестановки букв (aaa, aab, aac и т.д.) и пробует все возможные комбинации. Представьте, что у нас есть только 2 символа и максимальная длина пароля - 3. Тогда количество вариантов будет 8 (000, 001, 010 и т.д.). Если же максимальная длина пароля - 8, то количество перестановок уже будет равно 256.

Наша задача усложнить процесс перебора пароля. Для этого нам нужно увеличить длину пароля, увеличить количество допустимых символов. Именно поэтому рекомендуется использовать кроме алфавитных еще и неалфавитные символы и цифры.

Итак, критерии надежного пароля:

- Минимальная длина - 8 символов
- Содержит символы разных регистров
- Содержит неалфавитные символы (_ # ! , . и т.д.)
- Содержит цифры
- Прост для запоминания. Можно, конечно, придумать пароль что-то вроде Du4nn?ED!!!!!!_!!!!!!#_1xsw. Его не подберут, но попробуйте его запомнить! Вы его забудете прежде, чем его успеют подобрать.

Интересно, что приведенный ранее пароль Bar_31_siK является надежным, хотя и основан на кличке животного: в нем есть символы разного регистра, неалфавитные и цифровые символы, он не содержит непрерывную кличку животного, содержит 10 символов. А самое главное - он прост для запоминания. Все, что нужно запомнить - первый и последний символ - в верхнем регистре, после первого слога идет знак подчеркивания, число 31, еще один знак подчеркивания и второй слог клички животного.

И еще несколько правил парольной безопасности:

- Если вы - администратор и создали пароль для пользователя, попросите его сразу же сменить пароль - после первого входа в систему. Администратор не должен знать пароль пользователя.
- Если вы - пользователь и получили пароль от администратора, смените его, как только войдете в систему.
- Периодически меняйте пароли, например, раз в 3 месяца.

- Не используйте один и тот же пароль для входа в систему и к другим сервисам (например, к общественному почтовому ящику, к социальной сети и т.д.)
- Не передавайте пароли по открытым каналам связи. Не передавайте пароли через Skype (он останется в истории сообщений на серверах Microsoft, даже если вы очистите эту историю на локальном компьютере). Лучший способ передачи пароли - лично в руки. Но в большинстве случаев это невозможно, поэтому передавайте пароли по e-mail, но с обязательным шифрованием. Второй способ - использование мобильной версии Telegram, которая поддерживает безопасные чаты с шифрованием. Можно передать пароль и через Skype, но в виде файла. Создайте файл с паролем, поместите его в архив с паролем и передайте пользователю. А пароль к архиву можно передать через другой сервис, например, Viber.

11.1.2. Управление группами

При создании пользователя создается группа с таким же именем. Если вы создаете пользователя ivan, то будет создана группа с таким же именем. Если вы работаете за личным компьютером, то все равно, что будет с группами. А вот если компьютер используется несколькими пользователя и нужно обеспечить их совместную работу, нужно задуматься об управлении группами.

Есть две стратегии:

- На основе пользовательских групп.
- На основе групп проектов.

Вернемся к нашей группе ivan. Представим, есть еще пользователи den и eva, которые работают вместе с ivan и им нужно обмениваться общими документами. Тогда можно этих пользователей добавить в группу ivan и они смогут получить доступ к домашнему каталогу Ивана. Это и есть стратегия на основе пользовательских групп. Подойдет для небольших компаний, состоящих из 3-5 человек.

Вторая стратегия заключается в создании групп для разных отделов, например, dev - для отдела разработчиков, admis - для администраторов, sales - для отдела продаж и т.д. Затем в эти группы добавляются пользователи и настраиваются права для каждой группы.

Эти два подхода не являются взаимоисключающими, но лучше использовать какой-то один, чтобы вы не запутались.

11.1.3. Создание учетной записи с помощью GUI-инструмента

Когда пользователь начинает использовать командную строку Linux, то он, как правило, забывает об использовании всяких графических конфигураторов. Как по мне, проще в терминале ввести команду adduser, чем искать в дебрях меню графических конфигуратор, заполнять его формы и т.д. Но некоторым пользователям нравятся только GUI-инструменты - видимо они так ненавидят DOS, что везде, где только можно используют графические программы.

В Fedora для добавления учетной записи пользователя откройте окно **Параметры** (рис. 11.1), в нем выберите **Пользователи** (рис. 11.2). Появится окно с информацией о вашей учетной записи. В верхнем правом углу будет кнопка **Разблокировать**. Нажмите ее и введите свой пароль. После этого данная кнопка исчезнет, а на ее месте появится новая - **Добавить пользователя**, нажмите ее.

Рис. 11.1. Параметры системы

Примечание. Создавать учетные записи имеет право или пользователь root или пользователь-администратор, то есть пользователь, внесенный в файл /etc/sudoers. Если ваша учетная запись не обладает такими правами, создать новую учетную запись не получится. Все, что вы можете - это изменить параметры своей учетной записи.

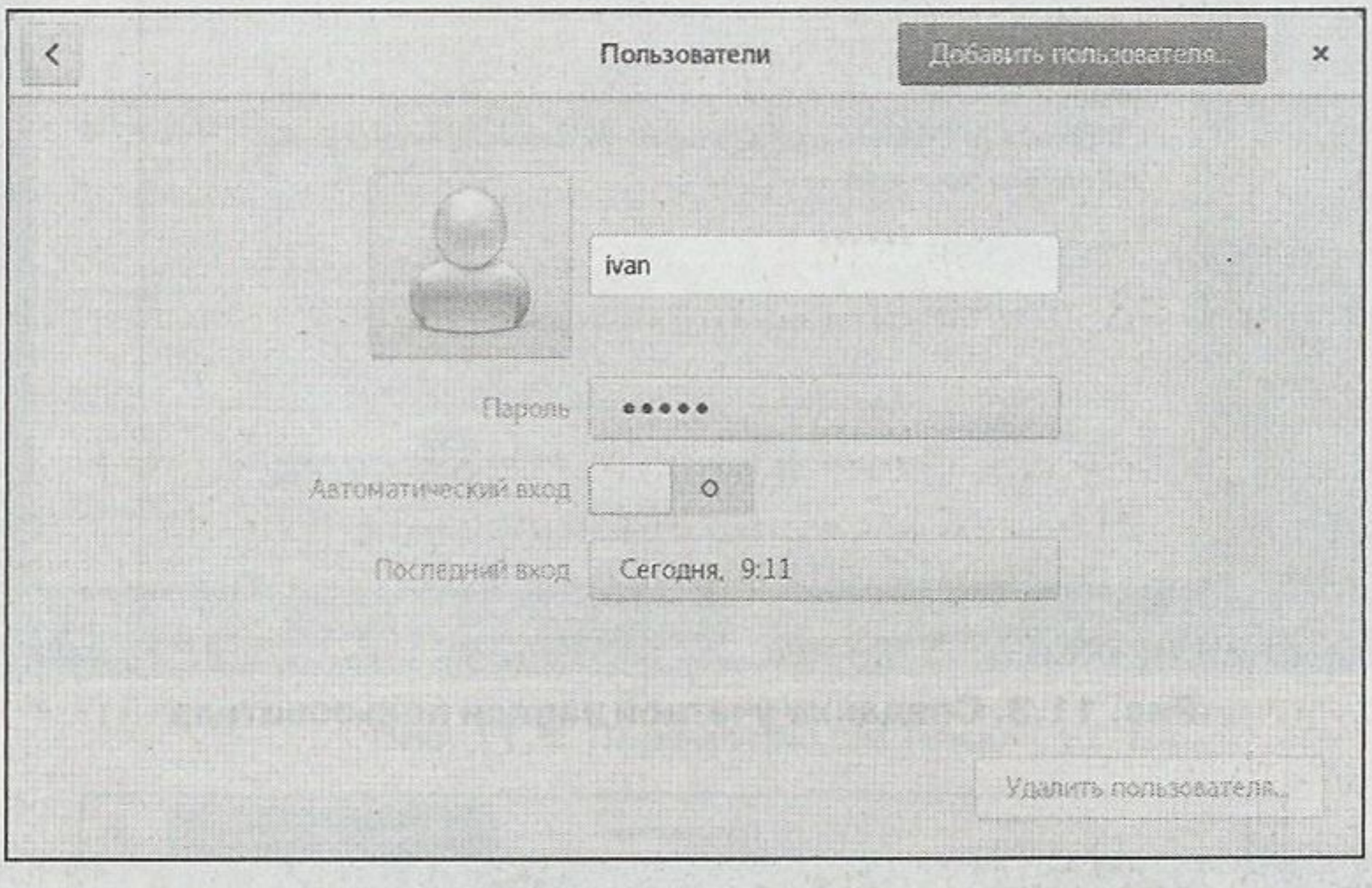


Рис. 11.2. Пользователи

Введите имя пользователя, выберите тип учетной записи - обычный или администратор. На рис. 11.3 видно, что я пытаюсь создать обычную пользовательскую запись для пользователя alina. Пароль можно не задавать сразу - можно разрешить пользователю установить пароль при следующем входе в систему. А можно установить пароль прямо сейчас.

Обратите внимание: конфигуратор не позволит добавить пользователя, пока пароль не будет соответствовать правилам безопасности системы - кнопка **Добавить** будет неактивной, пока вы не введете надежный пароль. Хорошо, хоть конфигуратор подсказывает, что именно ему не нравится, например, что нужно добавить еще буквы, цифры и символы - введенный пароль слишком короткий.

В Ubuntu процесс добавления пользователя практически такой же. Нужно открыть окно **Параметры системы**. Само окно и меню, позволяющее его открыть, показано на рис. 11.5. Выберите **Учетные записи** и нажмите кнопку **Разблокировать** в верхнем правом углу окна. Введите свой пароль и под списком учетных записей появится кнопка + (рис. 11.6). Нажмите ее.

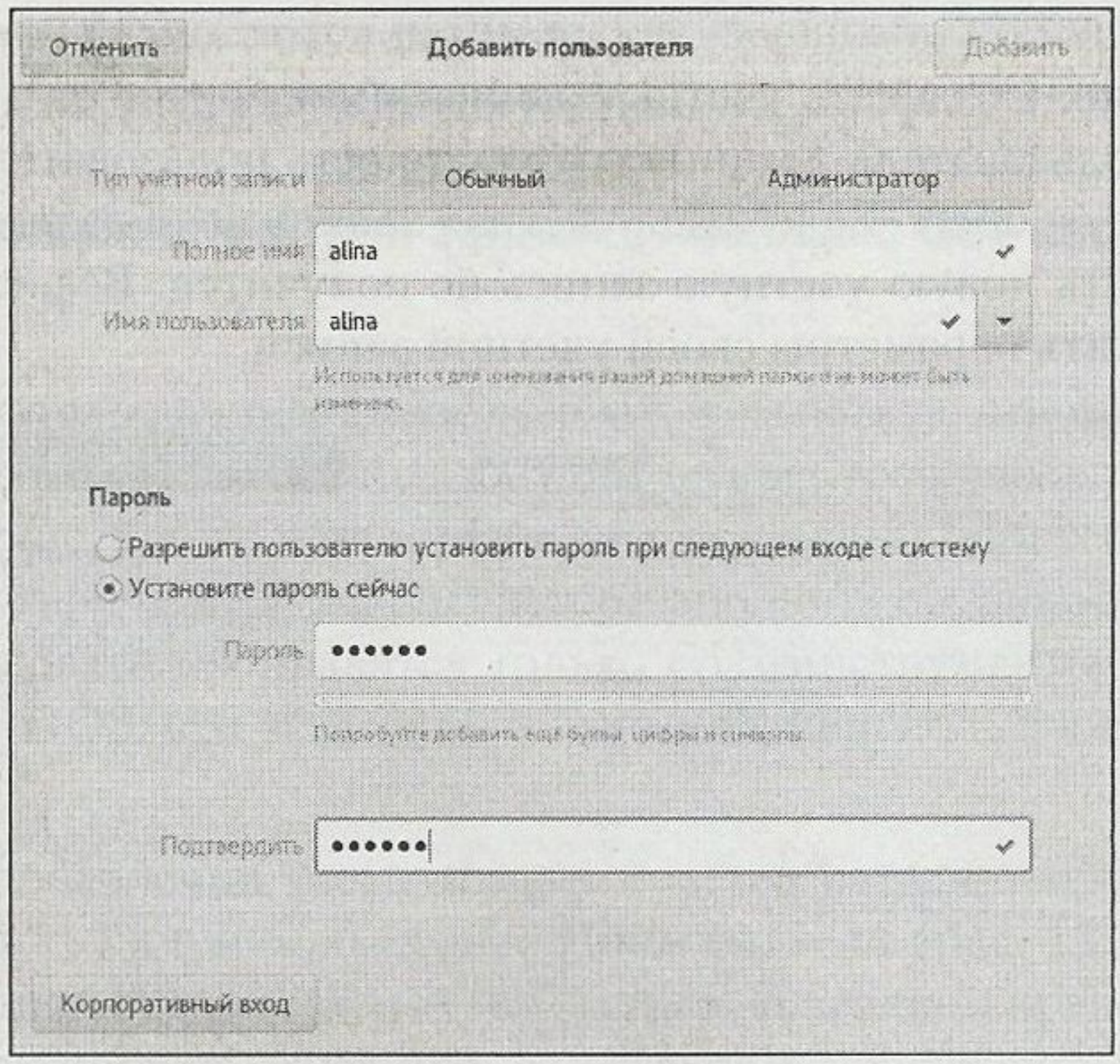


Рис. 11.3. Создание учетной записи пользователя

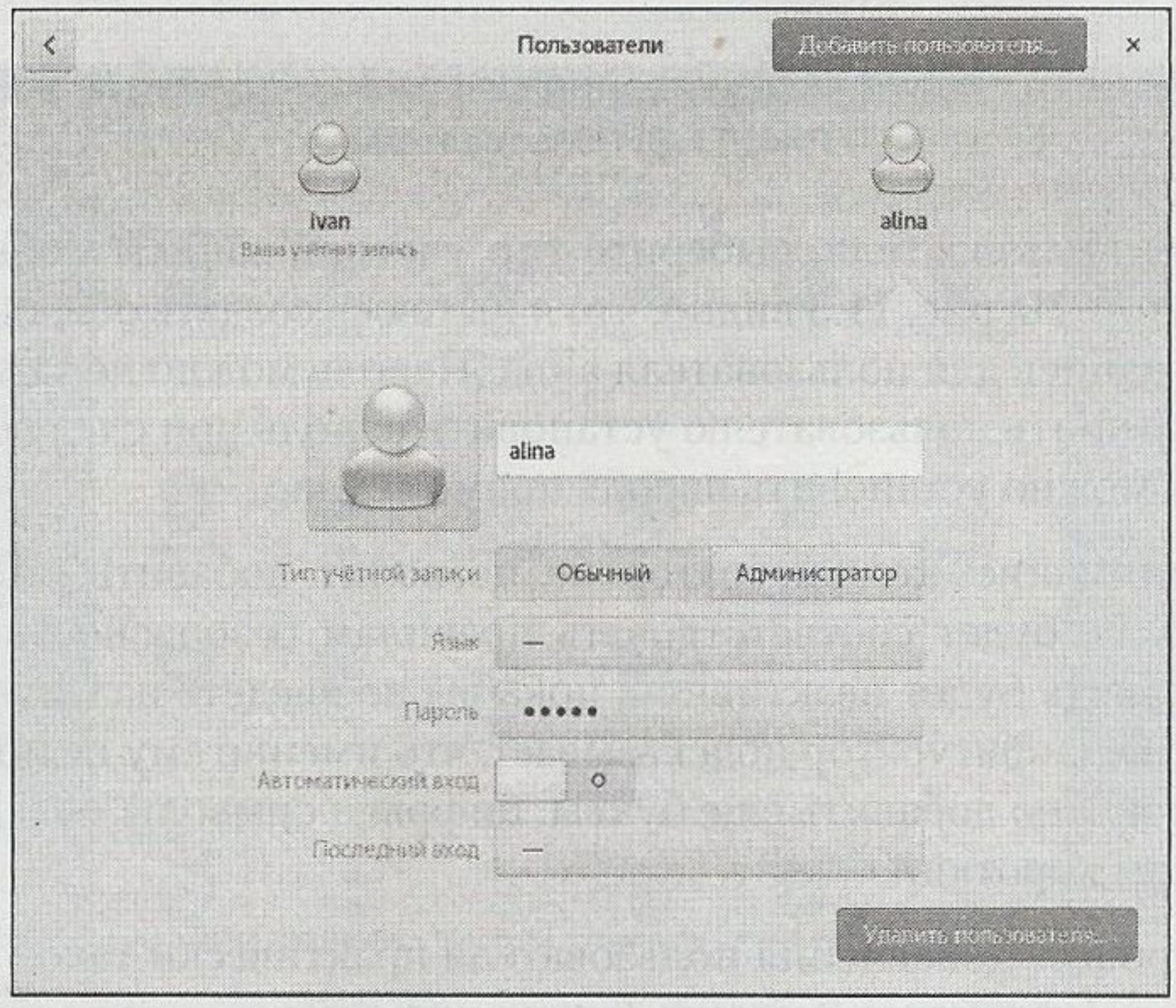


Рис. 11.4. Учетная запись пользователя создана

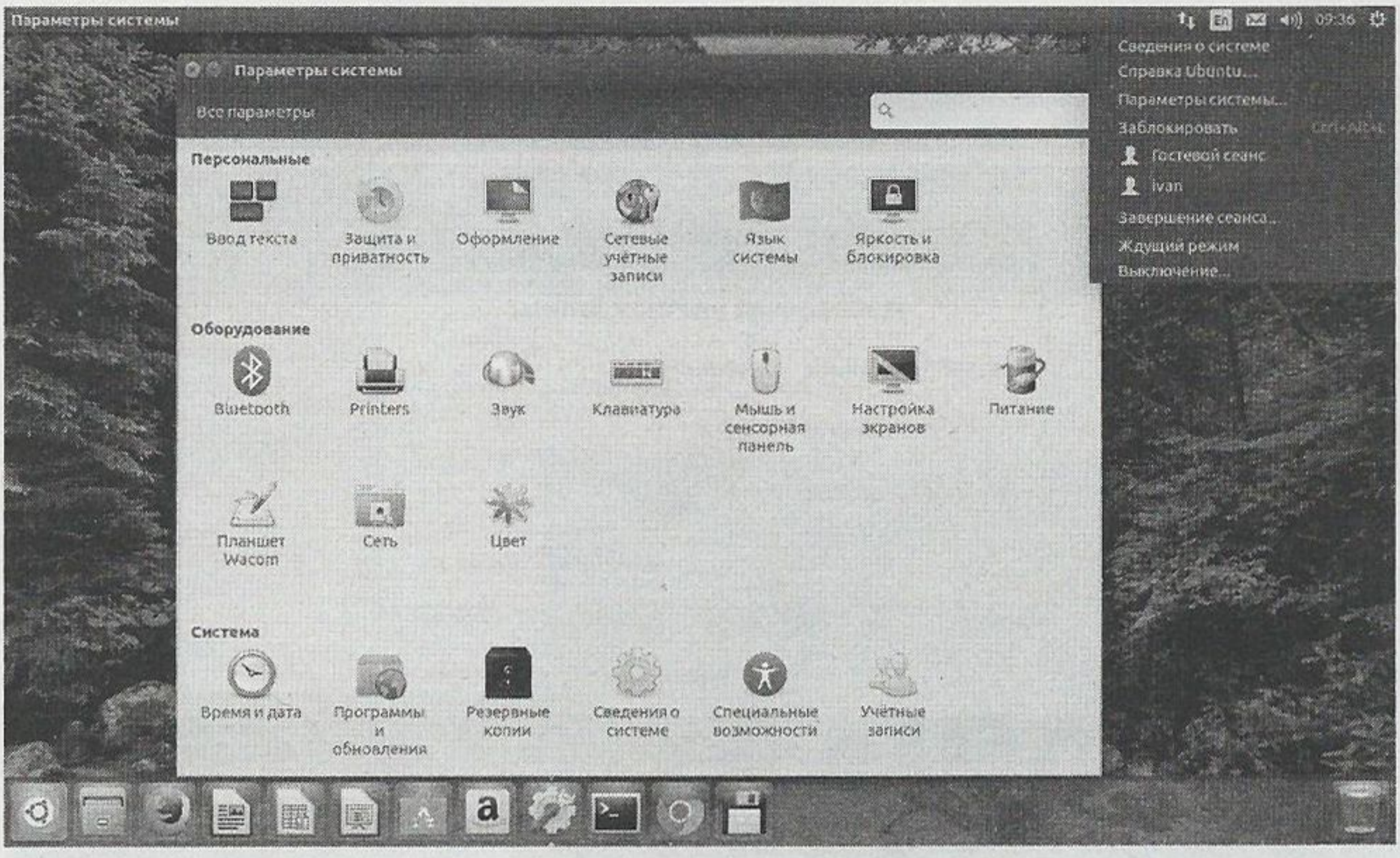


Рис. 11.5. Параметры системы

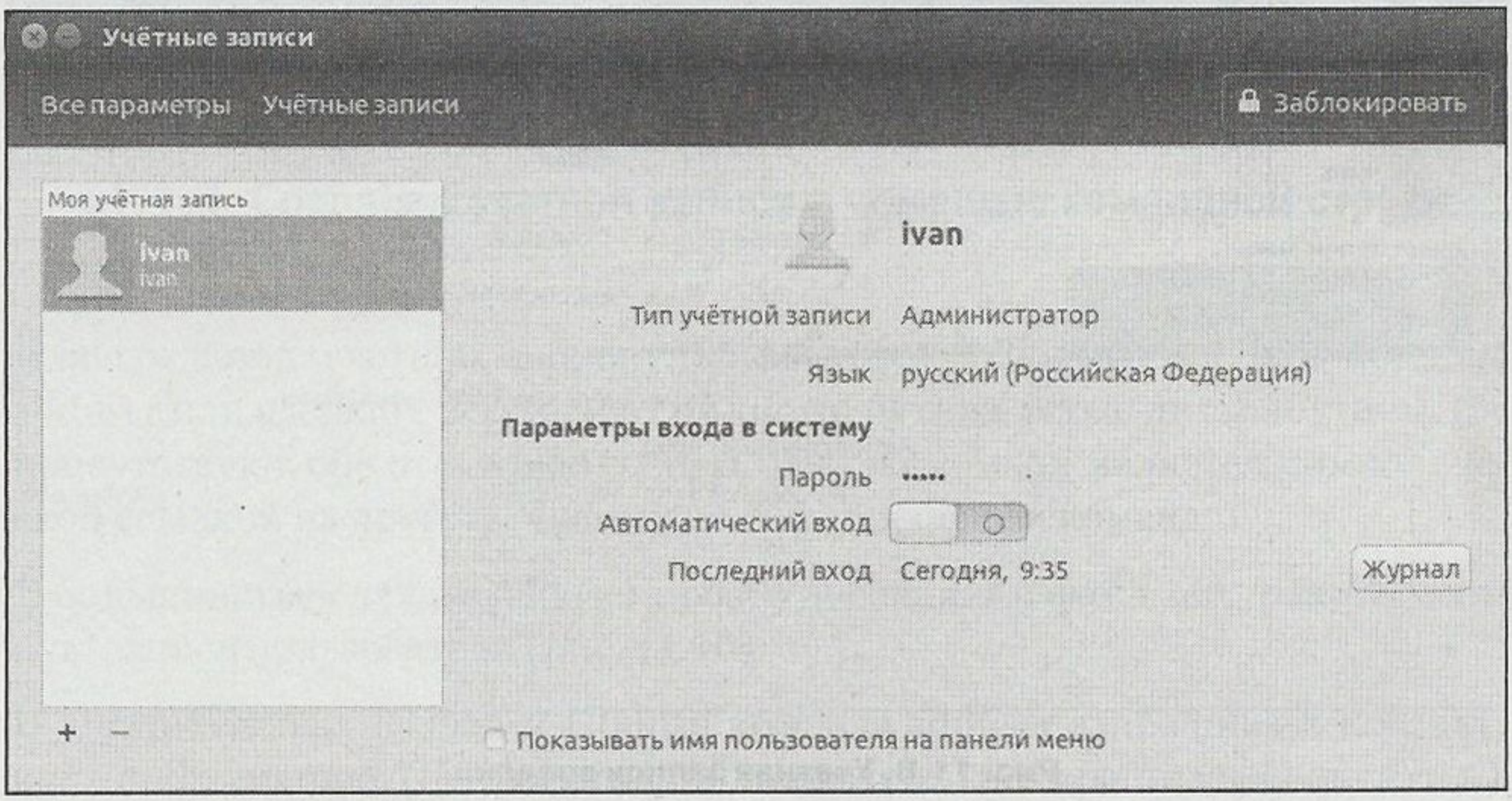


Рис. 11.6. Учетные записи

Окно создания учетной записи кардинально отличается. Вы можете выбрать только тип учетной записи (обычная или администратор), ввести реальное имя пользователя и его логин (рис. 11.7). Предполагается, что пароль пользователь установит при первом входе в систему.

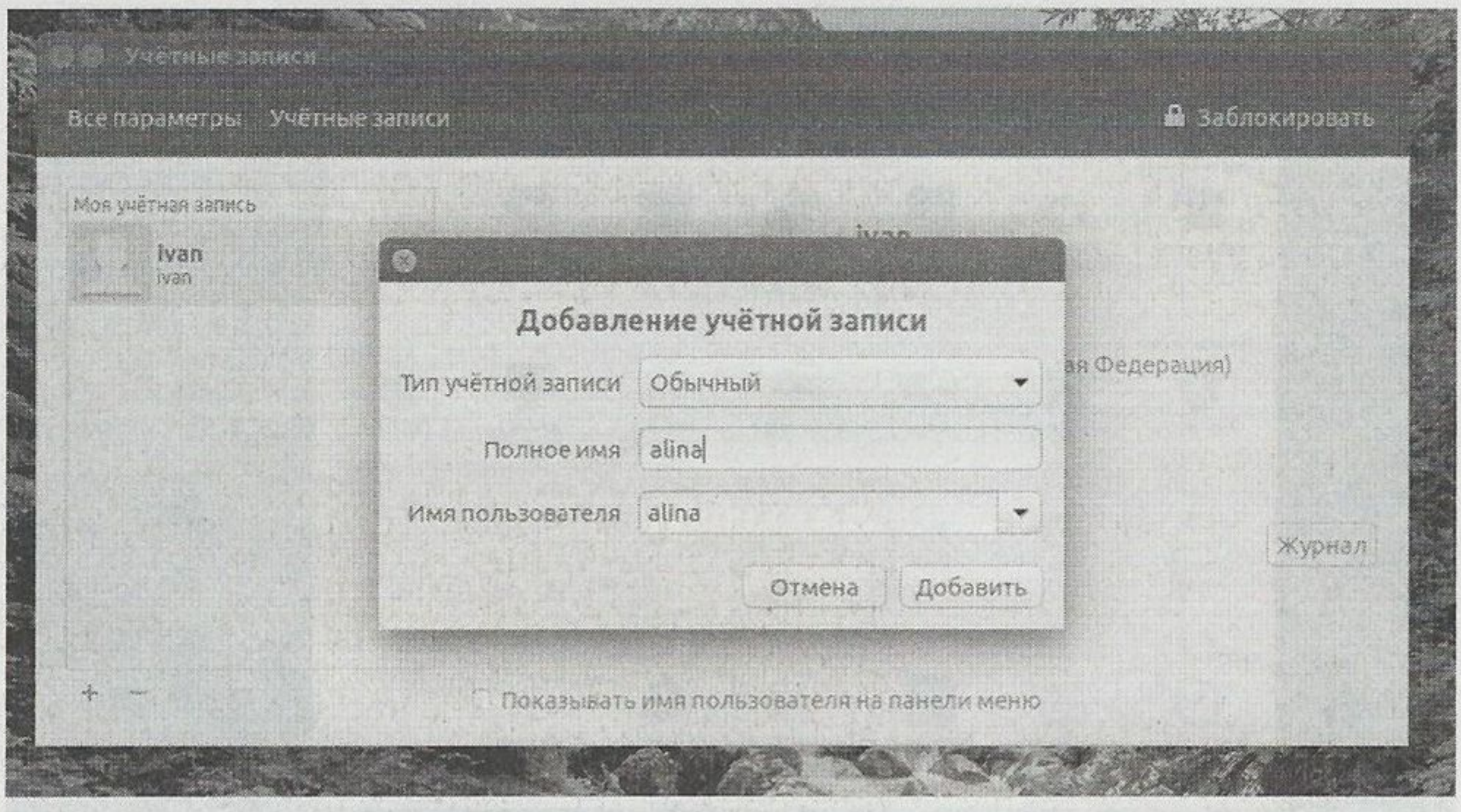


Рис. 11.7. Создание учетной записи

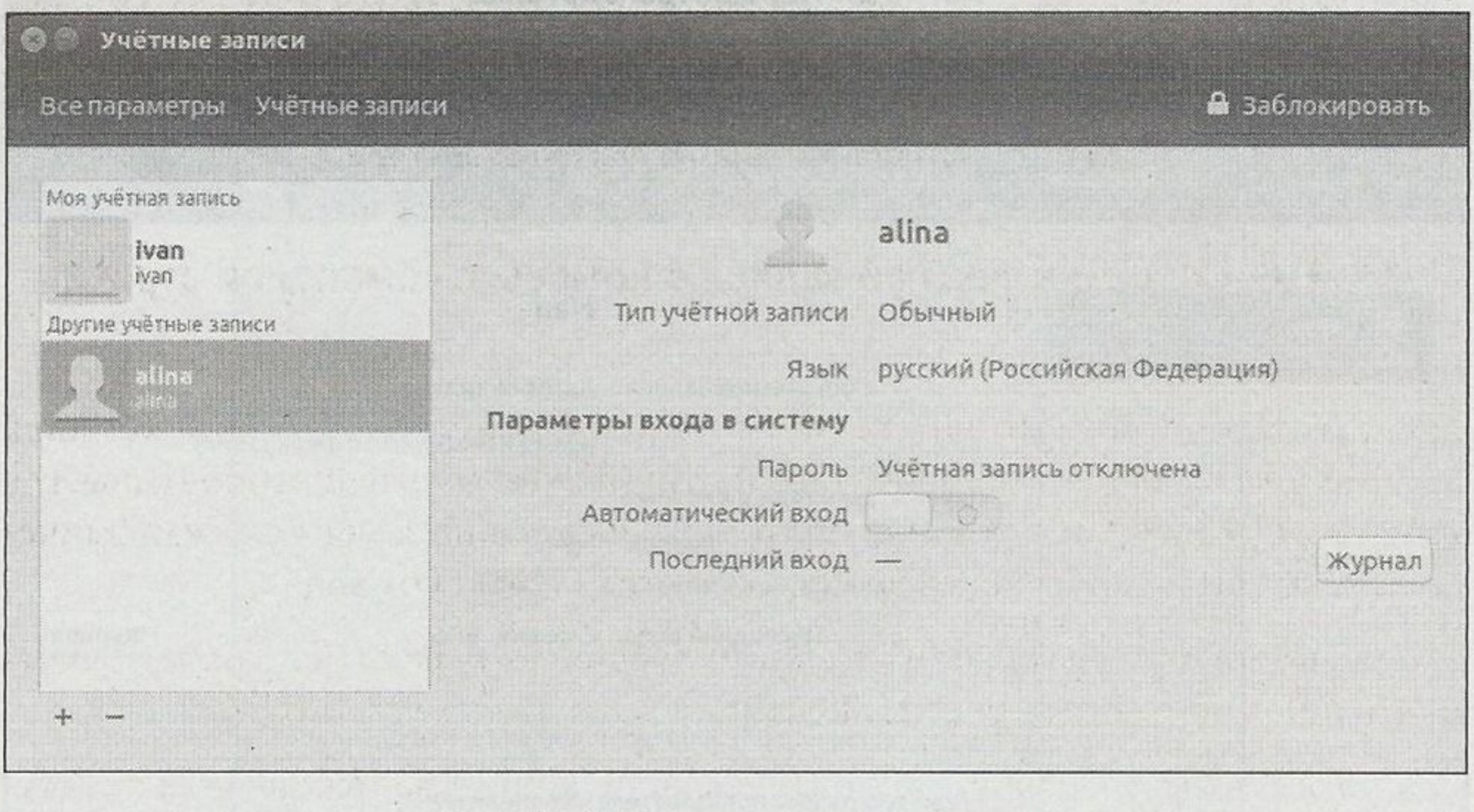


Рис. 11.8. Учетная запись создана

С одной стороны, такая концепция правильная - администратор не знает пароль. С другой стороны, а где гарантия, что первым в систему войдет пользователь, для которого эта учетная запись создавалась? Создать пароль для пользователя можно командой passwd (рис. 11.9). Введите команду:

```
sudo passwd <логин>
```

Сначала система запросит ваш пароль, затем пароль пользователя и его подтверждение.

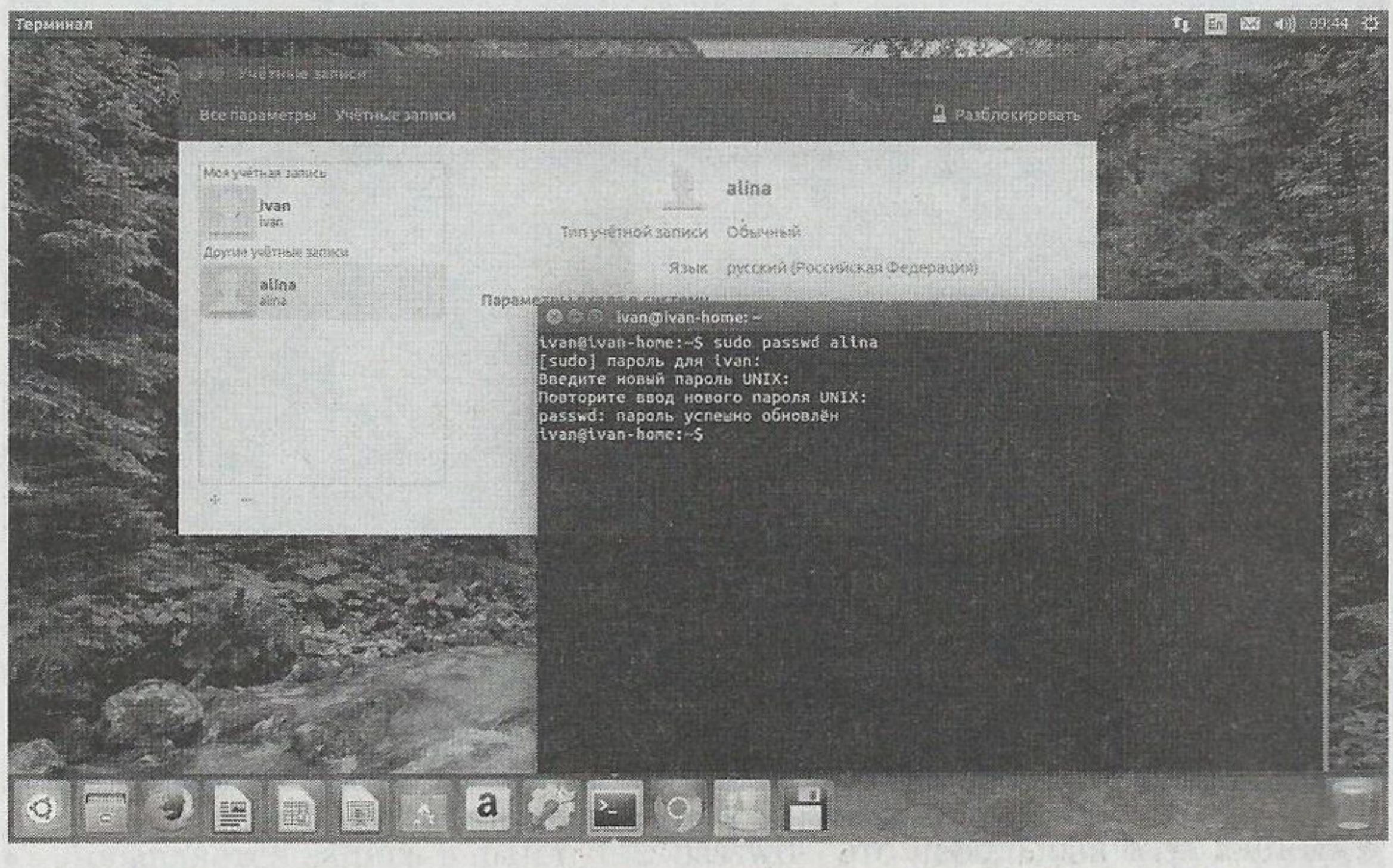


Рис. 11.9. Задание пароля для пользователя

11.1.4. Создание учетной записи с помощью командной строки

Для создания учетной записи пользователя используется или команда adduser или useradd - все зависит от дистрибутива, есть и дистрибутивы, где присутствуют обе эти команды или одна из команд является символической ссылкой на другую. Чаще используется первая команда.

В большинстве случаев adduser просто добавляет в файл /etc/passwd учетную запись пользователя (рис. 11.10).

В дистрибутивах Debian и Ubuntu команда adduser запрашивает контактную информацию (полное имя пользователя, номера телефонов и т.д.), а также сразу устанавливает пароль пользователя. Как по мне, команда adduser в Ubuntu гораздо удобнее, чем использование графической утилиты.

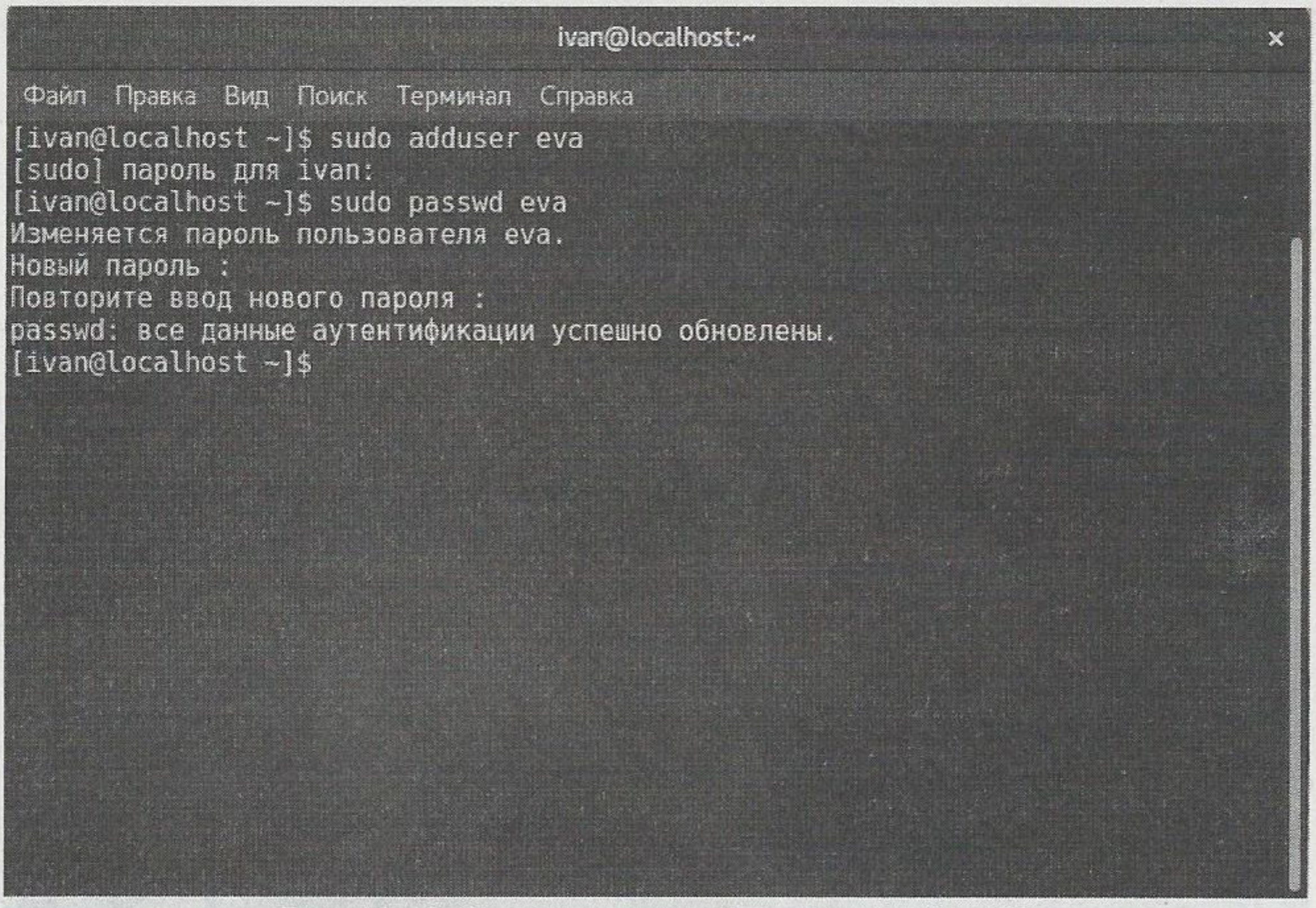


Рис. 11.10. Создание учетной записи в Fedora 26

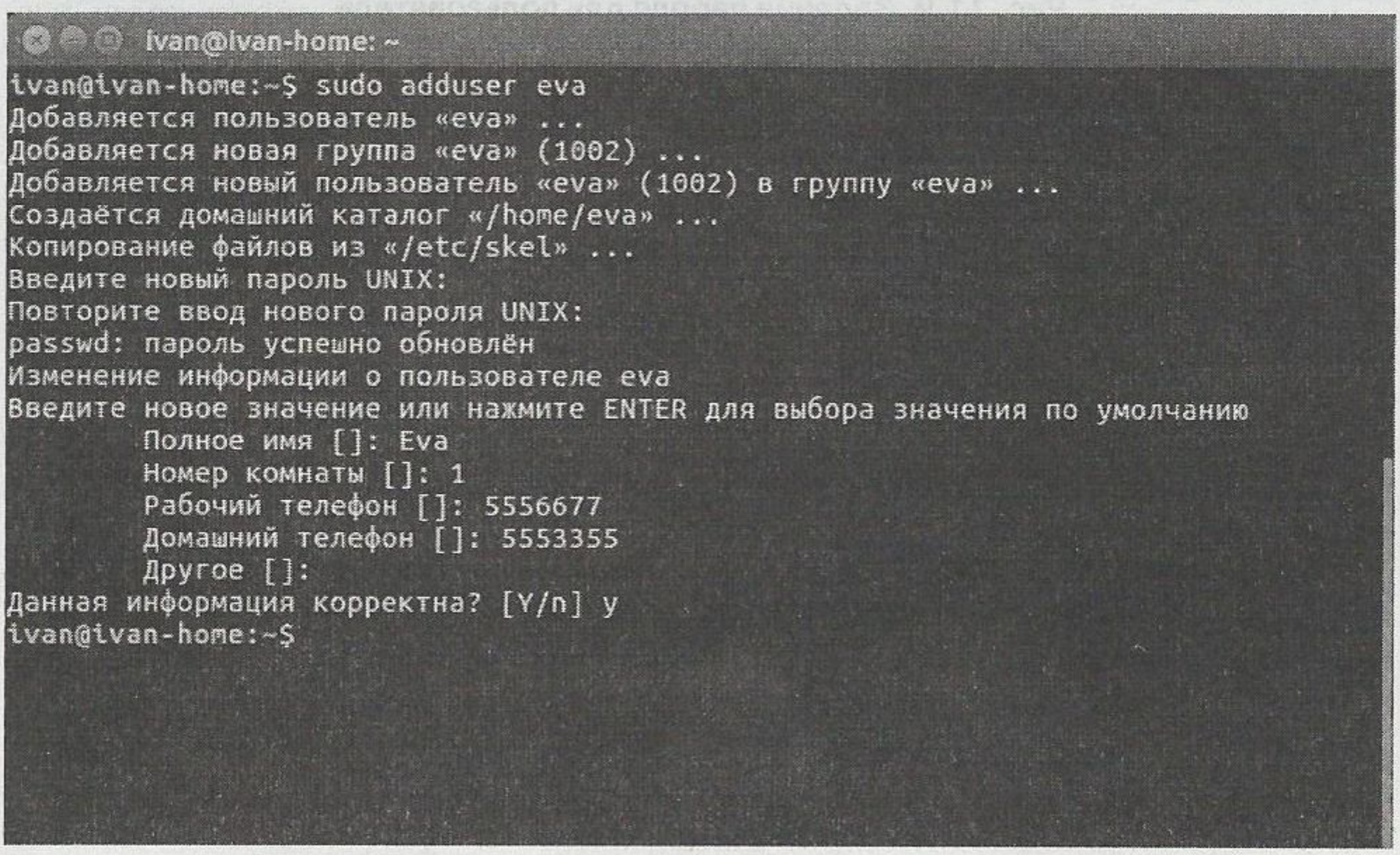


Рис. 11.11. Создание учетной записи в Ubuntu 17.04

Если в вашем дистрибутиве команда adduser не запросила пароль пользователя, а просто добавила его учетную запись, тогда вам нужно еще ввести команду passwd <имя пользователя> для установки его пароля (рис. 11.10), иначе пользователь не сможет войти в систему.

Итак, в одних дистрибутивах достаточно команды:

```
sudo adduser <имя>
```

В других же нужно ввести две команды:

```
sudo adduser <имя>
sudo passwd <пароль>
```

Напомню, что для добавления учетной записи пользователя нужны права root, который пользователь может получить через su или sudo, если он внесен в /etc/sudoers и ему разрешена операция добавления пользователя.

При добавлении учетной записи происходят следующие действия (см. рис. 11.11):

- Добавляется запись в файл /etc/passwd - это небольшая база данных о пользователях в текстовом формате. Этот файл могут просмотреть все пользователи.
- Если при создании учетной записи утилита запрашивает пароль, то он будет внесен в файл /etc/shadow. Пароли в этом файле хранятся в зашифрованном виде, а доступ имеет только root. Команда passwd <имя>, изменяющая пароль пользователя, вносит изменения как раз в этот файл.
- Создается домашний каталог /home/<имя> и в него копируется содержимое каталога /etc/skel.
- Создается почтовый ящик пользователя в каталоге /var/spool/mail.
- Владельцем каталога /home/<имя> и всех файлов и каталогов в нем назначается создаваемый пользователь.

Обратите внимание: если вы хотите добавить какой-то файл в каталог каждого из пользователей, просто поместите его в /etc/skel - при создании новой учетной записи система автоматически скопирует содержимое этого каталог в домашний каталог пользователя.

11.2. Модификация учетной записи

11.2.1. Графические инструменты

Для изменения учетных записей мы можем использовать те же инструменты, что и для их создания. Посмотрите на рис. 11.4 (Fedora) и 11.8 (Ubuntu). В Fedora просто можете ввести новое имя пользователя, выбрать тип учетной записи и щелкнуть по полю **Пароль**, чтобы установить новый пароль (рис. 11.12).

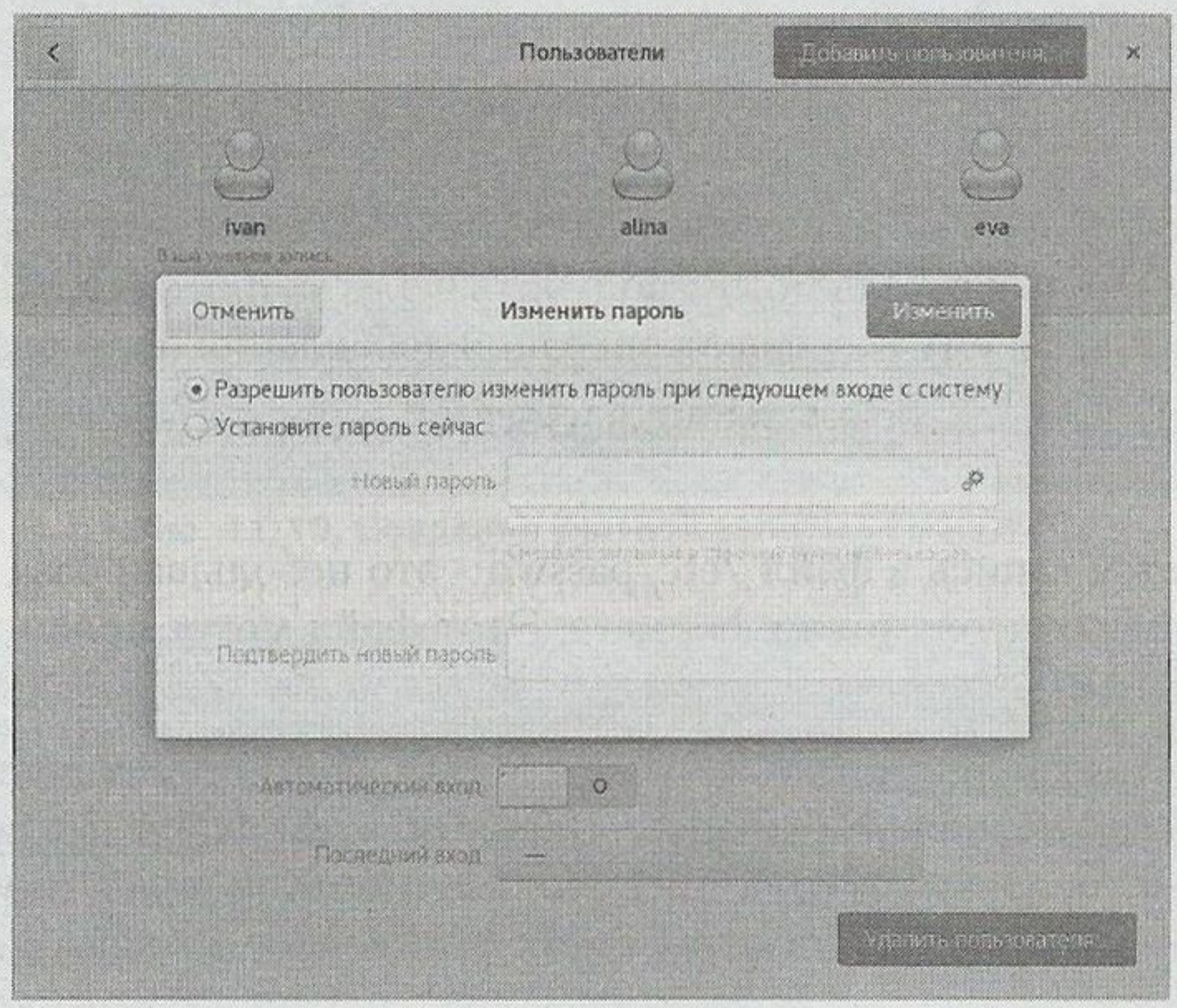


Рис. 11.12. Изменение пароля пользователя в Fedora

В Ubuntu нужно сначала разблокировать окно **Учетные записи**, выбрать учетную запись, которую вы хотите изменить и щелкнуть по полю, которое вы хотите изменить. На рис. 11.13 показано, как изменить логин пользователя.

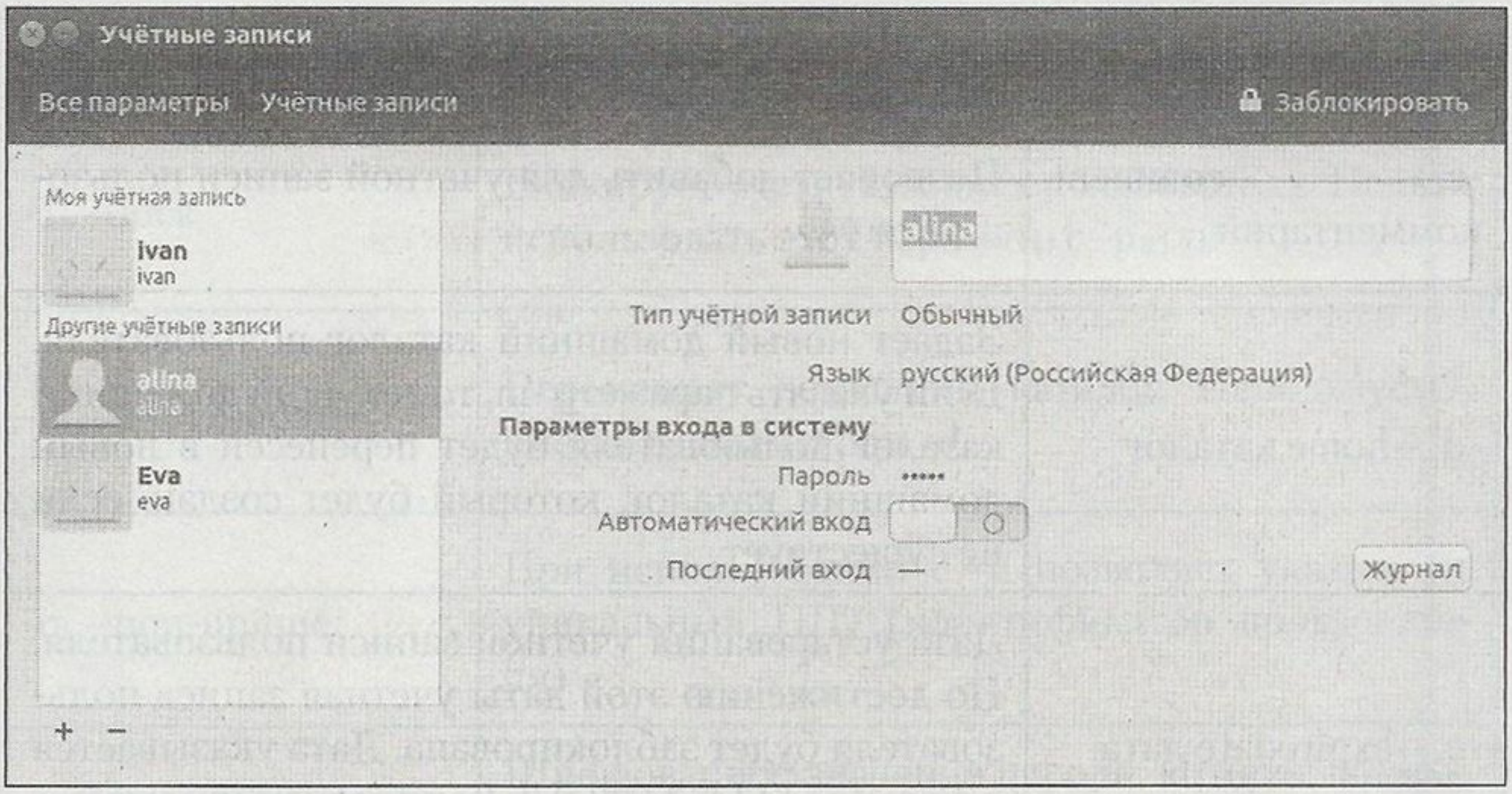


Рис. 11.13. Изменение учетной записи в Ubuntu

11.2.2. Команды usermod и passwd

Для установки и изменения пароля пользователя используется команда passwd:

```
sudo passwd <имя>
```

Если хотите изменить собственный пароль, то указывать имя не нужно:

```
$ passwd
```

Команда usermod используется для установки остальных (кроме пароля) параметров учетной записи. Формат вызова этой команды следующий:

```
sudo usermod [параметры] учетная_запись
```

Параметры команды usermod описаны в таблице 11.1.

Таблица 11.1. Параметры команды usermod

Параметр	Описание
-a, -append	Добавляет пользователя в дополнительную группу. Используется только с параметром -G

Параметр	Описание
-c, --comment комментарий	Позволяет добавить для учетной записи пользователя
-d, --home каталог	Задает новый домашний каталог пользователя. Если указать параметр -m, то текущий домашний каталог пользователя будет перенесен в новый домашний каталог, который будет создан, если не существует
-e, --expiredate дата	Дата устаревания учетной записи пользователя. По достижению этой даты учетная запись пользователя будет заблокирована. Дата указывается в формате ГГГГ-ММ-ДД. Если дату не указывать, то устаревание учетной записи будет отключено.
-f, --inactive дни	После указанного числа дней, которые пройдут после устаревания пароля, учетная запись будет блокирована. Значение -1 означает, что эта возможность не используется, а 0 - запись будет заблокирована сразу же после устаревания пароля.
-g, --gid группа	Указывает имя или GID первичной группы пользователя. Группа с таким именем/GID должна существовать. Все файлы в домашнем каталоге пользователя, которые принадлежали бывшей первичной группе, теперь будут принадлежать новой группе.
-G, --groups группа1[, группа2,..., группаN]	Список дополнительных групп, в которых находится пользователь. Перечисление групп осуществляется через запятую без дополнительных пробелов. Например, -G group1,group2
-l, --login новое_имя	Изменяет имя пользователя на новое_имя.

Параметр	Описание
-L, --lock	Блокирует учетную запись пользователя. Нельзя использовать этот параметр с -p или -I
-m, --move-home	Перемещает домашний каталог. Используется вместе с параметром -d
-o, --non-unique	При использовании с -u позволяет указать не уникальный UID (идентификатор пользователя)
-p, --password пароль	Шифрованное значение пароля, которое возвращает функция crypt. Использовать этот параметр не рекомендуется, поскольку другие пользователи увидят незашифрованный пароль в списке процессов.
-R, -root chroot	Выполняет изменения в каталоге chroot и использует файлы конфигурации из этого каталога
-s, --shell оболочка	Задает оболочку для пользователя. Если оболочка не указана, то будет использована оболочка по умолчанию
-u, --uid UID	Задает новый UID пользователя, который должен быть уникальным.
-U, --unlock	Разблокирует учетную запись пользователя

Рассмотрим несколько примеров:

```
# usermod -d /home/new_home -m eva
# usermod -L ava
# usermod -G sales,sudo alina
```

Первая команда задает новый каталог для пользователя eva. Теперь он будет называться /home/new_home. Старые файлы (из каталога /home/eva) будут перемещены в новый домашний каталог.

Вторая команда блокирует учетную запись пользователя eva. Третья команда вносит пользователя alina в группы sales и sudo.

11.2.3. Команда visudo

Чтобы сделать пользователя администратором (то есть предоставить ему право использовать команду sudo), можно или использовать графическую утилиту или команду visudo.

Редактировать файл /etc/sudoers можно только через команду visudo (рис. 11.14):

```
export EDITOR=joe
sudo visudo
```

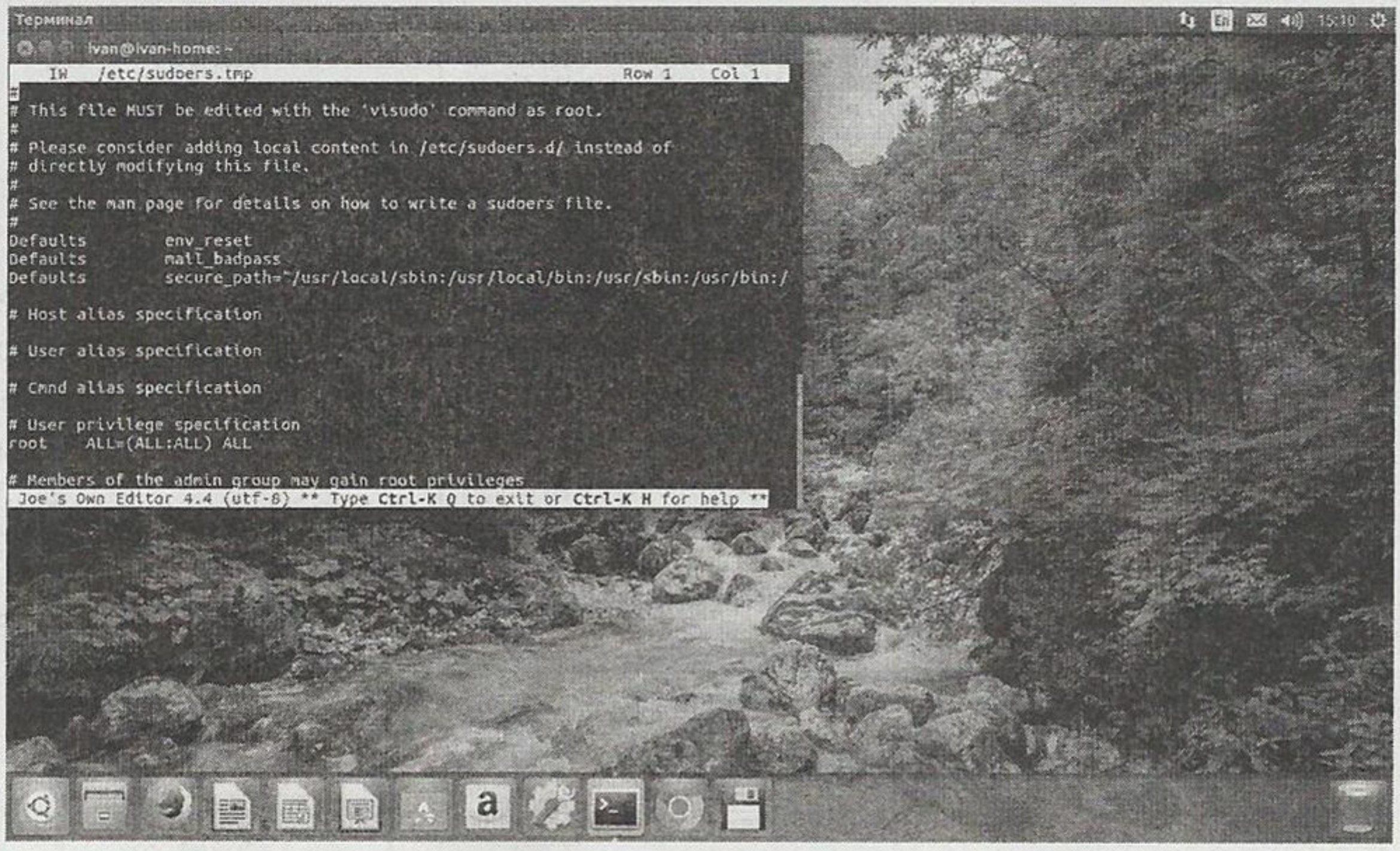


Рис. 11.14. Редактирование файла /etc/sudoers

Первая команда устанавливает переменную окружения EDITOR, задающую удобный текстовый редактор, который будет использован для /etc/sudoers. Вторая команда вызывает утилиту для редактирования файла /etc/sudoers.

Представим, что у нас есть пользователь eva, которому нужно разрешить делать все, что можно пользователю root. Для этого нужно добавить в /etc/sudoers запись вида:

```
eva   ALL=(ALL:ALL) ALL
```

Можно также добавить запись:

```
%sudo   ALL=(ALL:ALL) ALL
```

Она означает, что членам группы sudo можно делать все, что можно делать пользователю root. Тогда всех администраторов-помощников нужно добавить в группу sudo.

Сохраните файл и выйдите из редактора. Войдите как пользователь, которому вы предоставили право sudo. В нашем случае - это пользователь eva. Далее введите команду, которая требует прав root через команду sudo:

```
sudo <команда>
```

Например:

```
sudo apt install nano
```

Контролировать получение прав sudo можно командой[1]:

```
# tail /var/log/secure | grep sudo
```

Посмотрите на рис. 11.15. 23 апреля 2017 года в 15:09 пользователь ivan пытался выполнить команду sudo, но у него ничего не получилось, так как его не было в файле /etc/sudoers. Учтите, что если настроена ротация журналов, то в auth.log будут не все записи, а только за текущие сутки (или за другой период - в зависимости от настроек ротации)

Сеансы sudo еще более безопасные, чем сеансы su, поскольку они автоматически закрываются - ведь пользователь может отвлечься и отойти, а если тем временем кто-то может этим воспользоваться.

---

1 В некоторых старых дистрибутивах журнал аутентификации называется не secure, а auth.log

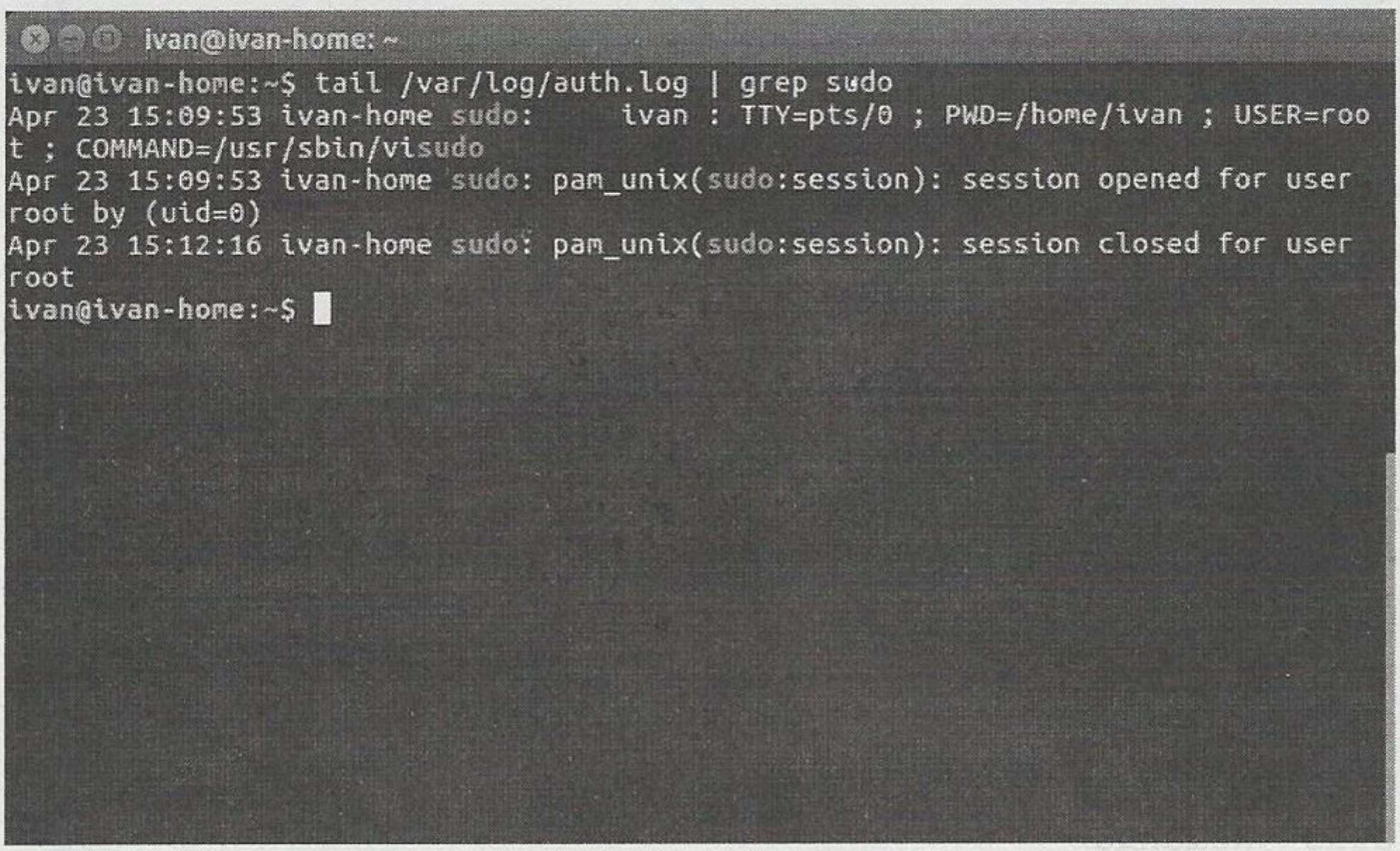


Рис. 11.15. Журнал аутентификации auth.log

11.3. Удаление учетной записи

Для удаления учетной записи нужно нажать кнопку Удалить пользователя (см. рис. 11.4) в Fedora или кнопку «-» в Ubuntu. В командной строке для удаления пользователя используется команда userdel.

Формат команды userdel (см. табл. 11.2):

```
sudo userdel [параметры] пользователь
```

Таблица 11.2. Параметры команды userdel

Параметр	Описание
-f, --force	Удаляет учетную запись, даже если пользователь работает в системе. Также будет удален домашний каталог и почтовый ящик, даже если другой пользователь использует тот же домашний каталог. Если в файле /etc/login.defs параметр USERGROUPS_ENAB равен yes, то будет удалена и первичная группа пользователя, даже если она является первичной и для другого пользователя. Довольно опасный параметр, который может привести систему в нерабочее состояние.

-r, --remove	Удаляет домашний каталог пользователя и почтовый ящик. Файлы этого пользователя, созданные на других файловых системах, нужно искать и удалять вручную.
-R, --root chroot	Выполняет изменения в каталоге chroot и использует файлы конфигурации из этого каталога

Пример удаления учетной записи eva, домашний каталог и почтовый ящик также будут удалены:

```
sudo userdel -r eva
```

11.4. Модули аутентификации

11.4.1. Знакомство с PAM

Подключаемые модули аутентификации PAM (Pluggable Authentication Modules) предоставляют администраторам дополнительные методы подтверждения подлинности пользователя.

Модули аутентификации были разработаны довольно давно, но используются и по сей день, даже в самых современных дистрибутивах, поскольку заменить их нечем - ничего другого на данный момент не придумано.

Модули PAM позволяют использовать несколько схем аутентификации. Большинство приложений, которые нуждаются в проверке подлинности пользователя, используют PAM. Модули PAM позволяют реализовать альтернативную аутентификацию, например, по отпечаткам пальцев или по сетчатке глаз, но для этого необходимо дополнительное оборудование, например, сканер отпечатков. В этой книге мы рассмотрим традиционный вариант использования PAM - когда аутентификация происходит посредством ввода пароля с клавиатуры.

11.4.2. Установка минимальной длины пароля

В целях безопасности современные дистрибутивы не позволяют установить слишком короткий пароль (меньше 8 символов). Для изменения длины пароля нужно отредактировать файл /etc/pam.d/common-password:

```
sudo joe /etc/pam.d/common-password
```

Найдите в этом файле строку:

```
password      [success=1 default=ignore]     pam_unix.so obscure sha512
```

Замените эту строку следующей строкой:

```
password      [success=1 default=ignore]     pam_unix.so sha512 minlen=10
```

Здесь мы убрали опцию obscure, которая требует введения сложного пароля (при желании ее можно оставить), но задали минимальную длину пароля - 10 символов.

Сохраните файл и выйдите из редактора. Больше никаких действий предпринимать не нужно.

11.4.3. Ограничиваем доступ к системе

Параметры PAM, относящиеся к ограничению доступа, находятся в каталоге /etc/security. В этом каталоге достаточно много различных файлов конфигурации, поэтому мы рассмотрим только некоторые из них. Начнем с файла access.conf, в котором описываются ограничения доступа к системе.

Формат этого файла следующий:

```
разрешения : пользователи : источники
```

Разрешение может начинаться с символа «+» (доступ разрешен) или «-» (доступ запрещен). Если нужно указать несколько пользователей, то их имена разделяют пробелом. Если нужно сделать исключение для некоторых пользователей, то перед их именами указывают служебное слово EXCEPT.

Третье поле может содержать список из одного или более имен консолей (tty) - для несетевого доступа к системе, имен узлов (для сетевого доступа), доменных имен (начинаются с «.»), IP-адресов узлов, IP-адресов сетей (заканчиваются «.»). Также можно указать все источники (ALL), ни один из источников (NONE) или только локальные источники (LOCAL).

Теперь несколько примеров:

```
-:ALL EXCEPT root:tty1
```

Первая консоль - это только консоль root. Другим пользователям запрещено ее занимать. Мы запрещаем доступ (-) всем пользователям (ALL) кроме (EXCEPT) пользователя root на консоли tty1.

Следующий пример - разрешение регистрации как root с определенных IP-адресов:

```
+ : root : 192.168.1.1 192.168.1.4 192.168.1.9
+ : root : 127.0.0.1
```

Если нужно разрешить регистрацию root со всей подсети 192.168.1.0, тогда укажите адрес этой подсети, указав точку вместо 0:

```
+ : root : 192.168.1.
```

Самый жесткий пример - запрещаем root вообще входить в систему:

```
- : root : ALL
```

> **Примечание**. Обратите внимание, что комментарии в этом файле начинаются с #, если вы хотите использовать один из примеров, приведенных в файле, убедитесь, что вы раскомментировали нужную вам строку.

Чуть выше мы разрешили вход пользователя root с определенных IP-адресов. К сожалению, одного только редактирования access.conf будет недостаточно. Нужно еще отредактировать соответствующие файлы в /etc/pam.d. Нас интересует регистрация по SSH (telnet уже не используется, поэтому вы будете регистрироваться по SSH) и обычная регистрация в системе. Поэтому нам нужно отредактировать файлы /etc/pam.d/sshd и /etc/pam.d/system-auth. В этих файлы вам нужно добавить строчку:

```
account   required /lib64/security/pam_access.so
```

Если у вас 32-разрядная система, тогда нужно добавить немного другую строку:

```
account   required /lib/security/pam_access.so
```

11.4.4. Разрешаем регистрацию только в рабочее время

Безопасностью системы лучше управлять в бизнес-время. Поэтому если вы настраиваете корпоративный веб-сервер, который пользователи могут использовать только в рабочее время, имеет смысл разрешить регистрацию только в это время, например, с 8:00 до 20:00 (на тот случай, если кому-то нужно будет задержаться).

Откройте файл /etc/security/time.conf и добавьте в него строку:

```
login;tty* & !ttyp*; !root & admin & ; !Al0800-2000
```

Здесь мы разрешаем пользователям регистрироваться только с 8:00 по 20:00. На пользователей root и admin это правило не распространяется. Также в файле time.conf вы найдете еще несколько примеров.

Затем вам нужно изменить файлы /etc/pam.d/sshd и /etc/pam.d/system-auth, в которые нужно добавить строку:

```
account   required  /lib64/security/pam_time.so
```

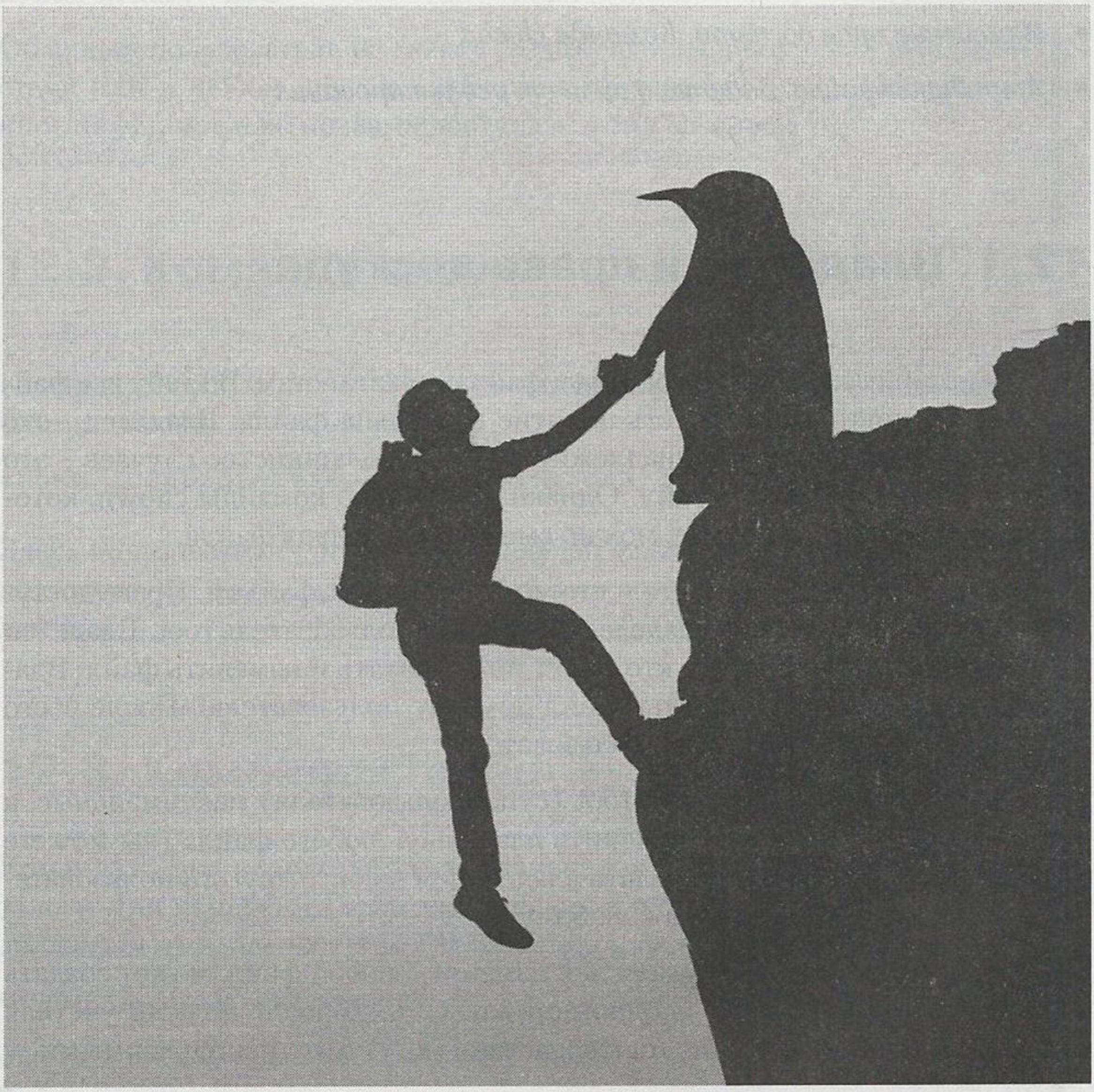
или строку (для 32-разрядной системы):

```
account   required  /lib/security/pam_time.so
```

В следующей главе мы продолжим тему безопасности и поговорим об ограничении прав доступа к файлам и каталогам.

ГЛАВА 12.

Права доступа к файлам и каталогам



В этой главе:

- *Владелец и права доступа*
- *Изменение владельца командой chown*
- *Изменение прав доступа. Команда chmod*
- *Атрибуты файла. Защита файла от редактирования*

12.1. Владелец и права доступа

Предмет этой главы - права доступа к файлам и каталогам. В Linux все файлы кому-то принадлежат - есть понятие владельца файла. Владелец - это пользователь, которому принадлежит файл. В большинстве случаев - это пользователь, создавший файл. Однако с помощью команды chown, которая будет рассмотрена позже, можно сменить владельца файла.

Права доступа определяют, кто и что может сделать с файлом. Права доступа файла может изменять владелец файла или пользователь root. Владелец может назначить, например, кто имеет право читать и изменять файл. Владелец также может «подарить» файл другому пользователю. После этого владельцем станет уже другой пользователь.

Права доступа у пользователя root (суперпользователя) максимальные, а это означает, что он может изменить владельца любого файла (вы можете создать файл, а root может сделать владельцем любого другого пользователя) и изменить права доступа к любому фалу.

Суперпользователь может удалить и изменить любой файл, может создать файл в любой папке и т.д. С одной стороны, это хорошо, но если учетная запись root попадет в чужие, то последствия могут быть непоправимыми.

Права доступа в Linux по умолчанию настроены так, что пользователь владеет только своим домашним каталогом /home/<имя_пользователя>. Поэтому создавать файлы и выполнять другие операции по работе с файлами (удаление, редактирование, копирование и т.д.) пользователь может только в этом каталоге и то при условии, что файлы принадлежат ему.

Если в домашнем каталоге пользователя root создал файл, пользователь не сможет удалить или изменить его, поскольку он не является его владельцем. Сможет ли он прочитать этот файл, зависит от прав доступа к файлу (о них мы поговорим позже).

Остальные файлы, которые находятся за пределами домашнего каталога, пользователь может только просмотреть и то, если это не запрещено правами доступа. Например, файл /etc/passwd пользователь может просмотреть, а /etc/shadow - нет.

Обычный пользователь не может создать файлы в корневой файловой системе или в любом другом каталоге, который ему не принадлежит, если иное не установлено правами доступа к этому каталогу.

12.2. Команда chown

Команда chown используется для изменения владельца файла/каталога. Формат такой:

```
chown <пользователь> <файл/каталог>
```

Здесь пользователь - это новый владелец файла. Чтобы подарить другому пользователю файл, вы должны быть или его владельцем или пользователем root.

12.3. Команда chmod

Команда chmod используется для изменения прав доступа к файлу или каталогу. Для изменения прав доступа, как и в случае с командой chown, вы должны быть владельцем файла/каталога или же пользователем root.

Формат команды следующий:

```
chmod <права> <файл/каталог>
```

Права доступа состоят из трех наборов:

- для владельца;
- для группы владельца;
- для прочих пользователей.

Первый набор задает возможности владельца файла, второй - для группы пользователей, в которую входит владелец и третий - для всех остальных пользователей.

В каждом наборе может быть три права - чтение (r), запись (w) и выполнение (x). Для файла право на выполнение означает возможность запустить этот файл на выполнение (обычно используется для программ и сценариев). Право выполнение для каталога - возможность просматривать этот каталог.

Права доступа в наборе определяются четко в определенном порядке и могут быть представлены, как символьном, так и числовом виде (в двоичной или восьмеричной системе). Рассмотрим несколько наборов прав доступа:

- 100 - только чтение
- 110 - чтение и запись
- 101 - чтение и выполнение
- 111 - чтение, запись, выполнение

Учитывая, что права доступа задаются для владельца, группы и остальных пользователей, полный набор прав доступа может выглядеть так:

```
110 100 000
```

В этом наборе мы предоставляем полный доступ (кроме выполнения) владельцу, группе владельца разрешено только чтение, остальным пользователям доступ к файлу/каталогу вообще запрещен.

Примечание. Право на выполнение имеет смысл задавать только для программ и сценариев командной оболочки. Для остальных файлов в этом праве нет смысла.

В двоичной системе права доступа мало кто записывает. В основном их преобразуют в восьмеричную систему. Для тех кто не знаком с ней, воспользуйтесь таблицей 12.1

Таблица 12.1. Преобразование из двоичной в восьмеричную систему

Двоичная система	Восьмеричная система
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Итак, право 666 означает полный доступ к обычному файлу для всех трех групп пользователей, 660 - полный доступ только для владельца и группы, а остальным пользователям запрещено даже чтение этого файла. Обычно для файлов устанавливаются права 664, то есть владельцу и группе разрешен полный доступ, а вот все остальные могут только читать файл.

Для каталога полные права доступа выглядят как 777 - чтение, изменение и просмотр каталога для владельца, группы и прочих пользователей.

Просмотреть текущие права доступа можно командой ls -l <файл/каталог>, например:

```
ls -l file.txt

-rw-r--r-- 1 root root 2179 Map 16 08:28 file.txt
```

Как мы видим, задано три набора rw-, r--, r--. Выходит, владельцу разрешена запись и чтение файла, остальным пользователям (группа и прочие) - только чтение. В восьмеричной системе этот набор прав доступа выглядит как 644.

Первый символ (в нашем случае это -) является признаком каталога. Если бы мы выводили права доступа каталога, то вместо - здесь был бы символ d. Для файла выводится просто «-».

Символьный способ задания прав доступа немного проще. Рассмотрим пример его использования:

```
chmod +x program
```

Просмотрим опять права доступа:

```
ls -l program
```

```
-rwxr-xr-x 1 root root 1176 Map 19 18:28 program
```

Как видите, право выполнение было добавлено во все три набора прав доступа.

12.4. Специальные права доступа

В Linux есть еще специальные права доступа SUID (Set User ID root) и SGID (Set Group ID root), позволяющие обычным пользователям запускать программы, которые требуют для своей работы прав root.

В современных дистрибутивах Linux вам придется изменять эти права доступа чрезвычайно редко (может быть даже вообще никогда), но вам нужно знать, как их изменить. Например, если программу /usr/sbin/program вы хотите разрешить запускать с правами root обычным пользователям, установите права доступа так:

```
# chmod u+s /usr/sbin/program
```

Использование SUID - плохое решение с точки зрения безопасности. Правильнее использовать команду sudo, если какому-то пользователю будут нужны права root (см. гл. 14).

12.5. Команда chattr

В Linux кроме прав доступа есть еще и атрибуты файла, подобно атрибутам файла в других операционных системах. Изменить атрибуты файла можно командой chattr:

```
chattr +/-<атрибуты> <файл>
```

Просмотреть установленные атрибуты можно командой lsattr:

```
lsattr <файл>
```

Некоторые полезные атрибуты файлов приведены в таблице 12.2.

Таблица 12.2. Полезные атрибуты файлов

Атрибут	Описание
i	Данный атрибут запрещает изменение, переименование и удаление файла. Его можно установить для критических конфигурационных файлов или для каких-либо других критических данных. Установить (как и сбросить) этот атрибут может только пользователь root или процесс с возможностью CAP_LINUX_IMMUTABLE. Другими словами, сбросить этот атрибут просто так нельзя - нужны только права root.
u	При удалении файла с установленным атрибутом u его содержимое хранится на жестком диске, что позволяет легко восстановить файл.
c	Файл будет сжиматься. Можно установить этот атрибут для больших файлов, содержащих несжатые данные. Доступ к сжатым файлам будет медленнее, чем к обычным, поэтому плохое решение устанавливать этот атрибут для файлов базы данных. Этот атрибут нельзя устанавливать для файлов, уже содержащих сжатые данные - архивы, JPEG-фото, MP3/MP4-файлы и т.д. Этим вы не только не уменьшите его размер, но и замедлите производительность.
S	Данные, записываемые в файл, сразу будут сброшены на диск. Аналогично выполнению команды sync сразу после каждой операции записи в файл.
s	Прямо противоположен атрибуту u. После удаления файла принадлежащие ему блоки будут обнулены и восстановить их уже не получится.

Пример установки атрибута:

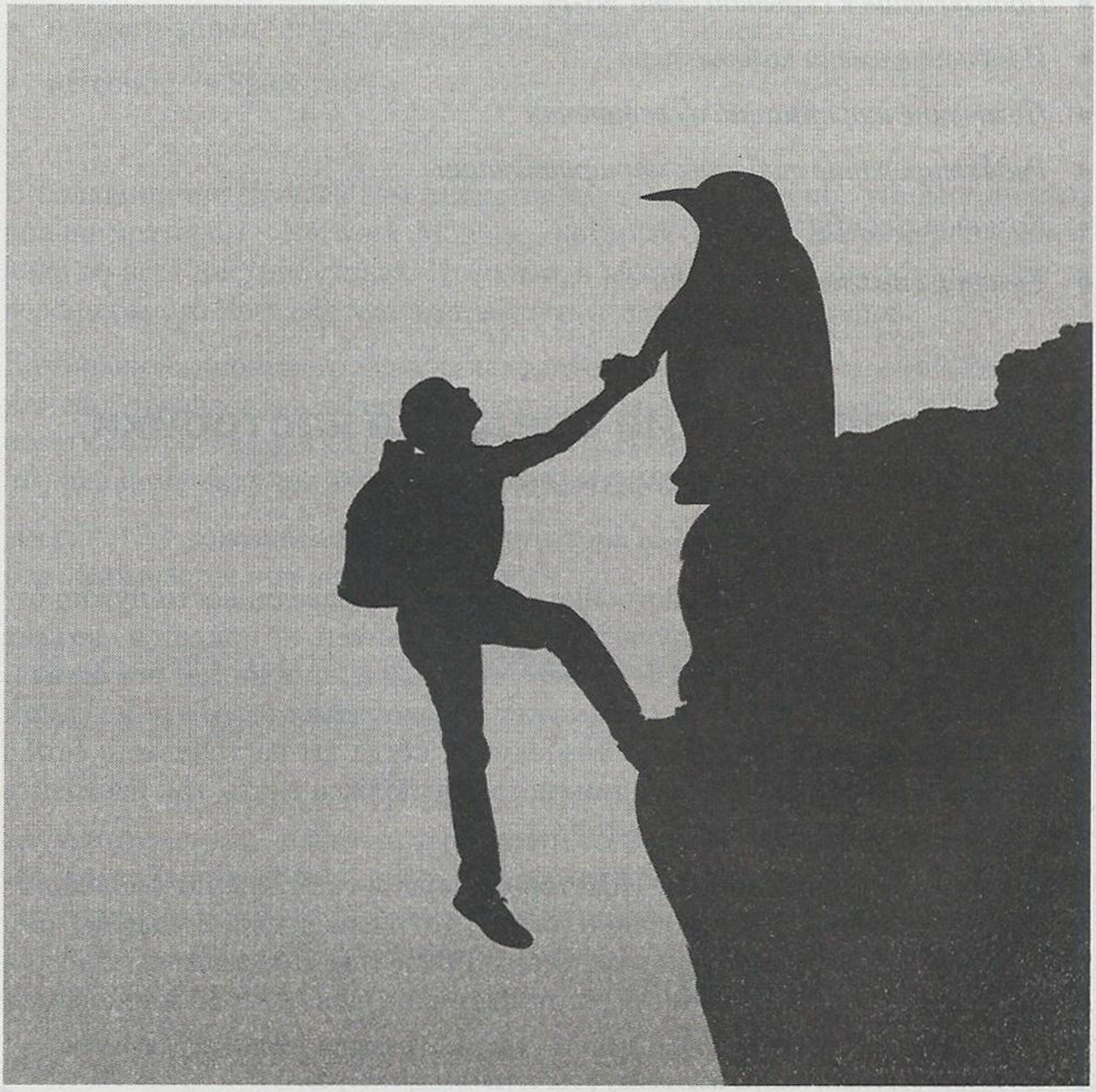
```
# chattr +i /etc/pam.d/time.conf
```

Пример сброса атрибута:

```
# chattr -i /etc/pam.d/time.conf
```

ГЛАВА 13.

Управление сетевыми подключениями



В этой главе:

- *Сервис NetworkManager*
- *Установка сетевых параметров вручную*
- *Подключение к скрытой сети Wi-Fi*
- *Изменение имени компьютера*
- *Получение информации о соединении*
- *Редактирование таблицы маршрутизации*
- *Команда ifconfig*
- *Команды диагностики сети*

13.1. Графические средства настройки сетевого подключения

Прежде, чем приступить к настройке сети, ради справедливости нужно отметить, что в большинстве случаев настраивать ничего не придется - во всех современных сетях есть DHCP-сервер[1], который и настраивает все остальные компьютеры. Настройка сети может понадобиться разве что на самом DHCP-сервере, но это только в том случае, когда вы настраиваете сеть с нуля. Однако в любом случае администратор должен знать, как настроить сеть, иначе какой он администратор?

Конечно, есть и исключения. Некоторые маршрутизаторы по умолчанию не используют DHCP (откровенно говоря, это очень древние модели), они используют статические IP-адреса. Однако DHCP можно включить, но для этого нужно зайти в админку маршрутизатора, а для этого нужно сначала

1 Dynamic Host Configuration Protocol - протокол динамической настройки узла, обеспечивает автоматическую настройку всех сетевых параметров.

своему компьютеру назначить статический IP-адрес, зайти в админку, выполнить настройку и только потом уже наслаждаться работой DHCP.

В каждом дистрибутиве Linux есть графические конфигураторы. Поэтому настройку сети мы начнем именно с этих конфигураторов. Список конфигураторов следующий:

- Network Manager (nm-connection-editor) - наиболее часто используемые конфигуратор. Встречается в настольных версиях Fedora (в Fedora Server сеть нужно настраивать вручную), Debian, CentOS, есть даже в OpenSUSE, но о ней отдельный разговор.
- system-config-network - в старых версиях Red Hat, Fedora, ASP Linux. Очень древний конфигуратор, в современных дистрибутивах вы его не встретите.
- network-admin - в старых версиях Debian и Ubuntu.
- netconfig - в Slackware.

В большинстве случаев вам придется иметь дело с одним универсальным конфигуратором - Network Manager, который выглядит примерно одинаково во всех дистрибутивах. Небольшая разница может быть только из-за версии самого Network Manager, но суть везде останется той же

Как правило, кнопка его запуска находится в области уведомлений. Ее внешний вид отличается в зависимости от дистрибутива, например, в Debian это розетка с проводом, в CentOS и OpenSUSE - два компьютера и т.д. Думаю, по внешнему виду, вы догадаетесь, что это именно Network Manager.

На рис. 13.1 показан значок конфигуратора Network Manager и меню, которое открывается при щелчке по нему.

13.2. Установка статического IP-адреса и других параметров сети

Щелкните по его значку правой кнопкой мыши и выберите команду **Изменить соединения** (рис. 13.1). Откроется основное окно Network Manager. Оно примерно одинаковое во всех дистрибутивах (интерфейс немного отличается в зависимости от версии Network Manager, используемой в дистрибутиве). На рис. 13.2 изображено, как выглядит Network Manager в Ubuntu 17.04.

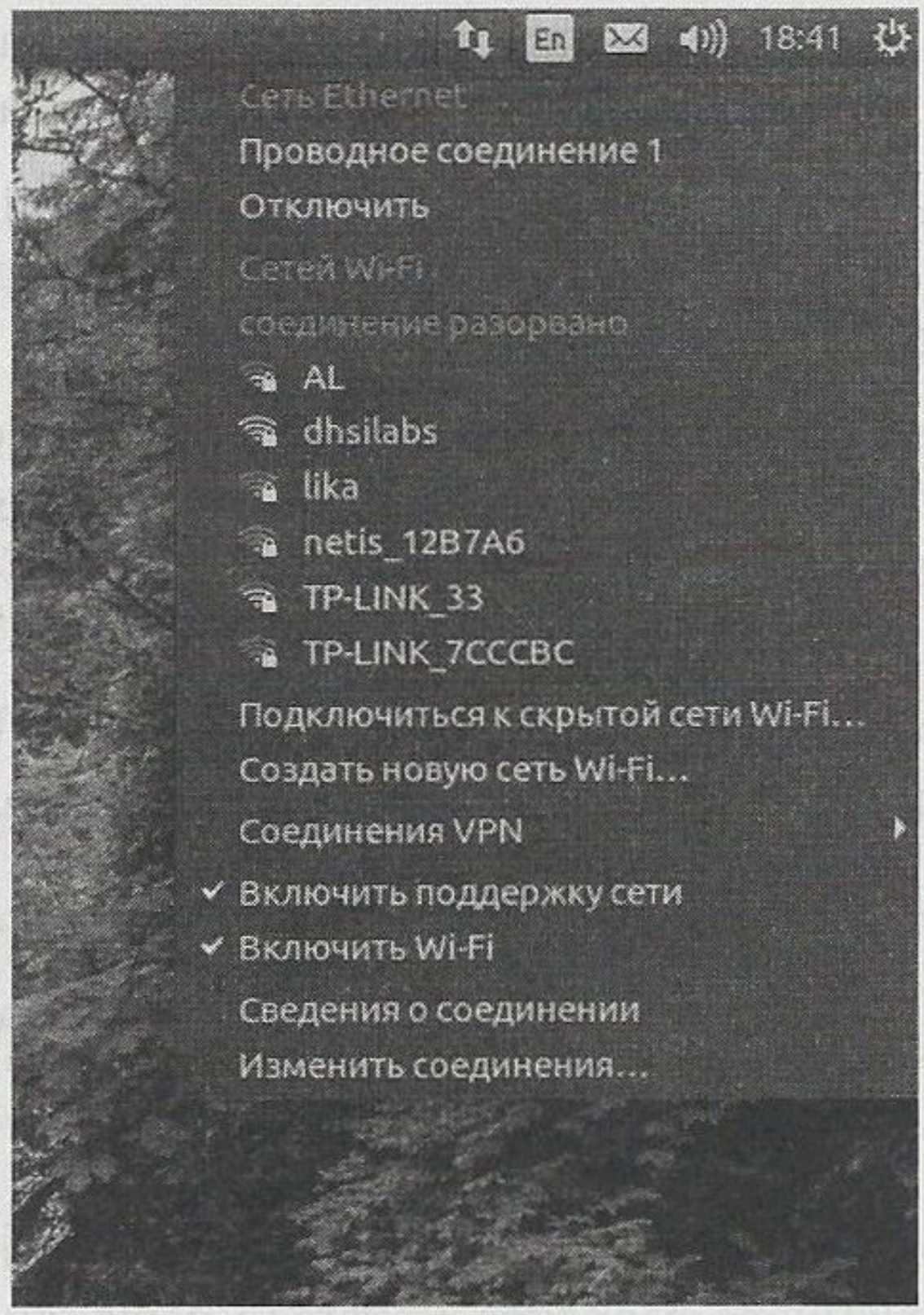


Рис. 13.1. Меню Network Manager

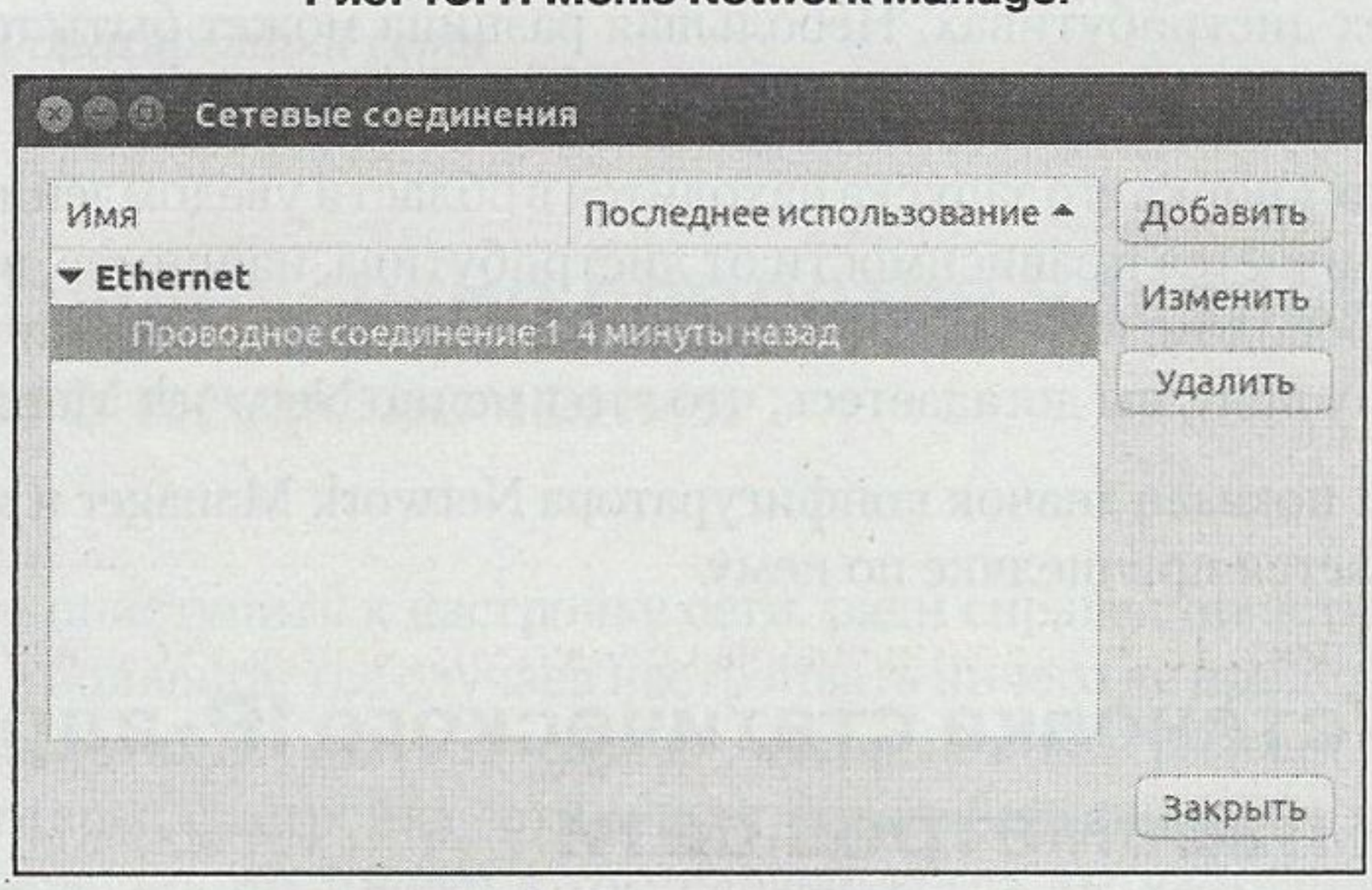


Рис. 13.2. Сетевые соединения

Как уже отмечалось, в большинстве случаев вам не нужно редактировать параметры Ethernet-соединения, поскольку настройка данного соединения осуществляется по протоколу DHCP автоматически. Конечно, в некоторых случаях, например, когда вам нужно по тем или иным причинам клонировать MAC-адрес устройства или у вас нет DHCP-сервера и нужно определить конфигурацию интерфейса вручную, придется редактировать параметры проводного соединения. Также настройка интерфейса вручную может

потребоваться на компьютере, который используется в качестве DHCP-сервере.

Выберите Ethernet-соединение и нажмите кнопку **Изменить** для редактирования его параметров. На вкладке **Ethernet** (рис. 13.3) можно:

- Выбрать физическую сетевую плату, которая будет использоваться для этого соединения (список **Устройство**). Вместо имени сетевого интерфейса в этом списке отображается MAC-адрес устройства.
- Изменить MAC-адрес соединения (**Cloned MAC address**).
- Задать MTU соединения.
- Установить пароль при пробуждении по сети.
- Включить/выключить само пробуждение.

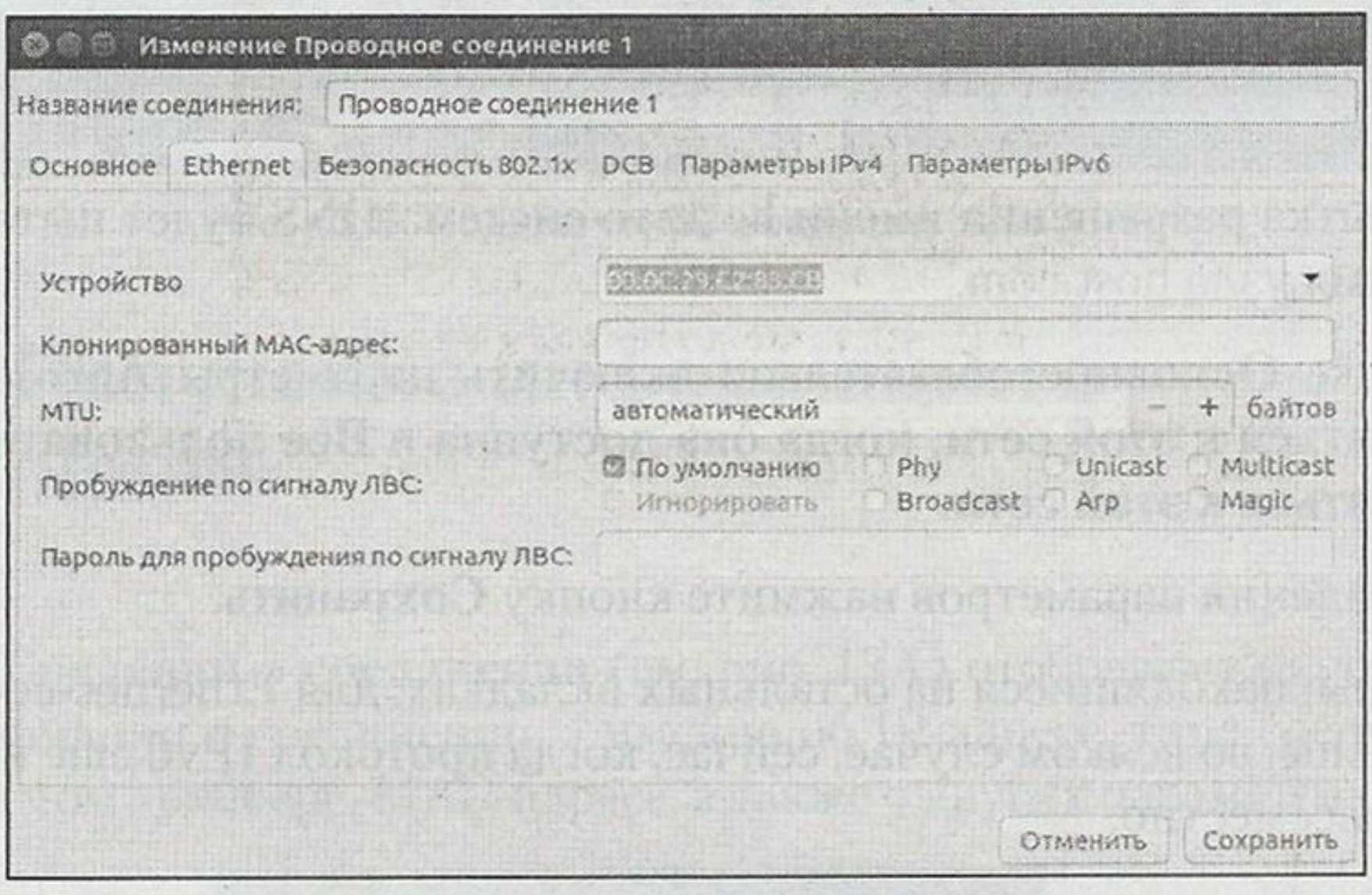


Рис. 13.3. Редактирование соединения. Вкладка Ethernet

Если в вашей сети нет DHCP-сервера, тогда перейдите на вкладку **Параметры IPv4** (рис. 13.4). Из списка **Метод** выберите **Вручную** и нажмите кнопку **Добавить**, чтобы добавить сетевую конфигурацию - IP-адрес, сетевую маску и шлюз по умолчанию.

В поле **Серверы DNS** нужно ввести IP-адреса DNS-серверов. Обычно здесь указываются DNS-серверы предприятия, если они есть, провайдера или можно указать DNS-серверы Google: 8.8.8.8 и 8.8.4.4.

В поле **Поисковый домен** добавляются домены для поиска, если не указано полное доменное имя. Например, если указать здесь .com и будет произве-

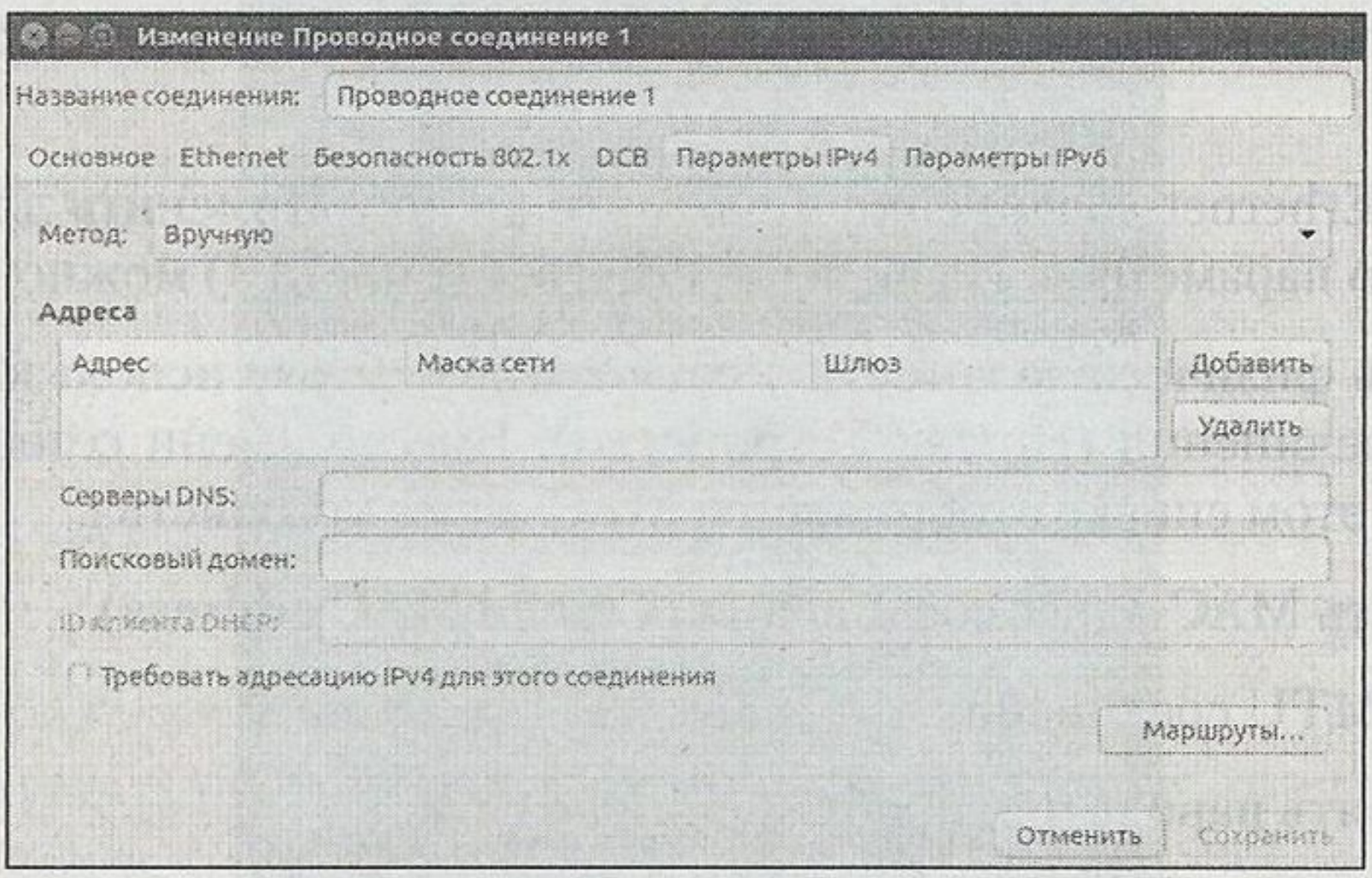


Рис. 13.4. Настройка сети вручную

дена попытка разрешения имени host, то система DNS будет пытаться разрешить имя узла host.com.

На вкладке **Основное** обязательно включить параметры **Автоматически подключаться к этой сети, когда она доступна** и **Все пользователи могут подключаться к этой сети**.

Для сохранения параметров нажмите кнопку **Сохранить**.

Параметры, находящиеся на остальных вкладках, для Ethernet-соединения неактуальны, во всяком случае, сейчас, когда протокол IPv6 еще не используется повсеместно.

13.3. Подключение к Wi-Fi сети

Подключиться к Wi-Fi сети очень просто. Откройте меню Network Manager (рис. 13.1) и просто выберите сеть, к которой вы хотите подключиться. Если Wi-Fi сеть является скрытой, то есть широковещание SSID отключено, выберите команду **Подключиться к скрытой сети Wi-Fi**.

В появившемся окне введите SSID сети, вы должны знать его, выберите тип безопасности и введите пароль (если нужно), см. рис. 13.5.

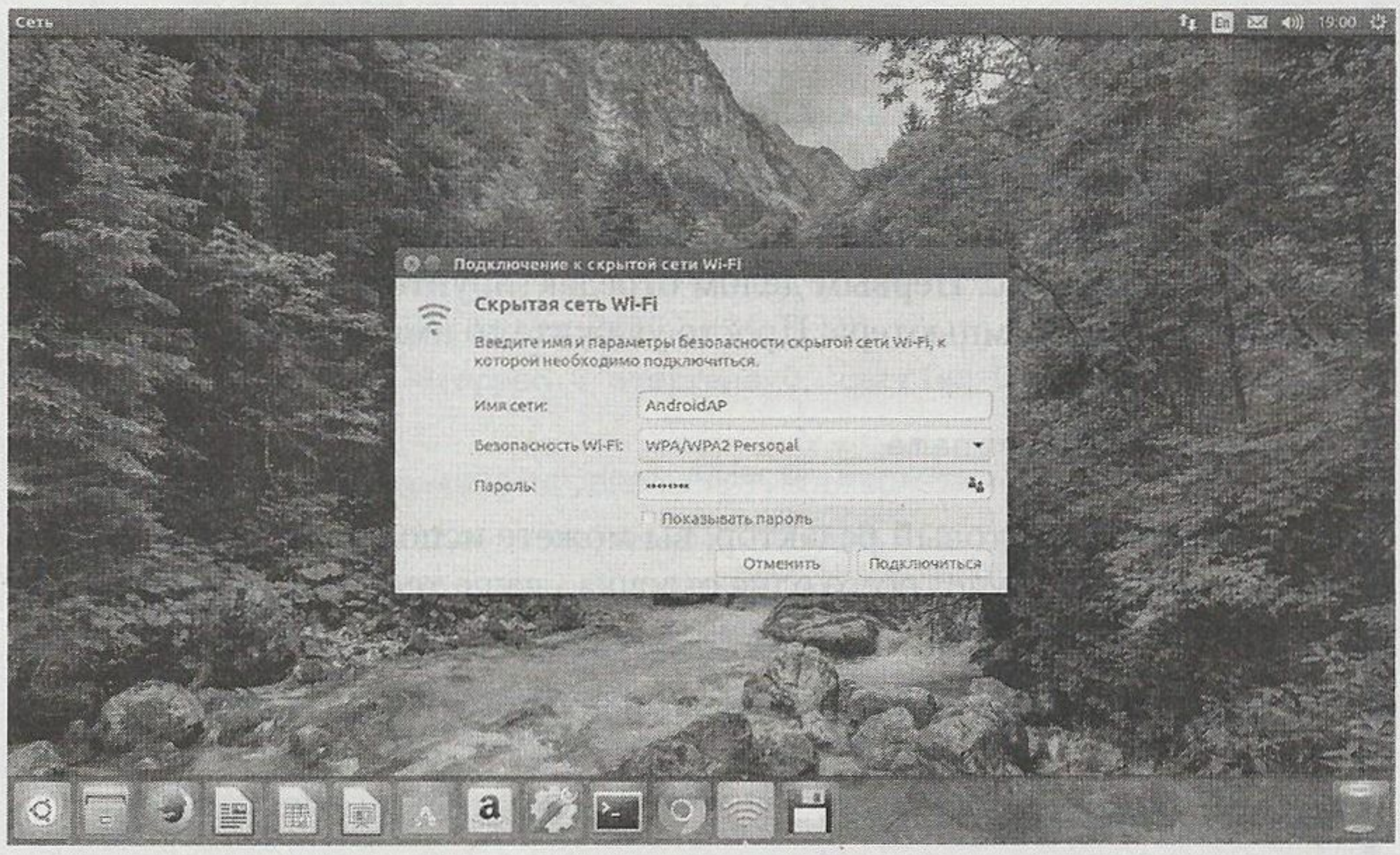


Рис. 13.5. Подключение к скрытой сети Wi-Fi

13.4. Получение информации о соединении

Команда **Сведения о соединении** (см. рис. 13.1) отображает окно с основной информации о соединении, а именно об IP-адресе, типе соединения и используемом драйвере, MAC-адресе, а также о других параметрах сети, см. рис. 13.6.

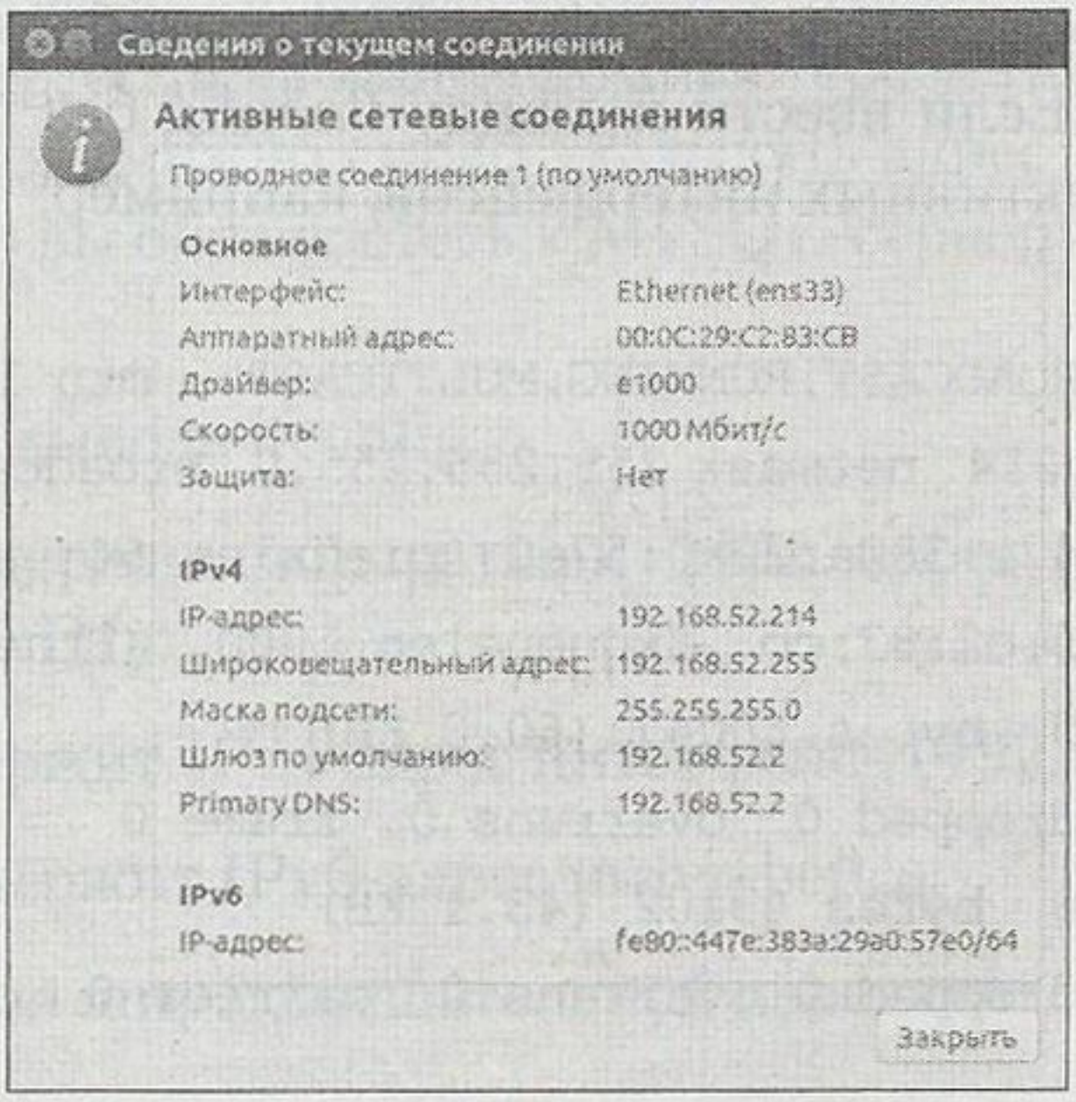


Рис. 13.6. Сведения о соединении

13.5. Изменение имени компьютера

Средствами Network Manager невозможно изменить имя компьютера. Это нужно сделать вручную - с помощью конфигурационных файлов, находящихся в каталоге /etc. Первым делом отредактируйте файл /etc/hostname - он содержит имя компьютера. Просто укажите то имя, которое вам нужно:

```
sudo nano /etc/hostname
```

Здесь nano - это текстовый редактор, вы можете использовать любой, хоть gedit. В этом файле будет всего одна строчка - ваше текущее имя компьютера. Измените его на новое.

Далее нужно изменить файла /etc/hosts и добавить в него записи:

```
127.0.0.1 <имя компьютера>
```

После этого лучше перезапустить компьютер. Можно перезапустить сетевые службы, но могут возникнуть небольшие проблемы, поэтому лучше последовать примеру Microsoft и перезагрузить всю систему. Тем более речь идет о рабочей станции, а не о сервере, где перезагрузка нежелательна.

13.6. Команда ifconfig

Команда ifconfig используется для настройки и отображения параметров сетевого интерфейса. Если ввести команду ifconfig без параметров, то мы получим список всех активных интерфейсов, например:

```
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.52.214  netmask 255.255.255.0  broadcast 192.168.52.255
        inet6 fe80::447e:383a:29a0:57e0  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:c2:83:cb  txqueuelen 1000  (Ethernet)
        RX packets 430  bytes 60047 (60.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 358  bytes 43102 (43.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Локальная петля (Loopback))
        RX packets 338  bytes 24802 (24.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 338  bytes 24802 (24.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlx001f1f831795: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      inet 192.168.52.224  netmask 255.255.255.0  broadcast 192.168.52.255
        ether e6:a8:fb:d9:6d:d1  txqueuelen 1000  (Ethernet)
        RX packets 338  bytes 24802 (24.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 338  bytes 24802 (24.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Примечание. Даже если сетевые интерфейсы не настроены, в выводе ifconfig должен быть интерфейс lo. Если его нет, служба Network Manager не запущена. Для ее запуска введите команду от пользователя root: service NetworkManager start.

Разберемся, что есть что. У нас есть три сетевых интерфейса - ens33 (Ethrenet-адаптер), lo (интерфейс обратной петли), wlx* (Wi-Fi адаптер).

Второй используется для тестирования работы сети и всегда имеет IP-адрес 127.0.0.1. Также он может использоваться для обращения к локальному компьютеру. Например, если на вашем компьютере установлен MySQL-сервер, то в PHP-сценариях можно смело указывать адрес 127.0.0.1 или localhost, чтобы указать, что вы обращаетесь к MySQL-серверу, запущенному на этом компьютере.

Рассмотрим поля вывода ifconfig:

- HWaddr - содержит аппаратный MAC-адрес (в нашем случае это 00:0C:29:C2:0D:D1);
- inet addr - содержит IPv4-адрес интерфейса (192.168.52.214);
- inet6 addr - содержит IPv6-адрес интерфейса;
- Bcast - адрес для широковещательной передачи;
- Mask - маска сети;

- MTU - значение MTU (Maximum transmission unit);
- collisions - счетчик коллизий, если количество коллизий больше 0, с вашей сетью творится что-то неладное. Как правило, в современных коммутируемых сетях коллизии отсутствуют вообще.
- RX packets - количество принятых пакетов;
- TX packets - количество переданных пакетов;
- RX bytes - количество принятых байтов;
- TX bytes - количество переданных байтов.

Получить список всех интерфейсов, а не только активных, можно с помощью параметра -a:

```
# ifconfig -a
```

С помощью команды ifconfig можно деактивировать (down) и активировать (up) любой интерфейс:

```
# ifconfig ens33 down
# ifconfig ens33 up
```

Иногда такой «перезапуск» интерфейса помогает наладить его работу. Особенно это полезно для «подвисших» беспроводных интерфейсов (wlan*).

С помощью ifconfig можно назначить IP-адрес и другие сетевые параметры интерфейса, например:

```
# ifconfig ens33 down
# ifconfig ens33 192.168.1.1 up
```

Сначала мы деактивируем интерфейс, затем назначаем новый IP-адрес и «поднимаем» (параметр up) интерфейс. Можно эту последовательность действий выполнить и за три команды:

```
# ifconfig ens33 down
# ifconfig ens33 192.168.1.1
# ifconfig ens33 up
```

Аналогичным образом можно назначить маску сети и широковещательный адрес:

```
# ifconfig eth0 10.0.0.1 netmask 255.255.255.240 broadcast
10.0.0.15
```

После изменений параметров интерфейса не забудьте его активировать с помощью команды up:

```
# ifconfig eth0 up
```

Изменить MTU интерфейса можно так:

```
# ifconfig eth0 down
# ifconfig eth0 mtu XXXX up
```

Самой интересной возможностью ifconfig является перевод интерфейса в так называемый promiscuous mode. В обычном режиме сетевая карта, если принимает пакет, который был адресован не ей, она его просто отбрасывает. В promiscuous mode карта будет принимать пакеты, даже которые не были ей адресованы. Такая возможность может понадобиться для перехвата трафика:

```
# ifconfig eth0 promisc
```

Перевести карту в обычный режим можно командой:

```
# ifconfig eth0 -promisc
```

Помните, что параметры, заданные с помощью ifconfig, хранятся только до следующей перезагрузки. Если вы хотите сохранить назначенные параметры, нужно или использовать конфигураторы сети (описаны ранее) или редактировать файлы конфигурации (будут описаны далее).

13.7. Сетевые интерфейсы в Linux

В Linux у каждого сетевого интерфейса есть свое имя, чтобы администратор сразу понимал, что это за соединение. Например, у всех PPP-соединений

имя ppp?, где ? - номер соединения: первое соединение называется ppp0, второе - ppp1 и т.д. Нумерация соединений начинается с нуля. В таблице 13.1 приведены названия часто используемых сетевых интерфейсов.

Таблица 13.1. Имена сетевых интерфейсов в Linux

Название интерфейса	Описание
lo	Интерфейс обратной петли
eth, он же ens	Сетевой интерфейс Ethernet
ppp	Соединение PPP (Point-to-Point)
wlx	Беспроводной сетевой интерфейс
tr	Сетевой интерфейс Token Ring
plip	Сетевой интерфейс PLIP (Perallel Line IP). Очень старый интерфейс, вряд ли вы будете с ним иметь дело.
sl	Сетевой интерфейс SLIP (Serial Line IP). Тоже «древний» интерфейс.
fddi	Интерфейс к карте FDDI

Все было бы хорошо, если по непонятным причинам в последних версиях ядра интерфейсы не переименовали. Так, старый добрый eth0 теперь почему-то называется ens33, в некоторых дистрибутивах даже enp3s0. Логики в таких именах не очень много, да и хочется, чтобы имена интерфейсов были такие же, как раньше - как-никак 20-летняя традиция.

На самом деле, логика в новых названиях есть, но только с точки зрения самого компьютера и разработчиков udev. Пользователям и администраторам такая логика не очень понятна. Например, имена, вовлекающие предоставленный BIOS индекс для встроенных в материнскую плату устройств (ID_NET_NAME_ONBOARD) выглядят так - eno1. Имена адаптеров, подключающихся к шине PCI Express, будут содержать номер слота (ID_NET_NAME_SLOT), например, ens1. Имена, вовлекающие физическое/географическое расположение коннектора (ID_NET_NAME_PATH), выглядят еще сложнее - enp2s0. Ну и самые «страшные» - это имена с MAC-адресами

интерфейсов (ID_NET_NAME_MAC). Пример такого имени: enx78e7d1ea46da.

Конечно, можно оставить все как есть - все будет прекрасно работать. Но если вам привычнее обычные имена, тогда есть несколько способов вернуть все, как было раньше.

Первый способ заключается в редактировании файла /etc/udev/rules.d/70-my-network.rules (или 70-persistent-net.rules). Если такого файла у вас нет, значит, его нужно создать. Примерное содержимое файла следующее:

```
# PCI device (r8169)
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="xx:xx:xx:xx:x
x:xx", NAME="eth0"

# USB device (usb)
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="yy:yy:yy:yy:y
y:yy", NAME="wlan0"
```

Здесь xx нужно заменить MAC-адресом вашей сетевой карты, а yy - на MAC-адрес вашего беспроводного адаптера.

Есть и второй способ - передать ядру параметр net.ifnames=0. О том, как это сделать, будет сказано в главе 14.

13.8. Введение в маршрутизацию

13.8.1. Таблица маршрутизации

Отправляемые данные, как вы знаете, не отправляются целиком - они разбиваются на меньшие части - пакеты. Маршрутизация - это процесс перенаправления пакета по различным сетям вплоть до места назначения.

В сетях TCP/IP информация маршрутизации хранится в виде таблицы маршрутизации. Таблица маршрутизации содержит ряд простых правил. Например, если пакет отправляется в сеть 1 он должен быть отправлен на маршрутизатор M1, если пакет отправляется в сеть 2 то его нужно отправить на маршрутизатор M2. Пакет, отправляемый на любую другую сеть (не 1 и не 1), нужно отправить на маршрутизатор M (шлюз по умолчанию). Получив пакеты, маршрутизаторы M1, M1 и M сами знают, что с ними делать.

Возможно, передать дальше другому маршрутизатору, а может отправить узлу-получателя пакета. Все зависит от такой же таблицы маршрутизации, но уже на тех маршрутизаторах. Что будут делать маршрутизаторы с полученными пакетами - это их дело, наше дело - доставить пакеты до этих маршрутизаторов.

Данные маршрутизации хранятся в таблице маршрутизации ядра Linux. Каждая запись в этой таблице содержит несколько параметров - адрес сети назначения, маска сети, флаги, интерфейс и т.д.

Ядро при отправке пакета исследует таблицу машрутизации: оно определяет, в какую сеть направляется пакет - если она есть в таблице маршрутизации, то пакет отправляется через заданный в таблице интерфейс. Если нужной сети в таблице нет, пакет отправляется или на шлюз по умолчанию или же (если он не задан), отправителю пакета передается ICMP-сообщение «Network Unreachable» (сеть недоступна).

Просмотреть таблицу маршрутизации ядра можно или командой netstat -rn или командой route:

```
# netstat -rn
# route
```

Вывод команд немного отличается, как видно из рис. 13.7. Здесь видно, что шлюз по умолчанию - 192.168.52.2. По сути, это самая простая таблица маршрутизации, какая только может быть.

```
ivan@ivan-home: ~
ivan@ivan-home:~$ netstat -rn
Таблица маршутизации ядра протокола IP
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0         192.168.52.2    0.0.0.0         UG        0 0          0 ens33
169.254.0.0     0.0.0.0         255.255.0.0     U         0 0          0 ens33
192.168.52.0    0.0.0.0         255.255.255.0   U         0 0          0 ens33
ivan@ivan-home:~$ route
Таблица маршутизации ядра протокола IP
Destination Gateway Genmask Flags Metric Ref Use Iface
default         gateway         0.0.0.0         UG    100    0        0 ens33
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 ens33
192.168.52.0    0.0.0.0         255.255.255.0   U     100    0        0 ens33
ivan@ivan-home:~$
```

Рис. 13.7. Команды `netstat -rn` и `route`

Разберемся, что содержится в столбцах таблицы маршрутизации. Столбец Destination хранит адрес сети назначения, Gateway - шлюз (маршрутизатор), которому нужно отправить пакеты, чтобы они достигли сеть из колонки Destination.

Столбец Genmask содержит маску сети, а Flags - флаги маршрута. Флаги могут быть следующими:

- U — маршрут активен;
- H — маршрут для хоста (H - host), а не для сети;
- G — флаг шлюза (G- gateway);
- D — динамический маршрут, который был установлен демоном маршрутизации;
- M — маршрут, который был модифицирован демоном маршрутизации;
- C — запись кэширована;
- ! — запрещенный маршрут.

Колонка MSS (Maximum Segment Size) содержит значение MSS - максимальный размер сегмента для TCP-соединений по этому маршруту. Столбец Window показывает размер окна по умолчанию для TCP-соединений по этому маршруту. Столбец irtt — это начальное время RTT. В большинстве сетей время RTT не нужно изменять, но в некоторых медленных сетях время RTT можно увеличить, чтобы избежать лишних повторений пакетов. Система отправляет пакет и ждет некоторое время (RTT) от получателя подтверждения получения. Если подтверждение получения не было, тогда система отправляет пакет еще раз. В медленных сетях подтверждение получения может не успеть дойти до отправителя пакета, поэтому RTT увеличивают (это можно сделать с помощью команды route). Столбец Iface задает интерфейс, используемый для отправки пакета.

В выводе команды route вместо столбцов MSS и Windows есть колонки Metric и Ref. Первая содержит метрику, то есть расстояние до маршрутизатора в хопах (переходах): один хоп - это один маршрутизатор. Столбец Ref содержит - это количество ссылок на маршрут. Ядром Linux этот параметр не учитывается.

13.8.2. Изменение и сохранение таблицы маршрутизации

Для редактирования таблицы маршрутизации используется команда route. Записи в таблице маршрутизации бывают статическими (добавляются командой route) или динамическими (добавляются по мере работы системы демоном маршрутизации).

Вот как можно добавить маршрут по умолчанию командой route:

```
# route add default gw 192.168.1.1 eth0
```

Думаю, эта команда понятна: 192.168.1.1 - это новый шлюз по умолчанию, а пакеты к нему будут отправляться через интерфейс eth0.

При перезагрузке таблица маршрутизации очищается, поэтому она формируется вручную, то есть командами route, а не каким-либо демоном маршрутизации, то эти команды нужно добавить в конфигурационные файлы сети или сценарии инициализации системы, чтобы внесенные вами изменения не были потеряны.

Шлюз по умолчанию можно сохранить в файле /etc/sysconfig/networking-scripts/ifcfg-имя_интерфейса. В листинге 13.1 приводится пример этого файла.

Листинг 13.1. Файл /etc/sysconfig/networking-scripts/ifcfg-имя_интерфейса

```
DEVICE=eth0
HWADDR=00:0C:29:E8:F0:C4
TYPE=Ethernet
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=192.168.1.101
GATEWAY=192.168.1.1
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
BOOTPROTO=none
NM_CONTROLLED=no
```

Параметр GATEWAY позволяет указать IP-адрес шлюза. Но обычно вам не придется редактировать файлы настройки сетевых интерфейсов обычных компьютеров, поскольку такая общая информация как IP-адрес шлюза задается DHCP-сервером примерно так:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    # Список маршрутизаторов (через пробел)
      option routers                  192.168.1.1;
    ...
```

Теперь рассмотрим общий формат вызова команды route:

```
# route [операция] [тип] адресат gw шлюз [метрика] [dev
интерфейс]
```

Параметр операция может принимать значения add и del. Первый добавляет маршрут, а второй - удаляет. Второй параметр (тип) необязательный - он позволяет задать тип маршрута: default (маршрут по умолчанию), -net (маршрут к сети), -host (маршрут к узлу).

Третий параметр, адресат, содержит адрес сети, если вы задаете маршрут к сети, или адрес узла, если добавляется маршрут к узлу). Если вы задаете маршрут по умолчанию, то этот параметр вообще не нужно указывать.

Параметр шлюз задает IP-адрес шлюза. Можно также указать и доменное имя, но обычно указывается IP-адрес. Параметр метрика задает число переходов (маршрутизаторов) на пути к адресату. Параметр dev нужно указывать, если в системе установлено несколько сетевых интерфейсов и нужно указать, через какой именно сетевой интерфейс нужно отправить пакеты. Оба последних параметра не являются обязательными.

Рассмотрим несколько примеров использования команды route:

```
# route add -net 192.168.16.0 netmask 255.255.255.0 dev
ens33
# route add -net 192.168.16.0 netmask 255.255.255.0 gw
192.168.16.1
# route add -net 10.100.0.0 netmask 255.0.0.0 reject
# route del 10.100.0.0
```

Первая команда добавляет маршрут к сети 192.168.16.0 через устройство ens33. Как видите, мы не указываем IP-адрес шлюза, а просто указали имя интерфейса - все пакеты, адресованные сети 192.168.16.0, будут отправлены через интерфейс ens33.

Вторая команда добавляет маршрут к сети 192.168.16.0 через шлюз 192.168.16.1. Все пакеты, адресованные этой сети, будут отправлены маршрутизатору с IP-адресом 192.168.16.1. В этом случае сетевой интерфейс указывать не обязательно.

Третья команда задает запрещающий маршрут. Отправка пакетов в сеть 10.100.0.0 запрещена (параметр reject). Последняя команда удаляет ранее заданный запрещающий маршрут.

Удаление маршрутов в Linux заслуживает отдельного разговора. Команда удаления маршрута выглядит так:

```
# route del IP-адрес
```

В Linux нет параметра -f (как в FreeBSD), который позволяет удалить сразу все маршруты (то есть очистить таблицу маршрутизации), поэтому вам придется ввести ряд команд route.

Как уже было отмечено, таблица маршрутизации очищается при перезагрузке. Чтобы этого не произошло, маршруты нужно определить в файлах конфигурации сети. В случае с Fedora/CentOS маршруты описываются в файле /etc/sysconfig/network-scripts/route-имя_интерфейса (например, /etc/sysconfig/network-scripts/route-eth0).

Вот примеры описания маршрута по умолчанию и двух статических маршрутов:

```
default via 192.168.1.1 dev eth0
10.0.0.0/100 via 192.168.1.1 dev eth0
192.168.2.0/100 via 192.168.1.1 dev eth0
```

В Debian/Ubuntu настройка статических маршрутов производится в файле /etc/network/interfaces. Просто добавьте в секцию настройки интерфейса команду up и укажите команду route, которую нужно выполнить. Допустим, для настройки маршрутов вы ввели три команды:

```
route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
route add -net 192.168.14.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
route add -net 192.168.22.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
```

Тогда в файл /etc/network/interfaces нужно добавить следующие строки:

```
up route add -net 192.168.1.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
up route add -net 192.168.12.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
up route add -net 192.168.21.0 netmask 255.255.255.0 gw 192.168.17.254 eth0
```

В этом примере пакеты ко всем трем сетям направляются на шлюз 192.168.17.254, а он уже сам решает, что с ними делать. Прежде, чем мы

рассмотрим настройку брандмауэра, нужно отметить, как превратить компьютер в шлюз. Для этого нужно разрешить пересылку пакетов протокола IPv4 (IPv4 forwarding). Для этого добавьте значение 1 в файл /proc/sys/net/ipv4/ip_forward:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Конечно, после перезапуска это значение будет потеряно. Чтобы его сохранить, добавьте в файл /etc/sysctl.conf следующую строку:

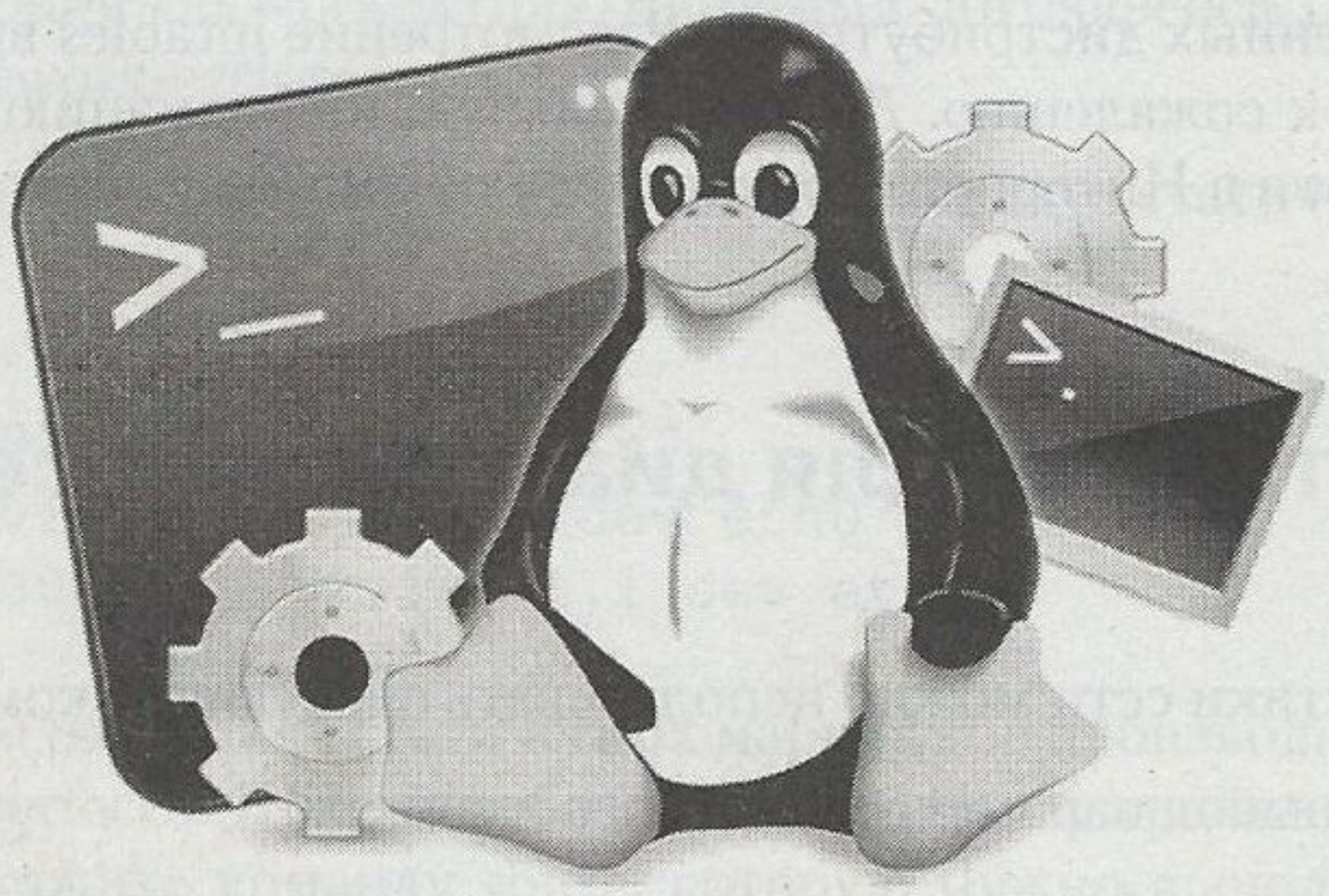
```
net.ipv4.ip_forward=0
```

Итак, ваш компьютер теперь превращен в шлюз. Но сам по себе перенаправлять пакеты он не будет. Нужно задать ряд правил перенаправления пакетов. Для этого нужно использовать брандмауэр iptables, который имеется во всех современных дистрибутивах. Рассмотрение iptables выходит за рамки этой книги, к сожалению. Дополнительную информацию вы сможете без проблем найти в Интернете - руководств по iptables достаточно.

13.9. Утилиты для диагностики сети

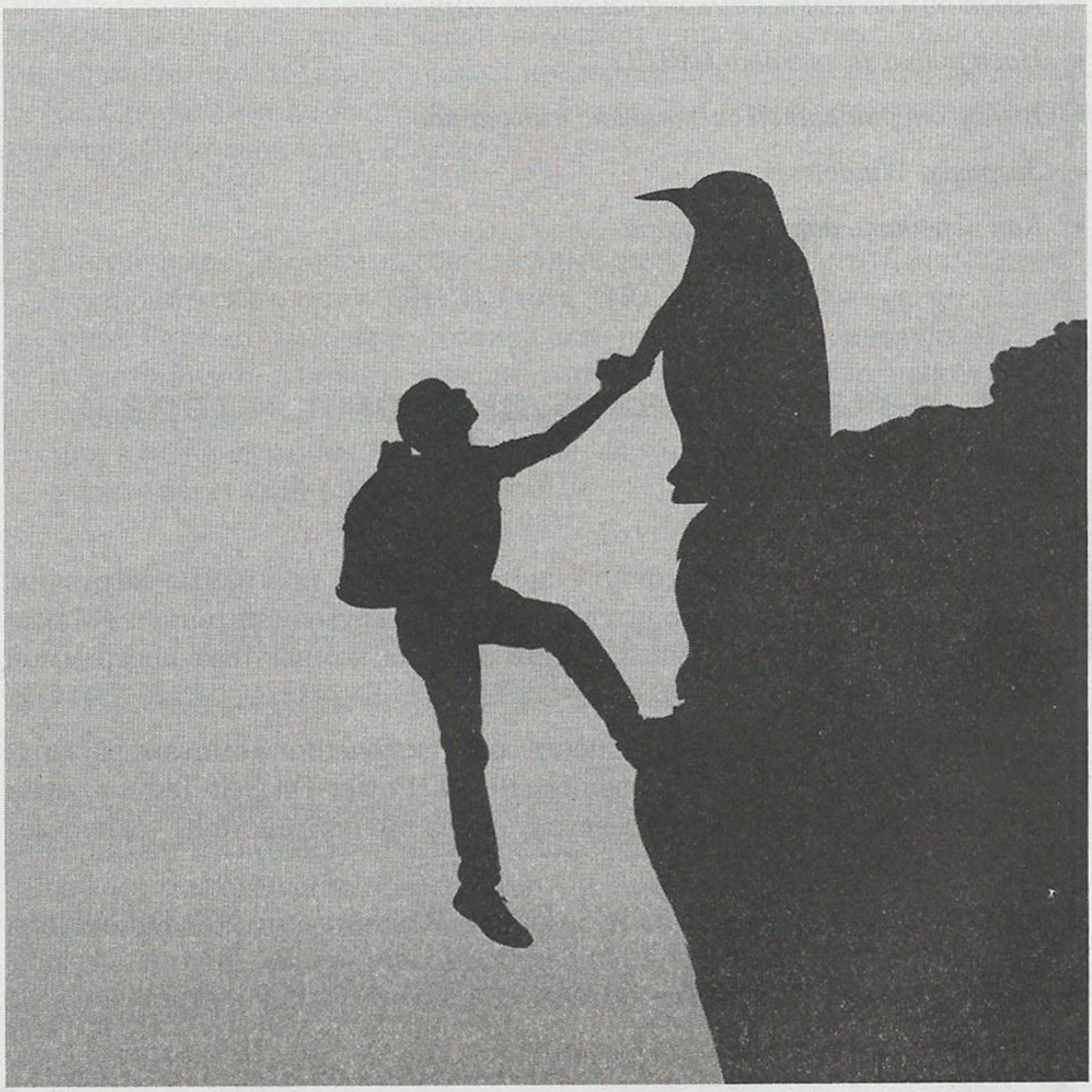
Для диагностики сети можно использовать следующие команды:

- ifconfig - вывод параметров сетевого адаптера.
- route - вывод таблицы маршрутизации.
- ping - пинг узла, то есть проверка доступности узла.
- tracepath - трассировка пути следования пакетов. Если ping выводит только сам факт получения пакета узлом, то tracepath позволяет просмотреть маршрут, по которому проходит пакет до узла.
- traceroute - традиционная версия tracepath, в современных дистрибутивах не устанавливается по умолчанию, нужно установить пакет traceroute.



ГЛАВА 14.

Как загружается Linux?



В этой главе:

- *Процедура загрузки операционной системы*
- *Какие бывают загрузчики*
- *Когда нужно изменять конфигурацию загрузчика*
- *Настройка загрузчика GRUB2*
- *Выбор операционной системы по умолчанию*
- *Загрузка Windows*
- *Как передать параметр ядра*

14.1. Начальная загрузка компьютера и загрузчики Linux

Для запуска любой операционной системы нужна программа-загрузчик. Как правило, загрузчик находится в главной загрузочной записи (Master Boot Record, MBR) и вызывается при запуске компьютера программой BIOS.

Получается такая последовательность: при включении компьютера запускается процедура самотестирования (POST, Power-On Self-Test), а затем BIOS компьютера инициирует загрузку операционной системы - запускает программу-загрузчик.

Программа загрузчик может быть разной. В Windows - это NTLDR, в Linux - как правило, GRUB2 (GRand Unified Bootloader, ver. 2). Другие операционные системы нас сейчас не интересуют, поэтому сконцентрируемся на Linux.

Раз есть GRUB2, то был и просто GRUB. Да, был, но сейчас этот загрузчик считается устаревшим. Лично мне, как системному администратору, больше по душе была первая версия GRUB - он обладал довольно простым и понятным конфигурационным файлом. Но разработчики GRUB значительно расширили его функционал и в связи с этим изменили формат конфигурационных файлов. Теперь редактировать конфигурацию загрузчика стало сложнее. Впрочем, об этом мы еще поговорим.

Вообще-то самым первым загрузчиком Linux был LILO (LInux LOader), но этот загрузчик безнадежно устарел и ему на смену пришел загрузчик GRUB. В свою очередь, загрузчик GRUB был заменен загрузчиком GRUB2.

Далее будет рассмотрен загрузчик GRUB2. В других книгах некоторые авторы рассматривают не только его, но и устаревшие загрузчики (GRUB, LILO). Как по мне, вероятность столкнуться с устаревшим загрузчиком ничтожно мала - разве что на какой-то совсем старой системе. Но поскольку информации по этим загрузчикам предостаточно в Интернете, то, думаю, особых проблем с их настройкой не будет. Тем более, что настраиваются они гораздо проще, чем GRUB2.

Примечание. Во многих современных дистрибутивах есть возможность установить загрузчик GRUB (хотя GRUB2 используется по умолчанию). Особого смысла в этом не вижу, разве что вы привыкли к GRUB, и часто будете изменять конфигурацию загрузки. В противном смысле проще оставить загрузчик по умолчанию, тем более, что вносить изменения в конфигурацию приходится не часто. Если вдруг надумаете, установите пакет grub-legacy.

14.2. Когда нужно изменять конфигурацию загрузчика

Если вы уже купили компьютер с предустановленной Linux или же вы установили Linux на свой компьютер самостоятельно и не планируете устанавливать другие операционные системы, то вам вообще не придется редактировать конфигурацию загрузчика. Но исключения есть всегда.

Давайте рассмотрим типичные ситуации, в которых понадобится редактирование загрузчика:

- Вы установили Linux на компьютер, где уже была установлена Windows и почему-то программа установки не «прописала» Windows в конфигурации загрузчика. Поэтому нужно изменить конфигурацию, чтобы в меню GRUB2 появилась команда запуска Windows.
- Когда вы хотите установить пароль на загрузку операционной системы или/и на изменение конфигурации загрузчика.
- Когда вы хотите добавить в меню загрузчика некоторые дополнительные команды, например, команду загрузки с ISO-образа.

14.3. Установка загрузчика

Загрузчик GRUB2 устанавливается в загрузочную запись при установке операционной системы автоматически. Вам ничего не нужно для этого делать. Но бывают ситуации, например, когда вы в очередной раз переустановили Windows, а ее инсталлятор перезаписал загрузчик.

В этом случае нужно загрузиться с LiveCD (или Live USB - не важно) и переустановить загрузчик. Переустановить загрузчик - это не означает, что нужно переустановить пакет загрузчика. Это означает, что нужно ввести команду установки загрузчика, которая восстановит загрузчик, используя созданную ранее конфигурацию.

Команда установки загрузчика выглядит так:

```
# /sbin/grub-install <устройство>
```

Например:

```
# /sbin/grub-install /dev/sda
```

14.4. Конфигурация загрузчика GRUB2

Конфигурационных файлов у загрузчика GRUB2 несколько. Конфигурация GRUB2 довольно запутанная, поэтому сейчас постараемся в ней разобраться. Я настоятельно рекомендую внимательно читать этот раздел.

В каталоге /etc/grub.d хранятся шаблоны, определяющие настройки GRUB2. Также некоторые его параметры хранятся в файле /etc/default/grub. По шаблонам из /etc/grub.d и файлу /etc/default/grub программой /usr/sbin/grub-mkconfig создается рабочий конфигурационный файл /boot/grub/grub.cfg, который по задумке разработчиков GRUB2 вы не должны редактировать вручную.

Именно поэтому есть две стратегии настройки GRUB2. Первая заключается в непосредственном редактировании файла /boot/grub/grub.cfg. Загрузчику GRUB2 все равно, кто или что отредактирует этот файл - или вы или программа grub-mkconfig. Вторая заключается в редактировании файлов из каталога /etc/grub.d и файла /etc/default/grub. После чего вы будете должны ввести команду grub-mkconfig для создания файла /boot/grub/grub.cfg по заданным вами настройкам.

Второй вариант является более правильным, но вы сами вольны решать, как именно редактировать конфигурацию загрузчика на своем компьютере.

Чтобы решить, какой вариант будет для вас лучше, нужно знать формат и содержимое всех этих файлов. Даже если вы будете использовать вторую стратегию, как наиболее правильную, все равно вам не помешает знать содержимое файла конфигурации grub.cfg.

Рассмотрим фрагмент файла конфигурации /boot/grub/grub.cfg (лист. 14.1). Данный файл взят с системы, где установлена Ubuntu 17.04 (последняя на момент написания этих строк версия Ubuntu). Кроме Ubuntu на этом компьютере больше не установлены какие-либо другие операционные системы. В листинге 14.1 приводится только фрагмент файла grub.cfg, поскольку он довольно длинный и полный его листинг может легко занять 10 страниц книги.

Листинг 14.1. Файл конфигурации /boot/grub/grub.cfg

```
#
# Не редактируйте этот файл!
#
# Он автоматически генерируется программой grub-mkconfig по
шаблонам
# из /etc/grub.d и настройкам из /etc/default/grub
#

### НАЧАЛО файла /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
  set have_grubenv=true
  load_env
```

```
fi
if [ "${next_entry}" ] ; then
   set default="${next_entry}"
   set next_entry=
   save_env next_entry
   set boot_once=true
else
# Загрузочная метка по умолчанию
   set default=»0»
fi

...

### Конец /etc/grub.d/00_header ###

### НАЧАЛО файла /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
set menu_color_highlight=black/light-gray
if background_color 44,0,30,0; then
  clear
fi
### КОНЕЦ /etc/grub.d/05_debian_theme ###

### НАЧАЛО файла /etc/grub.d/10_linux ###
function gfxmode {
   set gfxpayload="${1}"
   if [ "${1}" = "keep" ]; then
      set vt_handoff=vt.handoff=7
   else
      set vt_handoff=
   fi
}
...
export linux_gfx_mode
# Далее описываются элементы загрузочного меню GRUB -
# именно это меню пользователь видит при загрузке
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class
gnu --class os $menuentry_id_option 'gnulinux-simple-0a23abd6-
d721-4def-a56c-c7a7457160ab' {
   recordfail
   load_video
   gfxmode $linux_gfx_mode
   insmod gzio
   if [ x$grub_platform = xxen ]; then insmod xzio; insmod
lzopio; fi
   insmod part_msdos
```

```
   insmod ext2
   set root='hd0,msdos1'
   if [ x$feature_platform_search_hint = xy ]; then
     search --no-floppy --fs-uuid --set=root --hint-
bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1  0a23abd6-d721-4def-a56c-c7a7457160ab
   else
     search --no-floppy --fs-uuid --set=root 0a23abd6-d721-4def-
a56c-c7a7457160ab
   fi
         linux  /boot/vmlinuz-4.10.0-19-generic
root=UUID=0a23abd6-d721-4def-a56c-c7a7457160ab ro  quiet
splash $vt_handoff
   initrd /boot/initrd.img-4.10.0-19-generic
}
submenu 'Дополнительные параметры для Ubuntu' $menuentry_
id_option 'gnulinux-advanced-0a23abd6-d721-4def-a56c-
c7a7457160ab' {
   menuentry 'Ubuntu, с Linux 4.10.0-19-generic' --class ubuntu
--class gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-4.10.0-19-generic-advanced-0a23abd6-d721-4def-a56c-
c7a7457160ab' {
     recordfail
     load_video
     gfxmode $linux_gfx_mode
     insmod gzio
     if [ x$grub_platform = xxen ]; then insmod xzio; insmod
lzopio; fi
     insmod part_msdos
     insmod ext2
     set root='hd0,msdos1'
     if [ x$feature_platform_search_hint = xy ]; then
       search --no-floppy --fs-uuid --set=root --hint-
bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1  0a23abd6-d721-4def-a56c-c7a7457160ab
     else
       search --no-floppy --fs-uuid --set=root 0a23abd6-d721-
4def-a56c-c7a7457160ab
     fi
     echo 'Загружается Linux 4.10.0-19-generic …'
          linux  /boot/vmlinuz-4.10.0-19-generic
root=UUID=0a23abd6-d721-4def-a56c-c7a7457160ab ro  quiet
splash $vt_handoff
     echo 'Загружается начальный виртуальный диск …'
     initrd /boot/initrd.img-4.10.0-19-generic
   }
```

```
  menuentry 'Ubuntu, с Linux 4.10.0-19-generic (upstart)'
--class ubuntu --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-4.10.0-19-generic-init-upstart-
0a23abd6-d721-4def-a56c-c7a7457160ab' {
    recordfail
    load_video
    gfxmode $linux_gfx_mode
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod
lzopio; fi
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
      search --no-floppy --fs-uuid --set=root --hint-
bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1  0a23abd6-d721-4def-a56c-c7a7457160ab
    else
      search --no-floppy --fs-uuid --set=root 0a23abd6-
d721-4def-a56c-c7a7457160ab
    fi
    echo 'Загружается Linux 4.10.0-19-generic …'
        linux /boot/vmlinuz-4.10.0-19-generic
root=UUID=0a23abd6-d721-4def-a56c-c7a7457160ab ro  quiet
splash $vt_handoff init=/sbin/upstart
    echo 'Загружается начальный виртуальный диск …'
    initrd /boot/initrd.img-4.10.0-19-generic
  }
…

### КОНЕЦ /etc/grub.d/10_linux ###
…
```

Файл огромный и его синтаксис напоминает синтаксис bash-сценариев. Если вы таки просмотрели этот конфигурационный файл полностью, то вы уже догадались, что делает программа grub-mkconfig: она собирает воедино все файлы из каталога /etc/grub.d (кстати, в листинге 14.1 перечислена большая часть из этих файлов) и вносит в общий конфигурационный файл из /etc/default/grub.

Но основное, что может интересовать нас в этом файле - это загрузочные метки. Каждая такая метка начинается ключевым словом menuentry, например:

```
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class
gnu --class os $menuentry_id_option 'gnulinux-simple-0a23abd6-
d721-4def-a56c-c7a7457160ab' {
  recordfail
  load_video
  gfxmode $linux_gfx_mode
  insmod gzio
  if [ x$grub_platform = xxen ]; then insmod xzio; insmod
lzopio; fi
  insmod part_msdos
  insmod ext2
  set root='hd0,msdos1'
  if [ x$feature_platform_search_hint = xy ]; then
    search --no-floppy --fs-uuid --set=root --hint-
bios=hd0,msdos1 --hint-efi=hd0,msdos1 --hint-
baremetal=ahci0,msdos1  0a23abd6-d721-4def-a56c-
c7a7457160ab
  else
    search --no-floppy --fs-uuid --set=root 0a23abd6-d721-
4def-a56c-c7a7457160ab
  fi
        linux /boot/vmlinuz-4.10.0-19-generic
root=UUID=0a23abd6-d721-4def-a56c-c7a7457160ab ro  quiet
splash $vt_handoff
  initrd /boot/initrd.img-4.10.0-19-generic
}
```

В одинарных кавычках после menuentry указывается название загрузочной метки. Название метки может быть любым. Обычно название метки соответствует названию дистрибутива, но вы можете указать в качестве названия все, что угодно.

В фигурных скобках - основная конфигурация. Директива load_video - это вызов функции load_video, которая также описана в этом файле конфигурации. Функция вставляет некоторые модули (команда insmod), необходимые для работы графического режима. Обратите внимание, что команды insmod загружают не модули ядра Linux, а модули GRUB2, которые находятся в каталоге /boot/grub.

Внутри {} можно использовать команду echo для обозначения различных этапов загрузки, что и сделано в нашем примере. Можете отказаться от echo, на загрузку это никак не повлияет.

Основные команды - это linux и initrd. Первая указывает путь к ядру Linux и задает параметры ядра. В нашем случае параметр ядра root указывает

устройство, на котором находится корневая файловая система. В данном случае указан UUID устройства. UUID-имена очень удобны, поскольку изменение конфигурации дискового накопителя не приведет к изменению UUID, в отличие от обычного символьного имени. Например, если у вас есть диск /dev/sda и вы случайно подключили его ко второму контроллеру SATA, то его имя изменится и станет /dev/sdb. Если в конфигурации загрузчика жестко прописано имя /dev/sda, то система загружена не будет. Использование UUID исключает эту ситуацию.

При желании, вы можете указать имя в обычном (символьном) формате, например, root=/dev/sda1. Параметр ядра ro задает монтирование корневой файловой системы в режиме «только чтение» (это нормально, позже она будет перемонтирована), initrd - задает файл RamDisk, а последний параметр ядра quiet задает «тихую» загрузку ядра, при которой будут выводиться только самые важные сообщения.

Команда initrd задает путь к файлу initrd. Обычно вам не нужно изменять имя этого файла, если вы не знаете, что делаете.

Прямое редактирование файла /boot/grub/grub.cfg может быть удобно, когда вам нужно внести незначительные изменения в конфигурацию, например, изменить название загрузочной метки или же просто добавить какой-то параметр ядра.

Теперь перейдем ко второй стратегии редактирования конфигурации GRUB2. Она подразумевает редактирование файла /etc/default/grub (листинг 14.2).

Листинг 14.2. Файл /etc/default/grub

```
# После редактирования этого файла запустите команду 'update-
grub' для
# обновления файла /boot/grub/grub.cfg.
# Для получения полной информации об этом файле введите
команду
#   info -f grub -n 'Simple configuration'

# Загрузочный элемент (menuentry по умолчанию)
GRUB_DEFAULT=0
# Таймаут
GRUB_TIMEOUT=5
# Задает название дистрибутива, не изменяйте эту строку
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo
Debian`
# Параметры ядра Linux по умолчанию
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
```

```
# Еще одна строка для задания параметров ядра
GRUB_CMDLINE_LINUX="initrd=/install/gtk/initrd.gz"

# Раскомментируйте эту строку, если вы хотите отключить
графический режим
#GRUB_TERMINAL=console

# Разрешение в графическом режиме
# Вы можете использовать только те режимы, которые ваша
видеокарта
# поддерживает через VBE. Просмотреть список
# таких режимов можно с помощью команды `vbeinfo'
#GRUB_GFXMODE=640x480

# Раскомментируйте эту строку, если вы не хотите
# использовать UUID-имена устройств
#GRUB_DISABLE_LINUX_UUID=true

# Раскомментируйте, если хотите запретить генерирование меток
восстановления
#GRUB_DISABLE_RECOVERY=»true»

# Раскомментируйте, если хотите получить гудок при загрузке
GRUB
#GRUB_INIT_TUNE=»480 440 1»
```

Параметры из файла /etc/default/grub понятны и не нуждаются в особых комментариях. Но в нем упоминается еще одна команда - update-grub. Так какую из них использовать - update-grub или grub-mkconfig?

На самом деле это почти одна и та же команда. Дело в том, что команда grub-mkconfig по умолчанию выводит конфигурацию GRUB2 на экран, поэтому, чтобы она записалась в файл /boot/grub/grub.cfg, запускать ее нужно так:

```
sudo grub-mkconfig > /boot/grub/grub.cfg
```

Или же вы можете ввести команду update-grub, которая сделает то же самое. Другими словами, команда update-grub - это сценарий, который вызывает только что приведенную команду. Получается, что использовать команду update-grub удобнее.

14.5. Выбор метки по умолчанию

Обычно даже если у вас установлена одна только Linux, у вас будет несколько загрузочных меток (несколько записей menuentry). Выбрать метку по умолчанию можно с помощью параметра GRUB_DEFAULT. Нумерация меток начинается с 0, то есть первой метке соответствует значение 0.

После того, как вы установите другой номер метки по умолчанию нужно ввести команду update-grub и перезагрузить систему.

Другими словами, последовательность такая: редактируем файл /etc/default/grub, изменяем значение параметра GRUB_DEFAULT и вводим команду update-grub.

14.6. Добавляем загрузочную метку Windows

Чтобы добавить возможность загрузки Windows из GRUB2, нужно отредактировать файл /etc/grub.d/40_custom и добавить в него следующие строки:

```
menuentry "Windows (/dev/sda1)" {
        insmod part_msdos
        insmod ntfs
        set root='(hd0,msdos1)'
        search --no-floppy --fs-uuid --set UUID
        drivemap -s (hd0) ${root}
        chainloader +1
}
```

Здесь Windows находится на /dev/sda1 и адрес этого раздела указывается в set root. Значение UUID нужно заменить на UUID вашего Windows-раздела. О том, как «вычислить» UUID раздела, было сказано в главе 4. Остальные параметры можете оставить без изменения (еще не забудьте исправить hd0 в drivemap, если Windows установлена не на первом жестком диске).

14.7. Загрузка с ISO-образа

Загрузчик GRUB2 поддерживает одну очень интересную особенность - загрузки с ISO-образов, что может быть полезно, если вы хотите загрузиться с LiveCD, но у вас нет желания портить «болванку».

Далее приведен пример загрузочной записи для загрузки с ISO-образа:

```
menuentry "ISO" {
    loopback loop /boot/my.iso
    linux (loop)/casper/vmlinuz boot=casper iso-scan/
filename=/boot/my.iso noeject noprompt --
    initrd (loop)/casper/initrd.lz
  }
```

В данном случае загрузка производится с ISO-образа my.iso, который помещен в каталог /boot. Можете даже не изменять загрузочный элемент, а просто переименовать ISO-файл в my.iso и поместить его в каталог /boot.

14.8. Пароли и паранойя

Пользователь может установить множество паролей, чем усложнить злоумышленнику доступ к своему компьютеру. Пароли могут быть разными:

- Пароль на вход в BIOS
- Пароль BIOS на загрузку операционной системы
- Пароль загрузчика GRUB2
- Пароль пользователя при входе в систему

Эффективность у всех этих паролей разная. Пароль на вход в BIOS легко устраняется путем извлечения батарейки с материнской платы. Все параметры BIOS будут сброшены, в том числе и пароль BIOS. Эффективность у этого пароля довольно низкая, но в случае с ноутбуками этот пароль более эффективен, чем в случае со стационарными компьютерами. Вы пробовали разбирать ноутбук? Конечно, если злоумышленник украл ноутбук и у него есть куча свободного времени, то произвести все необходимые манипуляции - не проблема. А вот когда речь идет о стационарном компьютере, то

снять заднюю крышку (при отсутствии пломбы) и извлечь батарейку можно за одну минуту. Затем нужно немного подождать, пока энергозависимая память BIOS будет стерта (до 5 минут), хотя можно просто установить джампер сброса параметров (технически подкованный злоумышленник наверняка о нем знает). На все про все - 1 минута. Поэтому, сами понимаете, толку от этого пароля немного. Остановит только неподготовленного пользователя.

Теперь о пароле загрузчика. Как можно сбросить пароль root? Достаточно передать ядру параметры «init=/bin/bash rw», а после загрузки системы ввести команду passwd root или passwd <имя_административного_пользователя>. Вся проблема в том, что загрузчик GRUB2 позволяет редактировать свою конфигурацию по умолчанию - прямо в меню загрузчика вы можете выбрать метку и нажать E для ее редактирования. Все, что вам останется - это просто добавить параметры ядра.

Если установить пароль на редактирование параметров GRUB2, то у злоумышленника не будет такой возможности. Да, он не сможет в обеденный перерыв перезагрузить ваш компьютер, сменить пароль и заполучить полный доступ к вашей системе. Но если у него будет достаточно времени, для него не составит труда загрузиться с LiveCD/LiveUSB и подмонтировать корневую файловую систему, выполнить команду chroot и затем просто изменить пароль. Легче простого! Подробную процедуру описывать не буду - не хочется никого учить плохому, но о такой возможности вы должны знать, если беспокоитесь о безопасности ваших данных.

Если вы - агент 007, то единственный выход из сложившейся ситуации (необходимость надежной защиты данных) - это шифрование. Вы можете или зашифровать всю файловую систему или зашифровать только домашний каталог. Для более надежной защиты, понятное дело, нужно шифровать всю файловую систему. Обычно опция шифрования файловой системы есть при установке Linux.

Однако у шифрования всей корневой файловой системы есть два недостатка. Первый - снижается производительность. Но, учитывая, что современные компьютеры довольно мощные и используют быстрые SSD-диски, думаю, это не проблема. Второй - шифрование всей файловой системы нельзя использовать на серверах, поскольку при загрузке системы нужно будет ввести пароль (ключ). Если сервер придется перезагрузить, то нужен будет физический доступ к нему. Если сервер находится в офисе - проблем нет, а вот если он в дата-центре или вообще вы арендуете VDS, то придется связываться со службой поддержки и... сообщать третьим лицам свой пароль. Ситуация малоприятная.

Все это - теория. А сейчас перейдем к практике. Загрузчик GRUB позволял только установить пароль - или общий или на загрузку определенной метки. Загрузчик GRUB2 более гибкий в этом плане, поскольку вы можете настроить не только пароли, но и логины. Также есть минимальная система разграничения прав доступа.

В GRUB2 существует суперпользователь, который имеет право редактировать загрузочные метки. Ранее упоминалось, как можно легко восстановить пароль root путем передачи параметра ядра init. Но для этого нужно отредактировать конфигурацию GRUB2. Если вы установите пароль суперпользователя, то изменить конфигурацию загрузчика вы сможете только после ввода этого пароля.

Также в GRUB2 есть обычные пользователи, которые имеют право только выбирать загрузочную метку. Они не имеют права редактировать конфигурацию загрузчика. В принципе, можно обойтись одним паролем суперпользователя, но при желании GRUB2 может довольно гибко разграничить права пользователей.

Давайте сначала добавим пароль суперпользователя. Для этого в файл /etc/grub.d/00_header добавьте строки:

```
set superusers="dhsilabs"
password dhsilabs 007secret007
```

Первая команда задает суперпользователя, а вторая - задает для него пароль. Старайтесь избегать общепринятых имен вроде admin, root и т.д. Так у злоумышленника, который хочет изменить конфигурацию GRUB2 будет две неизвестных.

Обычные пользователи задаются инструкцией password, например:

```
password ivan q1w2e3r4t5y6
```

Пользователь ivan может только загружать операционные системы.

Параметр --users, добавленный к menuentry, позволяет задать пользователей, которые имеют право выбирать загрузочную запись:

```
menuentry «Test» --users ivan {
...
}
```

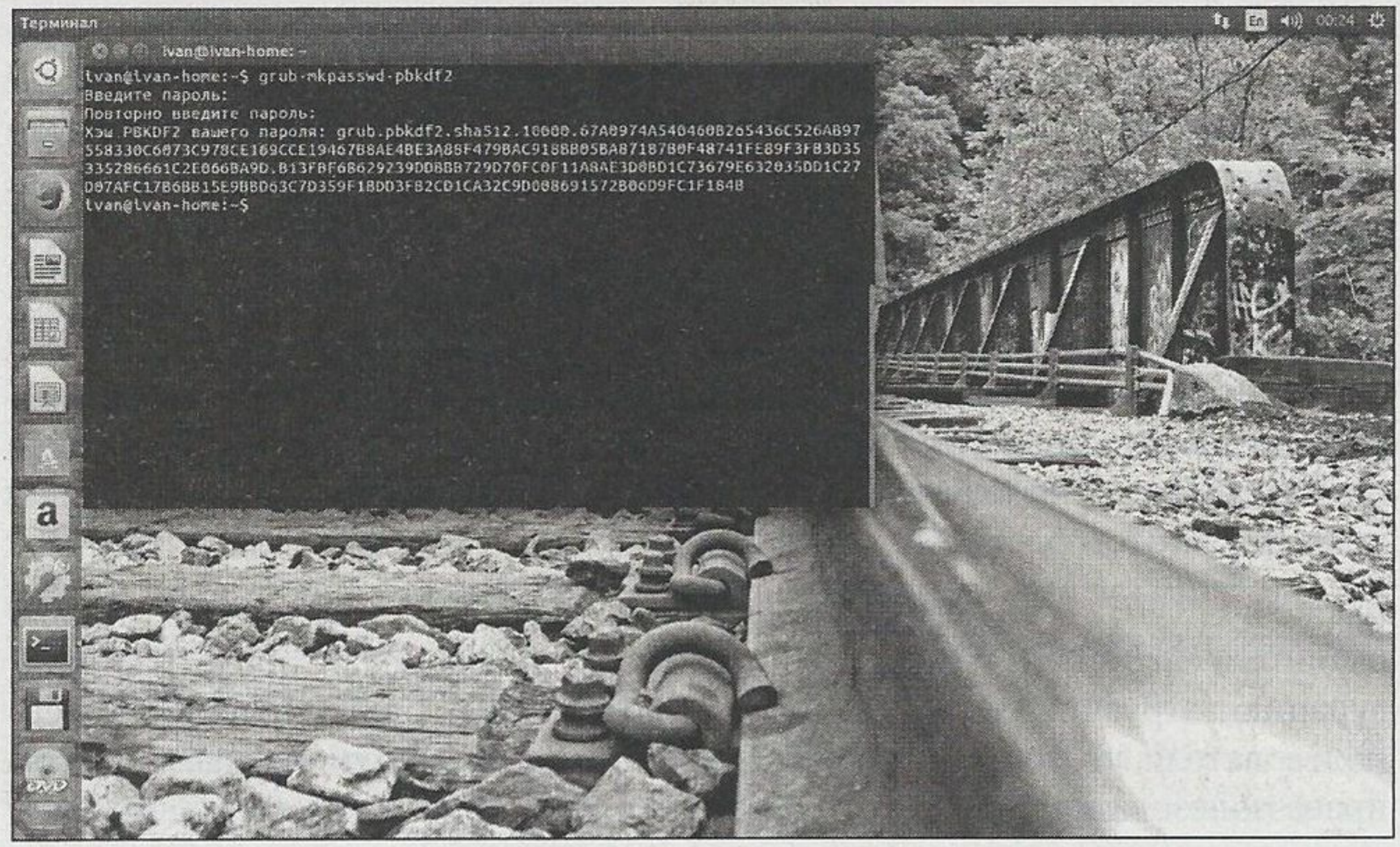


Рис. 14.1. Программа grub -mkpasswd-pbkdf2 в действии

Разберемся, как зашифровать пароль. Для этого используется программа grub-mkpasswd-pbkdf2. Введите эту команду (рис. 14.1), она попросит ввести ваш пароль, а затем сообщит его хэш. Так вот этот хэш и нужно указать в файле конфигурации, например:

```
password_pbkdf2 ivan grub.pbkdf2.sha512.10000……
```

На этом, собственно все. Следующая глава будет не менее интересной - в ней мы рассмотрим систему инициализации systemd, которая используется во всех современных дистрибутивах.

14.9. Как передать параметр ядра

В прошлой главе я обещал рассказать, как передать параметр ядра net.ifnames. На рис. 14.2 показано начальное меню загрузчика GRUB 2 (дистрибутив Fedora 26). Выберите метку ядра и нажмите e.

Затем найдите строку linux и добавьте в нее нужный параметр, в нашем случае net.ifnames=0 (рис 14.3).

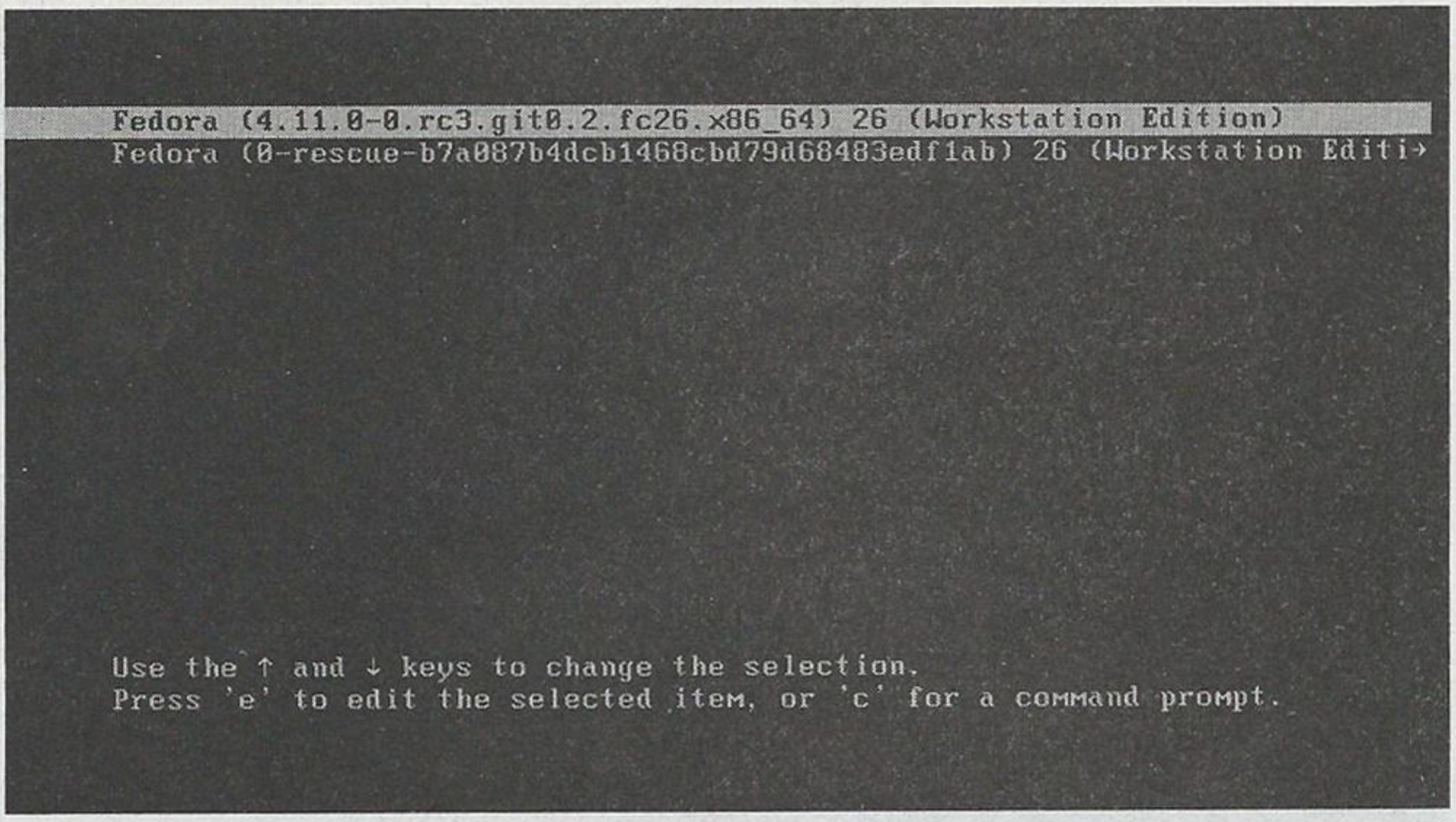


Рис. 14.2. Начальное меню GRUB2

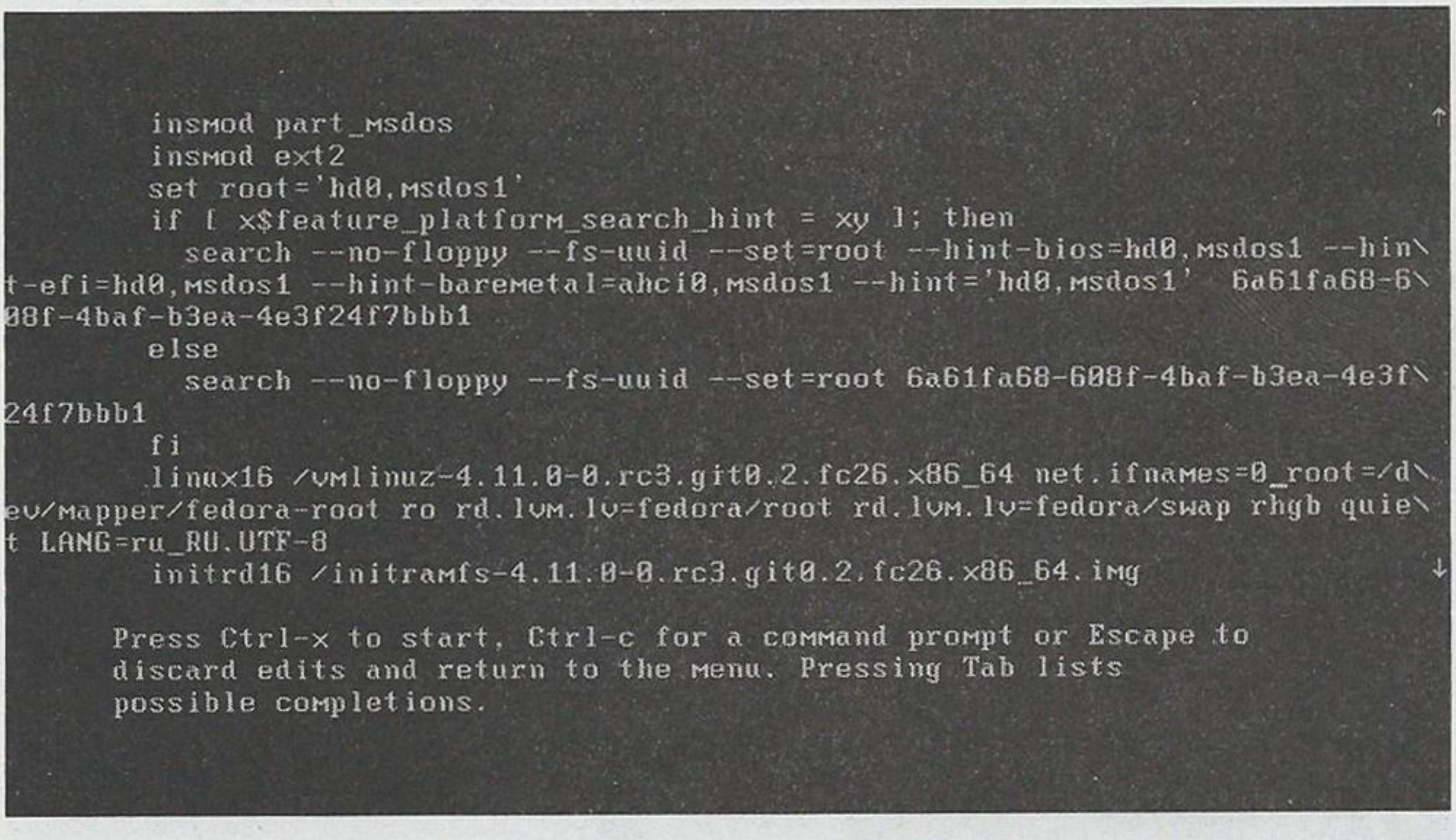


Рис. 14.3. Редактирование параметров ядра

Для продолжения загрузки нажмите Ctrl + X. После загрузки проверьте эффект от использования параметра ядра. В нашем случае вместо странных имен сетевых интерфейсов вида ens33 теперь используются классические имена (ethX), см. рис. 14.4.

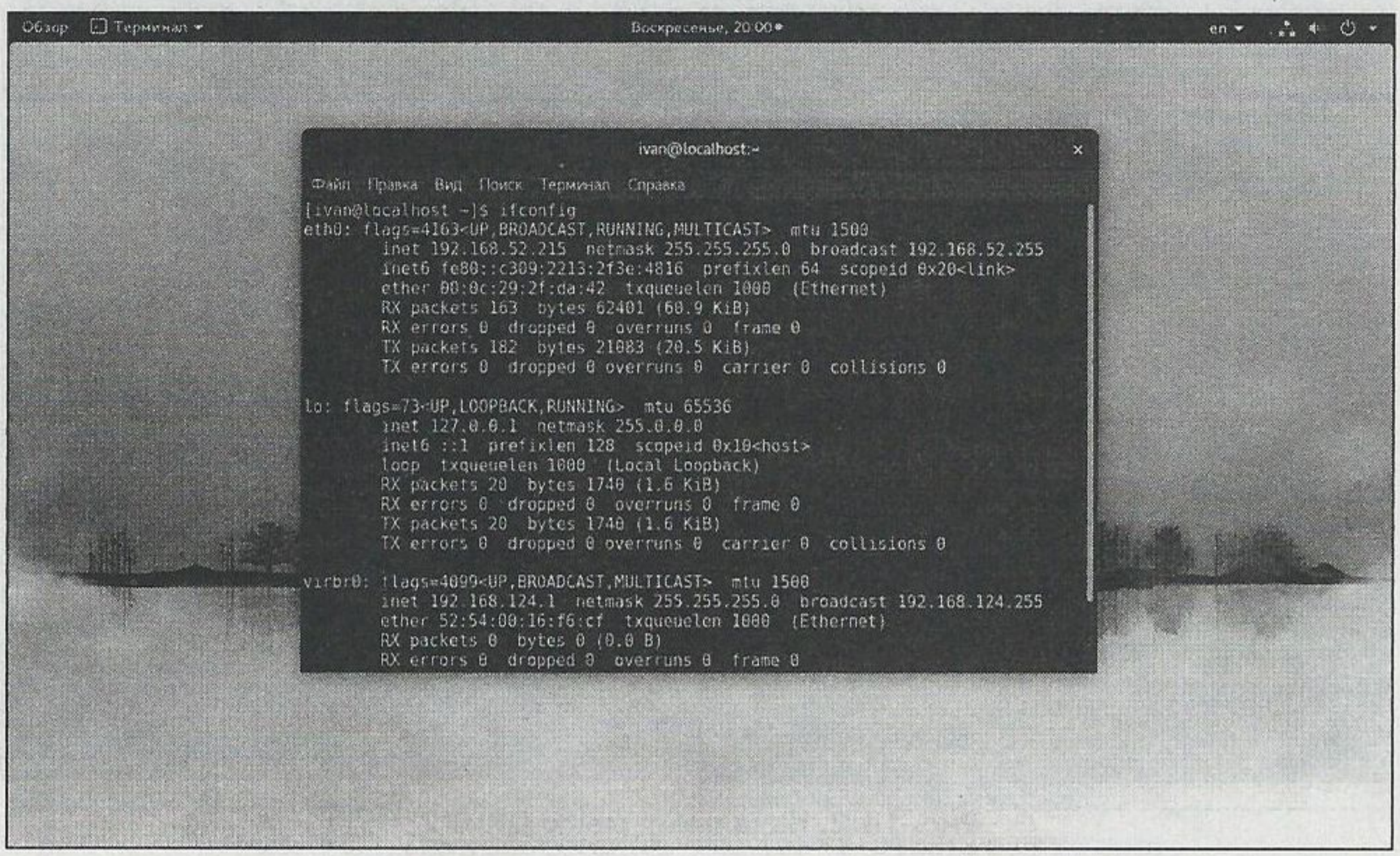
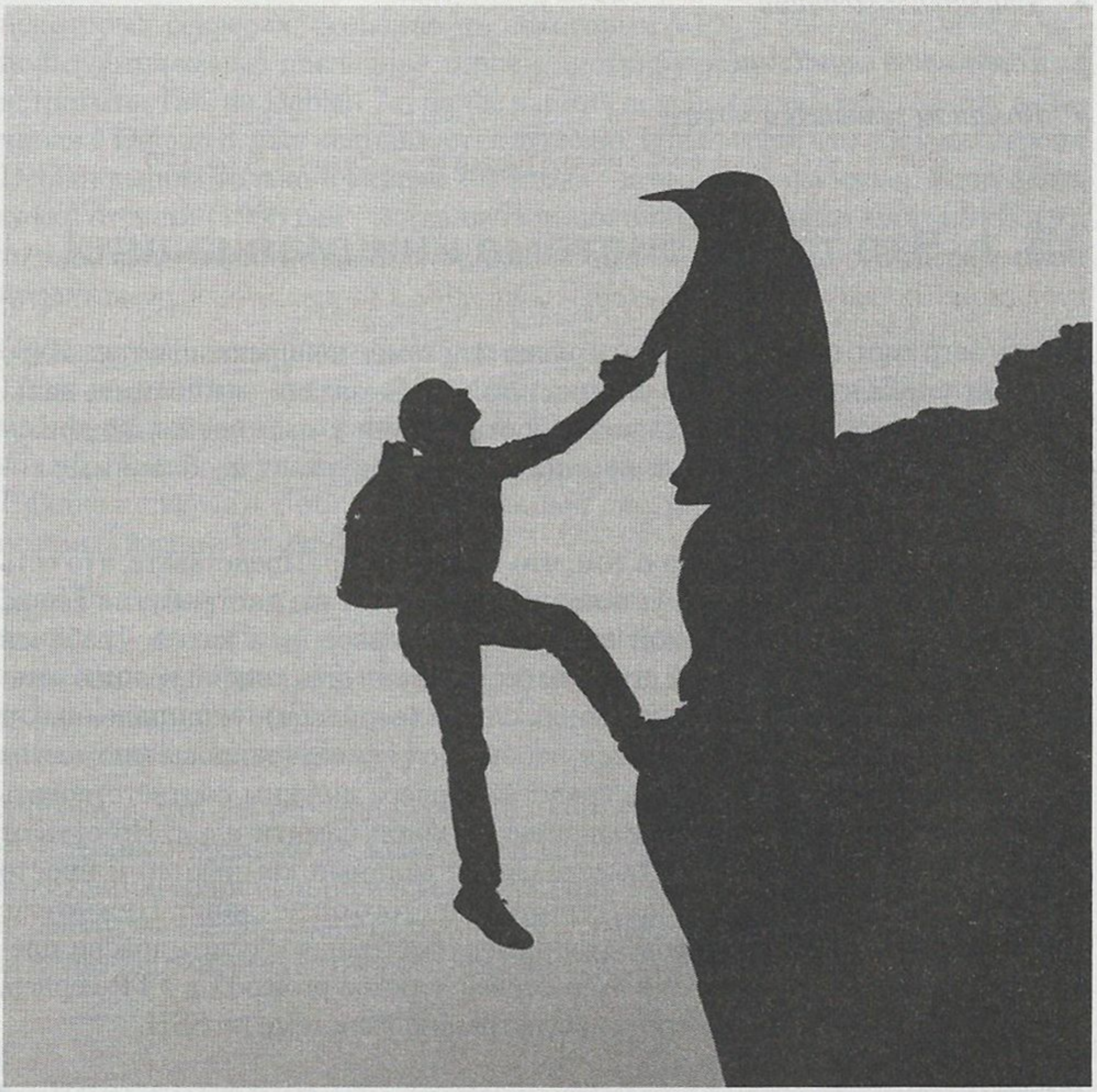


Рис. 14.4. Классическое имя сетевого интерфейса в Fedora 26

Если эффект вас устроил, самое время «прописать» этот параметр в конфигурационных файлах GRUB, чтобы не вводить при каждой загрузке системы.

ГЛАВА 15.

Системы инициализации



В этой главе:

- *Системы инициализации init и systemd*
- *Запуск служб*
- *Утилита systemadm*
- *Управление службами*
- *Команды systemctl и service*

15.1. Что такое система инициализации

После загрузки ядра процесс загрузки системы не прекращается. Ядро передает управление системе инициализации. Ее задача - выполнить дальнейшую инициализацию системы в соответствии с имеющейся конфигурацией. Как правило, система инициализации запускает необходимые системные службы.

Во второй главе мы говорили о том, что такое сервер. Представьте, что есть два компьютера, на которых установлен один и тот же дистрибутив Linux, даже версии совпадают. Какой из них будет сервером, а какой - рабочей станцией? Сервер от рабочей станции (если не считать конфигурацию «железа») отличается набором системных служб (сервисов), установленных и запущенных на нем. Понятное дело, чтобы обслуживать запросы множества рабочих станций, и аппаратные средства сервера должны соответствовать - более мощный процессор, больше оперативной памяти и т.д. Но суть от этого не меняется: если вы возьмете самый мощный компьютер и просто установите на него ту же Ubuntu, сервером от этого он не станет. Нужно еще установить системные службы. Например, системная служба apache превращает обычный компьютер в Web-сервер, служба proftpd - в FTP-сервер, а сервис ssh позволяет удаленную регистрацию в системе по SSH.

Чтобы администратор не запускал все эти службы вручную при каждом запуске компьютера, используется система инициализации. Конечно, она не только запускает службы, а еще и управляет процессом запуска. Ведь важно не просто запустить службу, а запустить ее в нужном порядке. Бывает так, что служба А зависит от службы Б и пока служба Б не сообщит о нормальной работе, службу А запускать нельзя. Это называется зависимостью служб.

Еще когда не было Linux, произошло разделение UNIX-подобных операционных систем на два типа: те, которые использовали традиционный стиль инициализации BSD и те, которые использовали систему SysV init. Так вот, та самая система инициализации init прожила во многих дистрибутивах Linux очень и очень долго.

Первая и самая древняя система инициализации init можно сказать практически дожила до наших дней - вплоть до 2015 года. Интересно, что на некоторых серверах, особенно на некоторых VPS/VDS, где не всегда используются самые последние версии дистрибутивов Linux, ее еще можно встретить. Так, на Debian 7 и более ранних версиях используется init, а, начиная с Debian 8, уже используется systemd. Интересно, что восьмая версия Debian вышла только в апреле 2017 года - всего два года назад. Если взять точку отсчета (1993 год - когда начали появляться первые дистрибутивы), то выходит, что init прожила 22 года. Это очень долго для программного продукта.

Система init была настолько удачной, что разработчики дистрибутивов Linux не спешили от нее отказываться, хотя попытки были. Здесь и система init-ng, которая ускоряла запуск системы за счет параллельного запуска служб. Нельзя не упомянуть и систему upstart, разработанную для Ubuntu. Начиная с версии Ubuntu 15.04, данный дистрибутив перешел на систему инициализации systemd.

Хорошая новость в том, что все современные дистрибутивы используют одну и ту же систему инициализации - systemd. Раньше была путаница - в зависимости от дистрибутива и от его версии, пользователь мог столкнуться с разными системами - init, upstart, systemd (init-ng прожила совсем недолго - оказалась неудачной).

Учитывая, что вы все еще можете столкнуться с системой init на некоторых компьютерах, мы ее таки рассмотрим. А после этого сразу приступим к изучению современной системы systemd. Учитывая, то все дистрибутивы перешли на нее, это говорит о многом - думаю, в ближайшие 5-10 лет будет использоваться именно эта система инициализации.

15.2. Классическая система инициализации SysV init

15.2.1. Файл конфигурации inittab

Принцип действия init следующий:

- При запуске init читает файл конфигурации /etc/inittab с целью определения уровня запуска по умолчанию и других параметров
- Запускается сценарий /etc/init.d/rcS - это основной сценарий инициализации/конфигурации системы
- Затем init, в зависимости от уровня запуска, запускает сценарии из каталога /etc/rcN.d, где N - уровень инициализации. Например, если выбран 5-ий уровень инициализации, то будут запущены сценарии из каталога /etc/rc5.d. Традиционно считается, что 5-ый уровень запуска - это многопользовательский режим с графическим интерфейсом, а 3-ий уровень - это многопользовательский режим без графики. Хотя реально то, что будет запущено, зависит от содержимого /etc/rcN.d.
- При изменении уровня запуска выполняются сценарии из каталога, который соответствует уровню запуска.

По сути, в каталогах /etc/rcN.d находятся не сами сценарии, а ссылки на сценарии. Сами же сценарии находятся в каталоге /etc/init.d. Это сделано умышленно: ведь на разных уровнях могут запускаться одни и те же службы. Если вам придется отредактировать сценарий запуска службы, то проще это сделать один раз - в каталоге /etc/init.d, чем в нескольких каталогах /etc/rcN.d.

Основным файлом конфигурации init является файл /etc/inittab. В листинге 15.1 приводится файл конфигурации /etc/inittab из дистрибутива Debian 7 - последнего современного дистрибутива, где используется init.

Листинг 15.1. Файл /etc/inittab

```
# /etc/inittab: init(8) configuration.
# $Id: inittab,v 1.91 2002/01/25 13:35:21 miquels Exp $

# Уровень запуска по умолчанию
```

```
id:3:initdefault:

# Сценарий инициализации/конфигурации системы времени загрузки
# Он запускается в первую очередь, за исключением загрузки
# в экстренном режиме (-b)
si::sysinit:/etc/init.d/rcS

# Что будем делать в однопользовательском режиме (single).
# Если удалить эту строчку, то после передачи ядру параметра
single
# пользователь получит доступ в консоль с правами root. Сами
понимаете:
# кто угодно сможет получить максимальные права, если укажет
параметр
# single! Используемая конфигурация предполагает вызов
sulogin,
# которая запросит и проверит пароль root и в случае его
правильности
# предоставит доступ к консоли
~~:S:wait:/sbin/sulogin

# Init выполняет S и K сценарии при изменении уровня запуска
#
# Runlevel 0 - останов системы.
# Runlevel 1 - однопользовательский режим.
# Runlevels 2-5 - многопользовательские режимы.
# Runlevel 6 - перезагрузка.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
# Обычно этот уровень не достигается, но в него система может
перейти
# в экстренном режиме
z6:6:respawn:/sbin/sulogin

# Что делать при нажатии CTRL-ALT-DEL.
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Действия при нажатии специальных комбинаций клавиш (ALT-
UpArrow).
```

```
#kb::kbrequest:/bin/echo "Keyboard Request--edit /etc/inittab
to let this work."

# Что делать, если получены сигналы сбоя/восстановления
питания
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# Запуск getty
# Вы будете редактировать эти строки очень редко
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

Первым делом разберемся с уровнями запуска:

- 0 - останов системы;
- 1 - однопользовательский режим;
- 6 - перезагрузка системы.

Уровни 2-5 я специально пропустил. Дело в том, что раньше уровень 2 считался многопользовательским режимом без поддержки сети, 3 - многопользовательским режимом с поддержкой сети, но без запуска графического интерфейса, 4 вообще не использовался, а 5 - это многопользовательский режим с поддержкой сети и запуском графического интерфейса. Теперь уровни 2, 4 и 5 - это одно и то же, в чем вы можете убедиться по содержимому каталогов /etc/rc2.d, /etc/rc4.d и /etc/rc5.d.

Уровень 3 - такой же, как и раньше, то есть - многопользовательский режим с поддержкой сети и без запуска графического интерфейса.

Изменить уровень запуска можно так:

```
# /sbin/init <уровень>
```

Если сейчас ваша система загружается в графическом режиме (уровни 2, 4 и 5), вы можете перейти в обычный консольный режим:

```
# /sbin/init 3
```

По сути, вместо команды reboot вы можете ввести команду (если в вашем дистрибутиве используется init):

```
# /sbin/init 6
```

Следующий параметр задает сценарий, который будет выполнен в первую очередь при запуске init:

```
si::sysinit:/etc/init.d/rcS
```

Вы можете открыть этот файл - в нем будет передача управления сценарию /etc/init.d/rc (в качестве параметра ему передается номер уровня запуска), в котором содержатся инструкции по запуску/останову служб из каталогов /etc/rcN.d. В принципе, ничего интересного и вам не придется редактировать этот сценарий. Разве что в исключительных случаях.

Следующий параметр задает, какая программа будет запущена в однопользовательском режиме, в который можно перейти или путем передачи ядру параметра single или путем ввода команды init 1:

```
~~:S:wait:/sbin/sulogin
```

Команда sulogin запросит пароль пользователя root. А это означает, то если кто-то передаст ядру при загрузке параметр ядра single, то при загрузке в однопользовательском режиме система запросит пароль root, а не просто передаст кому-то права root.

Вообще-то данная защита - не панацея, но лучше, чем вместо sulogin вы будете запускать /bin/bash - тогда злоумышленник, который введет параметр single, получит доступ к оболочке bash с правами root вовсе без ввода пароля.

Но не нужно считать, что защита с помощью init сработает. Ведь ничто не мешает злоумышленнику передать параметр ядра init=/bin/bash и тогда вместо init будет запущена оболочка bash с правами root. Чтобы обезопаситься от этого вида угрозы, нужно запретить редактировать конфигурацию загрузчика GRUB2. Как это сделать, было показано в главе 14. Но защита средствами GRUB2 тоже не панацея. Почему? Потому что злоумышленник может загрузиться с LiveCD, изменить корневую файловую систему с помощью chroot и ввести команду passwd для изменения пароля root. Защитить систему от такого способа взлома можно с помощью установки пароля BIOS SETUP либо с помощью шифрования файловой системы. Чтобы система загружалась с другого носителя, нужно войти в BIOS SETUP. А раз на нем будет пароль, войти в SETUP не получится. С другой стороны,

злоумышленник может извлечь жесткий диск и подключить его к своему компьютеру. Здесь вас спасет только шифрование файловой системы. Тогда ему не удастся подмонтировать вашу файловую систему, поскольку она зашифрована. Зашифровать файловую систему можно, например, с помощью eCryptfs, но ее рассмотрение выходит за рамки этой книги.

15.2.2. Сценарии запуска служб

Рассмотрим содержимое каталогов /etc/init.d и /etc/rc.d. В каталоге /etc/init.d находятся сценарии запуска служб, например, сценарий /etc/init.d/atd управляет службой atd (один из планировщиков, рассмотренных в главе 5), а /etc/init.d/ssh - это сценарий управления SSH-сервером и т.д.

Все сценарии из этого каталога построены так, что принимают следующие параметры:

- start - запуск службы;
- stop - остановка службы;
- restart - перезапуск службы;
- reload - повторное чтение файла конфигурации без перезапуска службы;
- status - вывод состояния службы.

После изменения файла конфигурации службы нужно сделать так, чтобы она перечитала свой файл конфигурации. Для этого используется параметр reload. Этот параметр очень полезен, поскольку не происходит завершение работы службы и не происходит разрыв текущих соединений. Если же нужно сделать так, чтобы новые параметры были применены ко всем соединениям сразу, тогда нужно использовать параметр restart: текущие соединения будут закрыты, а когда пользователи подключатся снова, вступят в силу уже новые параметры.

Пример перезапуска службы:

```
# /etc/init.d/ssh restart
```

В каталогах /etc/rcN.d содержатся ссылки на сценарии из каталога /etc/init.d. Но ссылки не простые. Формат ссылок следующий:

```
S<номер><имя>
K<номер><имя>
```

Если имя ссылки начинается с буквы S, то служба будет запущена (start), то есть сценарию <имя> из каталога /etc/init.d будет передан параметр start. Если имя ссылки начинается с буквы K (kill), то служба будет остановлена (параметр stop).

Номер в имени ссылки просто определяет порядок запуска сценария. Например, сценарий S04gdm3 будет запущен раньше, чем сценарий S05bootlogs.

15.3. Современная система инициализации systemd

15.3.1. Как она работает?

Принцип работы systemd полностью отличается от init, так что не нужно думать, что будет «по образу и подобию». Разработчики systemd изменили не только формат конфигурационных файлов, но и саму концепцию управления службами.

Традиционная система init была довольно медленной, и она тормозила запуск даже быстрых и современных компьютеров. Давайте подумаем, как можно ускорить запуск Linux? Можно пойти по пути init-ng - параллельно запускать службы. Но параллельный запуск - не всегда хорошо. Нужно учитывать зависимости служб. Например, сервис А нужен многим другим сервисам. Пока сервис А не будет запущен, нельзя запускать сервисы, которые от него зависят.

Если сначала запускать основные сервисы и ждать, пока они будут запущены, а потом уже запускать службы, которые от них зависимы, особого выигрыша в производительности по сравнению с init вы не увидите. Но если сервис А (или любой другой, от которого зависят какие-то другие сервисы) запускается долго, то все остальные службы будут ждать его.

Обойти это ограничение можно так. При своем запуске службы проверяют, запущена ли необходимая им служба, по наличию файла сокета. Например, в случае с d-bus - это файл /var/run/dbus/system_bus_socket. Если мы создадим сокеты для всех служб, то мы можем запускать их параллельно, особо не беспокоясь, что произойдет сбой какой-то службы при запуске из-за отсутствия службы, от которой они зависят. Даже если несколько служб, которым нужен сервис А, запустятся раньше, чем сам сервис А: ничего страшного. Каждая из этих служб отправит в сокет (главное, что он уже открыт!)

сообщение, которое обработает сервис А после того, как он запустится. Вот и все.

Но это не единственный способ, посредством которого осуществляется ускорение запуска компьютера, инициализацию которого производит systemd. Она запускает только необходимые сервисы. Остальные же будут запущены по мере необходимости. Концепция отложенного запуска используется и в других операционных системах - например, в macOS (там система инициализации называется launchd) и в Windows (концепция отложенного запуска служб). Так что решение не очень новое, но зато проверенное.

Рассмотрим основные функции systemd:

- Активация на основании сокетов - система инициализации systemd прослушивает сокеты всех системных служб. Сокеты передаются системным службам сразу после запуска сервисов. Благодаря этому осуществляется параллельный запуск сервисов. Также это позволяет перезапускать сервисы без потери любых отправленных им сообщений, то есть пока сервис перезапускается, отправленные ему сообщения накапливаются и он сможет их обработать после того, как будет запущен.
- Активация на основании устройств - systemd может запустить определенные службы, когда станет доступным определенный тип оборудования. Например, вы подключили Bluetooth-адаптер, может быть запущен сервис bluetooth.
- Активация на основании d-bus - служба инициализации может запустить сервисы, которые используют d-bus для межпроцессного взаимодействия, например, когда клиентское приложение попытается связаться с системной службой.
- Активация на основании путей - systemd может запустить службу, если изменится содержание каталога.
- Управление точками монтирования и автоматическим монтированием - система инициализации отслеживает и управляет точками монтирования и автоматического монтирования.
- Снимки системных состояний - благодаря этой возможности systemd может сохранить состояние всех модулей и восстановить предыдущее состояние системы.
- Параллелизация - systemd запускает системные службы параллельно благодаря активации на основании сокетов. Параллельная активация существенно сокращает время загрузки системы.

Как видите, система инициализации - это нечто большее, чем просто программа, которая запускает те или иные сервисы. Впрочем, система init, по сути, как раз и была такой программой, а вот systemd переросла в нечто большее.

15.3.2. Настраиваем systemd

Обилие различных конфигурационных файлов systemd может вызвать затруднения даже у бывалого линуксоида, не говоря уже о пользователе, который видит systemd впервые. Что ж, давайте попытаемся разобраться.

В systemd используется концепция модулей (юнитов). Существующие типы модулей описаны в таблице 15.1.

Таблица 15.1. Типы модулей системы инициализации systemd

Тип	Описание
service	Служба (сервис, демон), которую нужно запустить. Пример имени модуля: network.service. Изначально systemd поддерживала сценарии SysV (чтобы управлять сервисами можно было, как при использовании init), но в последнее время в каталоге /etc/init.d систем, которые используют systemd практически пусто (или вообще пусто), а управление сервисами осуществляется только посредством systemd.
target	Цель. Используется для группировки модулей других типов. В systemd нет уровней запуска, вместо них используются цели. Например, цель multi-user.target описывает, какие модули должны быть запущены в многопользовательском режиме.
snapshot	Снимок. Используется для сохранения состояния systemd. Снимки могут использоваться для перевода системы из одного состояния в другое, например, в состояние сна и пробуждение.

mount	Точка монтирования. Представляет точку монтирования. Система инициализации systemd контролирует все точки монтирования. При использовании systemd файл /etc/fstab уже не главный, хотя все еще может использоваться для определения точек монтирования.
automount	Автоматическая точка монтирования. Используется для монтирования сменных носителей - флешек, внешних жестких дисков, оптических дисков и т.д.
socket	Сокет. Представляет сокет, который находится в файловой системе или в Интернете. Поддерживаются сокеты AF_INET, AF_INET6, AF_UNIX. Реализация довольно интересная. Например, если сервису service1.servive соответствует сокет service1.socket, то при попытке установки соединения с service1.socket будет запущен service1.servive.
device	Устройство. Представляет устройство в дереве устройств. Работает вместе с udev: если устройство описано в виде правила udev, то его можно представить в systemd в виде модуля device.
path	Файл или каталог, созданный где-то в файловой системе
scope	Процесс, который создан извне.
slice	Управляет системными процессами. Представляет собой группу иерархически организованных модулей.
swap	Представляет область подкачки (раздел подкачки) или файл подкачки (свопа)
timer	Представляет собой таймер системы инициализации systemd

Модули хранятся в следующих каталогах:

- /etc/systemd/system/ - обладает самым высоким приоритетом. Здесь содержатся модули, которые созданы и управляются системным администратором. Довольно часто модули в этом каталоге являются ссылками

на файлы, которые находятся на самом деле в каталоге /usr/lib/systemd/system/ (рис. 15.1).

- /run/systemd/system/ - модули, созданные во время выполнения. Приоритет этого каталога ниже, чем каталога /etc/systemd/system/, но выше, чем у /usr/lib/systemd/system.
- /usr/lib/systemd/system/ - модули, которые установлены из пакетов.

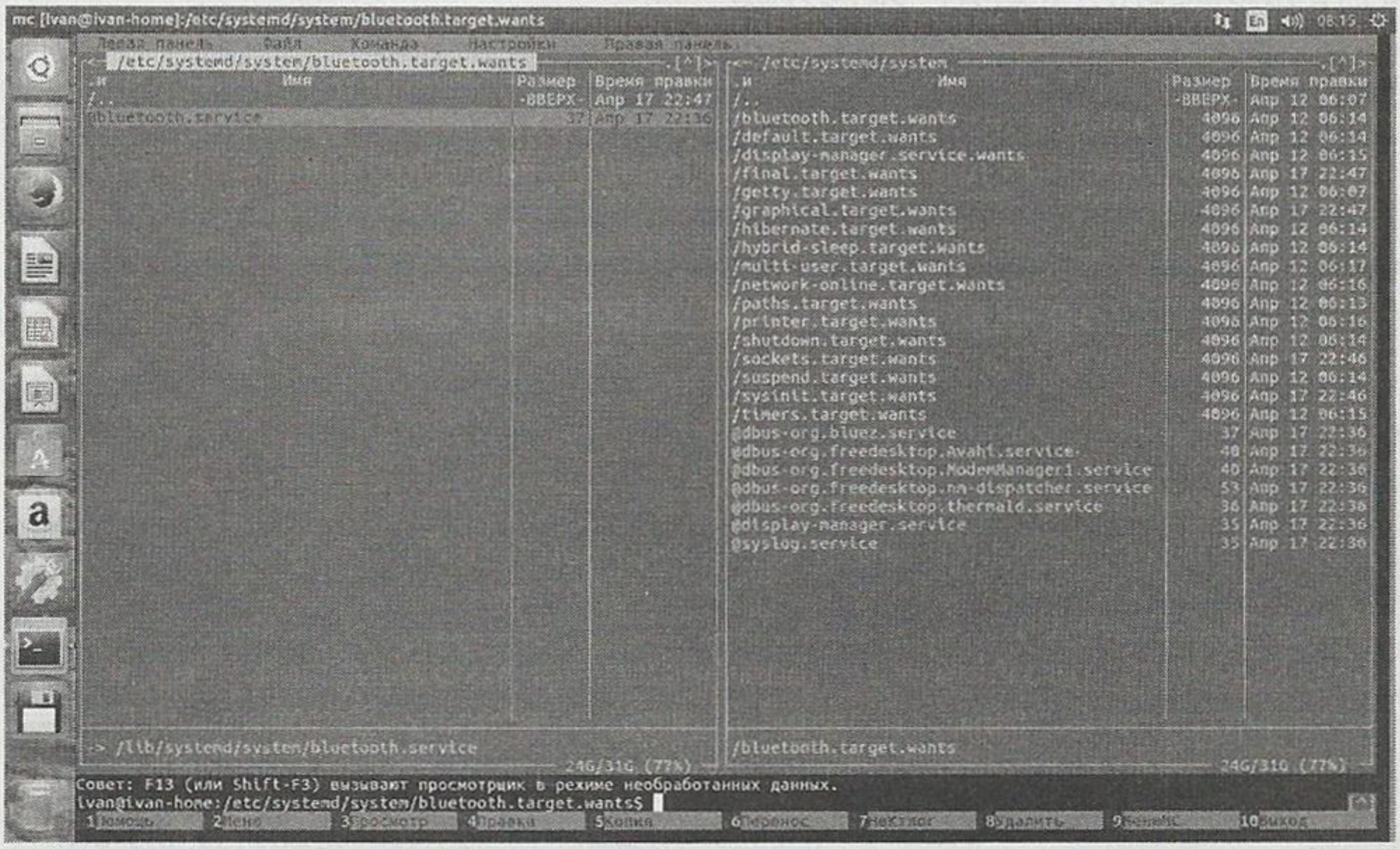


Рис. 15.1. Файл /etc/systemd/system/bluetooth.target.wants/bluetooth.servive на самом деле является ссылкой на /lib/systemd/system/bluetooth.service

Типичный файл модуля типа service приведен в листинге 15.2.

Листинг 15.2. Типичный файл модуля типа service

```
[Unit]
Description=Bluetooth service
Documentation=man:bluetoothd(8)
ConditionPathIsDirectory=/sys/class/bluetooth

[Service]
Type=dbus
BusName=org.bluez
ExecStart=/usr/lib/bluetooth/bluetoothd
NotifyAccess=main
#WatchdogSec=10
```

```
#Restart=on-failure
CapabilityBoundingSet=CAP_NET_ADMIN CAP_NET_BIND_SERVICE
LimitNPROC=1
ProtectHome=true
ProtectSystem=full

[Install]
WantedBy=bluetooth.target
Alias=dbus-org.bluez.service
```

В секции Unit содержится общая информация о сервисе. Эта секция есть и в других модулях, а не только в сервисах.

Секция Service содержит информацию о сервисе. Параметр ExecStart описывает команду, которую нужно запустить. Параметр Type указывает, как сервис будет уведомлять systemd об окончании запуска.

Секция Install содержит информацию о цели, в которой должен запускаться сервис. В нашем случае видно, что сервис будет запущен при активации цели bluetooth.target.

Вы можете использовать эту «болванку» для написания собственного сервиса, который потом нужно поместить в файл /etc/systemd/system/имя_сервиса.service. После этого нужно перезапустить саму systemd, чтобы она узнала о новом сервисе:

```
# systemctl daemon-reload
```

15.3.3. Цели

Цели (targets) являются очень важной концепцией в systemd. Файлы целей *.target предназначены для группировки вместе других юнитов systemd через цепочку зависимостей. Так, модуль цели graphical.target, который используется для запуска графического сеанса, запускает системные службы GDM (файл gdm.service) и Accounts Service (accounts–daemon.service), а также активирует цель multi-user.target. В свою очередь, цель multi–user.target запускает другие системные службы, например, D-Bus (dbus.service) и активирует другие цели вроде basic.target.

Файлы целей находятся в каталоге /lib/systemd/system. В каталогах /etc/systemd/system/<название цели>.wants содержатся ссылки на сервисы, необходимые для выполнения этой цели. Так, для цели bluetooth.target нужен сервис bluetooth.service, поэтому ссылка на этот файл находится в каталоге bluetooth.target.wants.

В systemd имеются предопределенные цели, которые напоминают стандартный набор уровней запуска.

Некоторые цели называются runlevelN.target, чтобы упростить переход бывших пользователей init на systemd, а именно:

- poweroff.target (ссылка runlevel0.target) – завершение работы и отключение системы;
- rescue.target (ссылка runlevel1.target) – однопользовательский режим, среда восстановления;
- multi–user.target (ссылки runlevel2.target, runlevel3.target, runlevel4.target) – многопользовательский режим, без графического интерфейса;
- graphical.target (ссылка runlevel5.target) – многопользовательский режим с графическим интерфейсом
- reboot.target (ссылка runlevel6.target) – завершение работы и перезагрузка системы

Управление службами осуществляется с помощью программы systemctl. Подробнее о службах мы поговорим в следующем разделе, а пока разберемся, как использовать systemctl для завершения работы системы:

- systemctl halt – останавливает систему;
- systemctl poweroff – выключает систему;
- systemctl reboot – перезагружает систему.

Конечно, как по мне, удобнее использовать старые команды halt, poweroff и reboot. Но все же теперь вы знаете, что есть альтернативные способы завершения работы.

15.4. Управление службами

15.4.1. Графические менеджеры

Как любому домашнему пользователю, иногда хочется какой-то графической оболочки для управления службами. Например, чтобы просмотреть,

Д

какие службы запущены, отключить неиспользуемые службы, перезапустить службу в случае изменения ее конфигурации. В мире Linux использование графических конфигураторов считается не совсем правильным, но, как пользователь, я вас понимаю. Поэтому расскажу о конфигураторах, доступных для systemd.

В Ubuntu вы можете использовать конфигуратор systemadm, который содержится в пакете systemd-ui. Этот пакет по умолчанию не устанавливается - разработчики как бы намекают, что негоже редактировать конфигурацию с помощью графических средств. Но поскольку спрос будет, в состав репозитария данный конфигуратор все-таки включили. Итак, введите следующие команды:

```
sudo apt install systemd-ui
sudo systemadm
```

Первая команда установит необходимый пакет, вторая - запустит конфигуратор (рис. 15.2).

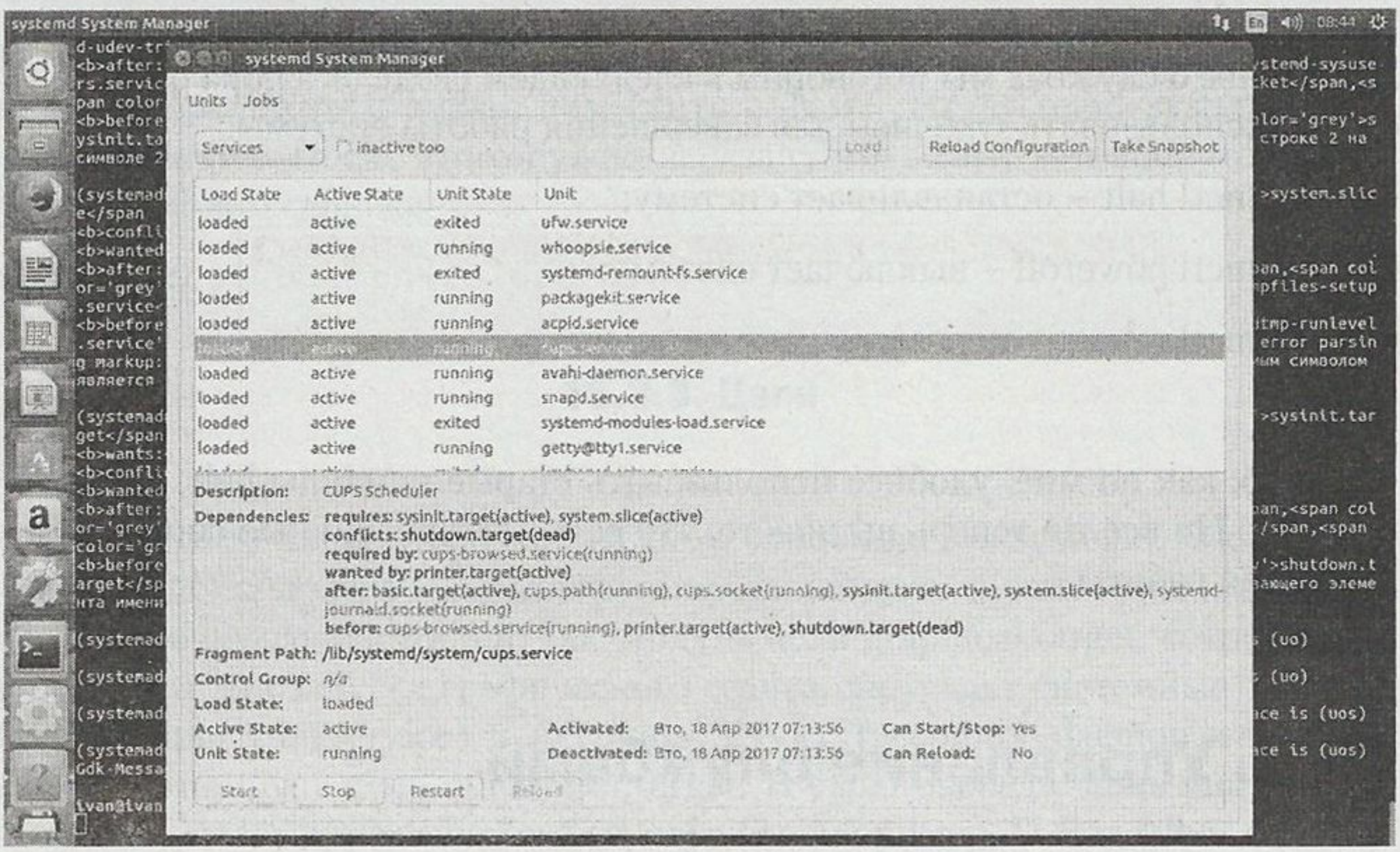


Рис. 15.2. Конфигуратор systemadm

Посмотрим на рис. 15.2. На вкладке Unit предоставляется информация обо всех типах юнитов. Поскольку нас интересуют службы, то выберите из списка Services, чтобы отфильтровать информацию только о службах. Да-

лее просмотрим информацию о какой-то службе, например, о cups.service (служба управления печатью). Данная служба загружена и активна - в соответствующих колонках выводятся строки loaded и active. Текущее состояние юнита - выполняется (running). В нижней части окна (которая не отображается почему-то по умолчанию, что сначала вызвало у меня небольшое замешательство) отображается описание службы и заветные кнопки Start, Stop, Restart и Reload, позволяющие управлять службой. Кнопка Restart перезапускает сервис (останавливает и снова запускает его), а кнопка Reload заставляет сервис перечитать файл конфигурации. Нужно отметить, что возможность перезагрузки (reload) доступна не для всех сервисов.

Обратите внимание на блок Dependencies, в нем описаны зависимости службы:

- **requires** - здесь перечисляются службы, которые нужны для выполнения текущей.
- **conflicts** - службы, с которыми конфликтует текущая служба.
- **required by** - здесь перечисляются зависимые от текущей службы.
- **wanted by** - цель, связанная со службой.
- **after** - перечисляются службы, которые должны быть запущены **до** запуска текущей службы.

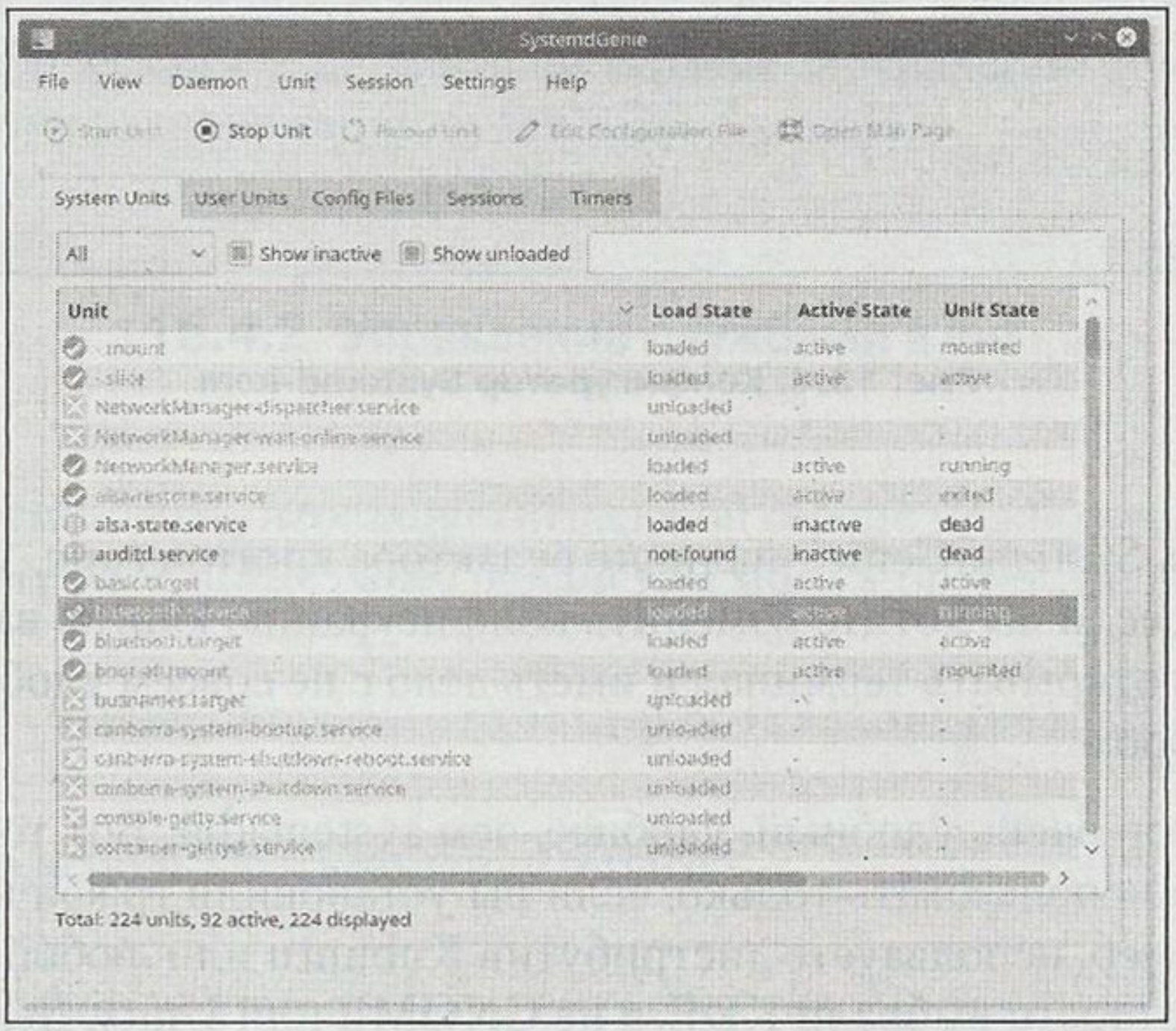


Рис. 15.3. Конфигуратор SystemdGenie

- **before** - перечисляются службы, **перед запуском которых** должна быть запущена текущая служба.

Для графической среды KDE разработан конфигуратор SystemdGenie. Он пришел на смену приложению Systemd-kcm (рис. 15.4) в KDE. SystemdGenie позволяет управлять запуском системных и пользовательских юнитов, настраивать сеансы logind, конфигурировать таймеры и периодические работы, просматривать и изменять файлы конфигурации systemd.

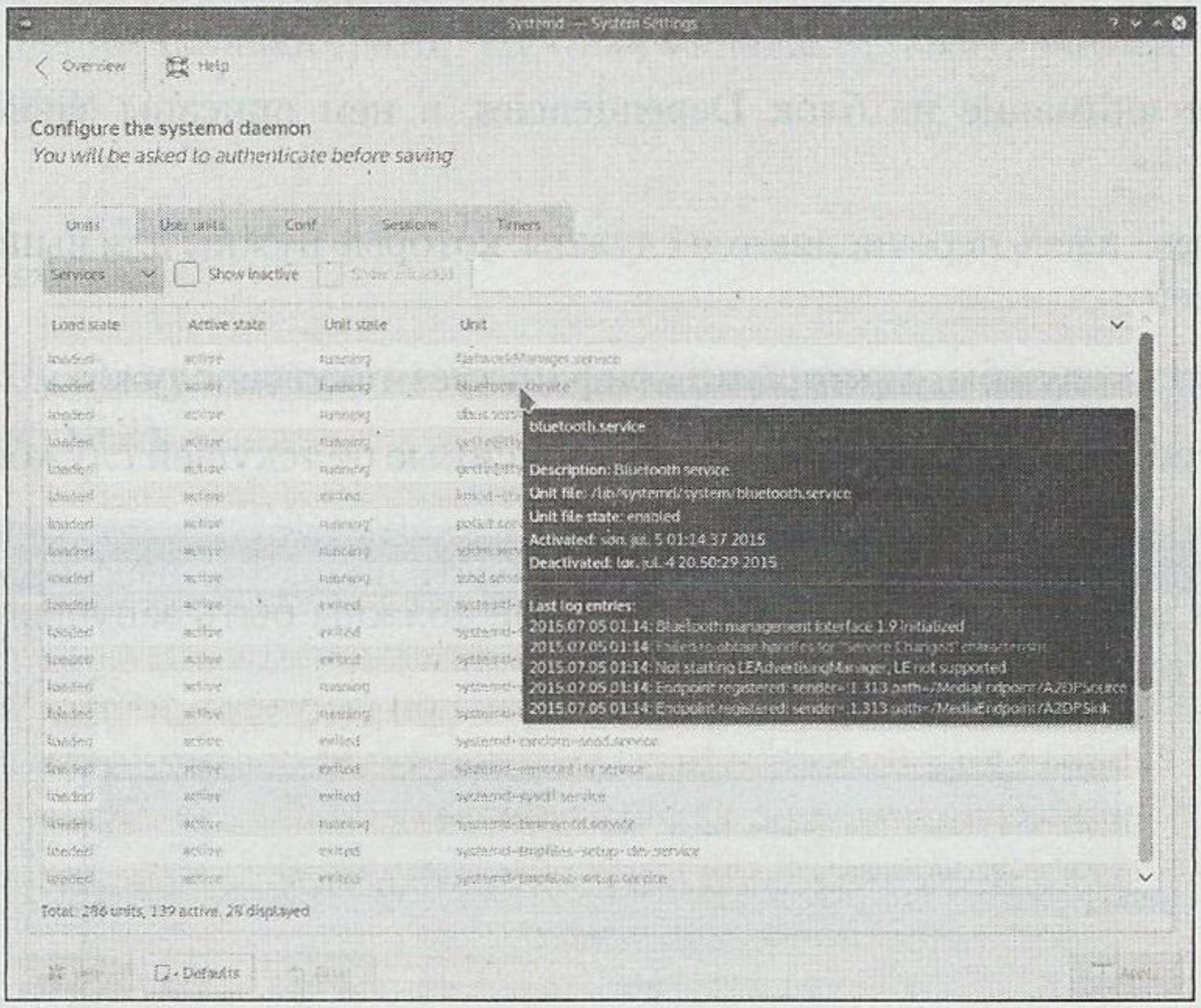


Рис. 15.4. Конфигуратор Systemd-kcm

Особенность SystemdGenie - наличие редактора юнитов (рис. 15.5). Другими словами, если вы хотите изменить конфигурацию юнита, вам совсем не обязательно открывать терминал и иметь дело с не совсем удобным текстовым редактором.

Конфигуратор очень удобный, удобнее, чем systemadm. Вот только его целесообразно использовать только, если вы установили графическую среду KDE, например, используете дистрибутив Kubuntu или любой другой, при установке которого выбрали KDE, а не GNOME или Unity. Дело в том, что для работы этого приложения нужно будет установить библиотеки KDE, которые занимают довольно много места и устанавливать их ради конфигу-

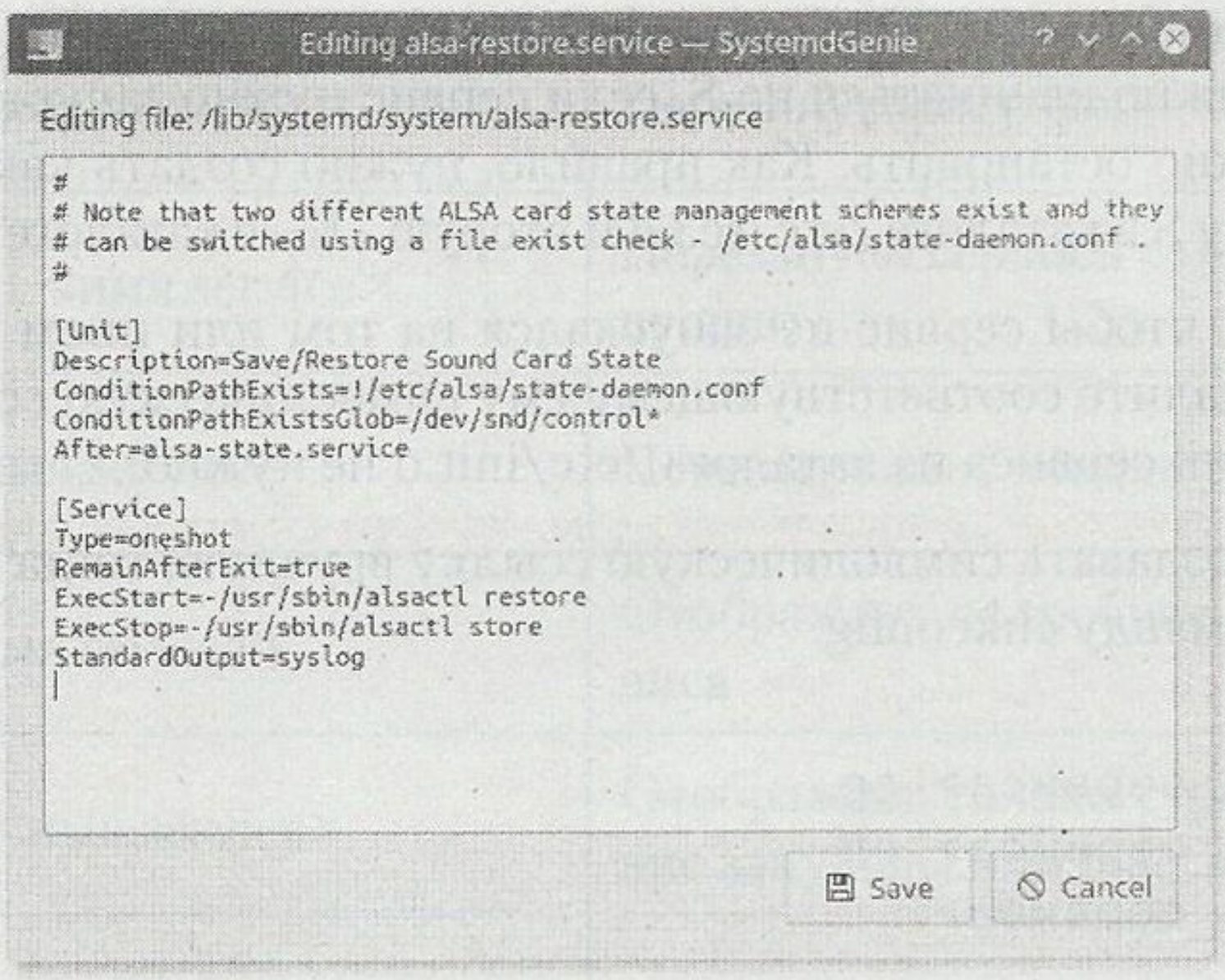


Рис. 15.5. Редактирование юнита

ратора служб, который вы будете использовать не так уж и часто - по меньшей мере, нерационально.

Графический конфигуратор - это хорошо, оно он не всегда доступен, например, если вы арендовали VPS или VDS, у вас будет доступ только по ssh и никакая графическая среда на вашем сервере запущена не будет, следовательно, в вашем распоряжении будет только консоль!

15.4.2. Управление службами в init

Запустить/остановить/перезапустить любой сервис можно непосредственно передав ему описанные ранее параметры start/stop/restart, например:

```
# /etc/init.d/имя_сервиса {start|stop|restart|reload|status}
```

Также можно использовать команду service:

```
# service <имя_сервиса> {start|stop|restart|reload|status}
```

Чтобы настроить сервис на автоматический запуск, вам нужно в одном из каталогов /etc/rcN.d (в зависимости от уровня запуска, на котором должен

запускаться сервис) создать ссылку на файл сервиса из каталога /etc/init.d. Имя ссылки должно начинаться на S, если сервис нужно запустить или K, если сервис нужно остановить. Как правило, нужно создать как минимум две ссылки - одну S в каталоге /etc/rc3.d и вторую - K - в каталоге /etc/rc6.d.

Если вы хотите, чтобы сервис не запускался на том или ином уровне запуска, просто удалите соответствующему ему символическую ссылку. Удалять сам сценарий сервиса из каталога /etc/init.d не нужно!

Если вам лень создавать символическую ссылку вручную, тогда вы можете использовать команду chkconfig:

```
chkconfig <имя_сервиса> on
chkconfig <имя_сервиса> off
chkconfig <имя_сервиса>
```

Первая команда включает сервис, то есть настраивает его для автоматического запуска при старте системы. Вторая - выключает автоматическую загрузку сервиса. Третья просто проверяет, будет ли сервис запускаться автоматически или нет.

А с помощью следующей команды можно увидеть список всех уровней запуска, на которых запускается сервис:

```
chkconfig <имя_сервиса> --list
```

15.4.3. Управление службами в systemd

При использовании системы инициализации systemd управление службами осуществляется посредством программы systemctl. Команда systemctl используется для разных целей, поэтому в таблице 15.2 представлены не все ее параметры, а только те, которые имеют отношение к сервисам.

Таблица 15.2. Параметры программы systemctl

Параметр	Описание
start <имя.service>	Запускает сервис
stop <имя.service>	Останавливает сервис

restart <имя.service>	Перезапускает сервис
try-restart <имя.service>	Перезапуск сервиса только, если он запущен
reload <имя.service>	Перезагружает конфигурацию сервиса
status <имя.service>	Отображает подробное состояние сервиса
is-active <имя.service>	Отображает только строку active (сервис запущен) или inactive (остановлен)
list-units --type service --all	Выводит состояние всех сервисов
enable <имя.service>	Включает сервис (обеспечивает его автоматический запуск)
disable <имя.service>	Отключает сервис (сервис не будет автоматически запускаться при запуске системы)
reenable <имя.service>	Деактивирует сервис и сразу его использует
list-unit-files --type service	Выводит список всех сервисов и сообщает, какие из них активированы, а какие - нет

Примеры:

```
# systemctl start httpd.service
# systemctl stop httpd
```

Первая команда запускает сервис httpd (веб-сервер), вторая - останавливает. Обратите внимание, что «.service» можно не указывать.

Линуксоиды - люди довольно консервативные. Они 22 года мирились с довольно нерасторопной системой инициализации. Таки здравый смысл побе-

дил и сейчас используется systemd - самая быстрая из всех созданных ранее систем. Однако привычка есть привычка. Когда ты управлял службами с помощью команды service последние 20 лет, сложно заставить себя использовать systemctl. Именно поэтому systemd поддерживает команду service и вы можете ее использовать, если вам так привычнее, например:

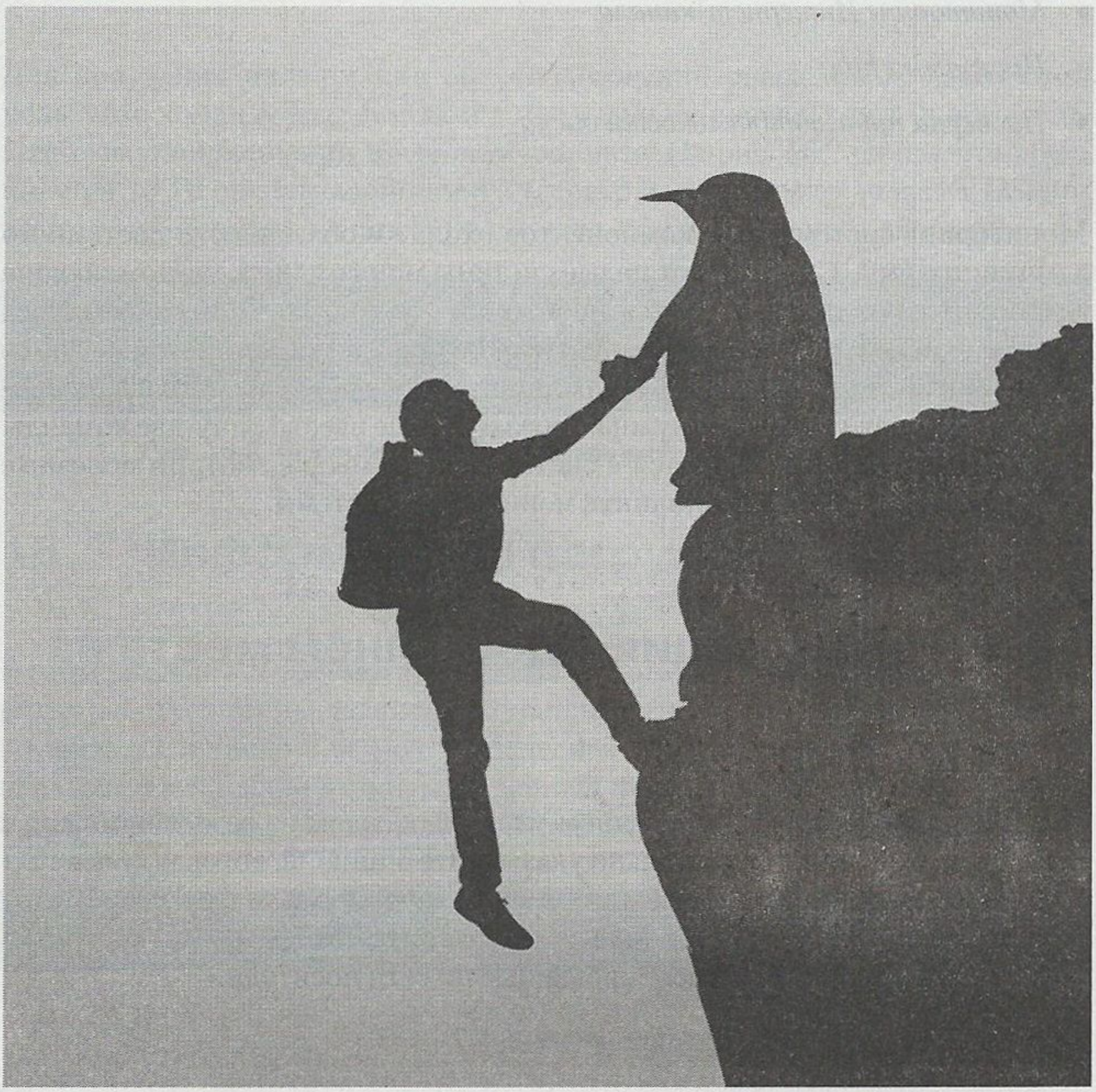
```
# service httpd restart
```

Обратите внимание: # означает права root. Обычный пользователь не может управлять сервисами. В Ubuntu права root можно получить с помощью команды sudo:

```
sudo service httpd restart
```

ГЛАВА 16.

Мониторинг системы



В этой главе:

- *Как узнать, сколько места на диске осталось*
- *Информация об использовании памяти*
- *Мониторинг Интернет-канала*
- *Программа htop*
- *Проверка производительности диска*

Мониторинг системы и ее компонентов входит в обязанности системного администратора. Существуют целые системы мониторинга, уведомляющие администратора о появившихся проблемах - например, если закончилось место на диске или «упал» Интернет-канал. Одна из таких систем - Zabbix. Но обычному пользователю или администратору единственного небольшого сервера такая система не нужна. Гораздо проще ввести одну-две команды и сразу выяснить, что творится в системе. Далее мы рассмотрим полезные утилиты, позволяющие выполнить мониторинг системы.

16.1. Информация о доступном месте на диске

Получить информацию о свободном месте на каждом из разделов можно с помощью команды df, обязательно указывайте опцию -h, которая обеспечит вывод информации в удобном для восприятия человеком формате. Иллюстрация сделана на реальном сервере, как видите, отображаются не только локальные разделы, но и смонтированные по сети носители.

```
root@hosting:/# df -h
Файл.система                                          Размер Использовано  Дост Использовано% Смонтировано в
/dev/sda1                                               30G         8,6G   20G           31% /
none                                                   4,0K            0  4,0K            0% /sys/fs/cgroup
udev                                                   9,8G         8,0K  9,8G            1% /dev
tmpfs                                                  2,0G         5,1M  2,0G            1% /run
none                                                   5,0M            0  5,0M            0% /run/lock
none                                                   9,9G            0  9,9G            0% /run/shm
none                                                   100M            0  100M            0% /run/user
/dev/sdb                                                40G          17G   22G           44% /srv
/dev/sdc1                                               79G          40G   36G           53% /var
//u133859.your-storagebox.de/backup/Hosting-Backups    2,0T         172G  1,8T            9% /mnt/backup
/dev/sdd2                                               43G          32G   12G           75% /media/reiser-hdd
root@hosting:/#
```

Рис. 16.1. Информация о свободном дисковом пространстве

16.2. Информация об использовании оперативной памяти

Для получения информации об использовании оперативной памяти используйте команду free -h. Как и в предыдущем случае, ключ -h обеспечивает вывод информации в удобном формате. На рис. 16.2 видно, что всего имеется 19 Гб оперативной памяти (сервер из себя представляет VDS, поэтому такой размер), из них используется 4.5 Гб, остальное - свободно.

```
root@hosting:/# free -h
             total       used       free     shared    buffers     cached
Память:         19G       4,5G        15G        65M       344M       2,5G
-/+ буферы/кэш:          1,7G        17G
Swap:          5,9G        12M       5,8G
root@hosting:/#
```

Рис. 16.2. Информация об использовании оперативной памяти

16.3. Информация о нагрузке системы

В этом разделе мы рассмотрим несколько команд, позволяющих «мониторить» работу системы. Начнем с команды uptime, которая сообщает, сколько времени прошло с момента последней перезагрузки, а также сколько пользователей вошло в систему в данный момент и среднюю загрузку за 1, 5 и 15 минут. Результат выполнения этой команды приведен на рис. 16.3.

```
root@hosting:/# uptime
 16:23:42 up 56 days, 13:20,  1 user,  load average: 0,39, 0,64, 0,66
root@hosting:/#
```

Рис. 16.3. Команда uptime

В нашем случае система работает 56 дней, 13 часов и 20 минут. В системе зарегистрирован 1 пользователь, а средняя нагрузка невысока.

Получить информацию об использовании процессорного времени и о запущенных процессах позволяет команда htop. В отличие от команды top, она не только выводит список запущенных процессов, но и сообщает о нагрузке на то или иное ядро процессора. На рис. 16.4 показано, что имеется 6 ядер, из них больше всего используется первое - 23.8%, остальные простаивают (практически). Также выводится и другая информация - об использовании памяти, подкачки (**Swp**), информация о запущенных процессах, о средней нагрузке, uptime. Под шапкой выводится список запущенных процессов подобно команде top.

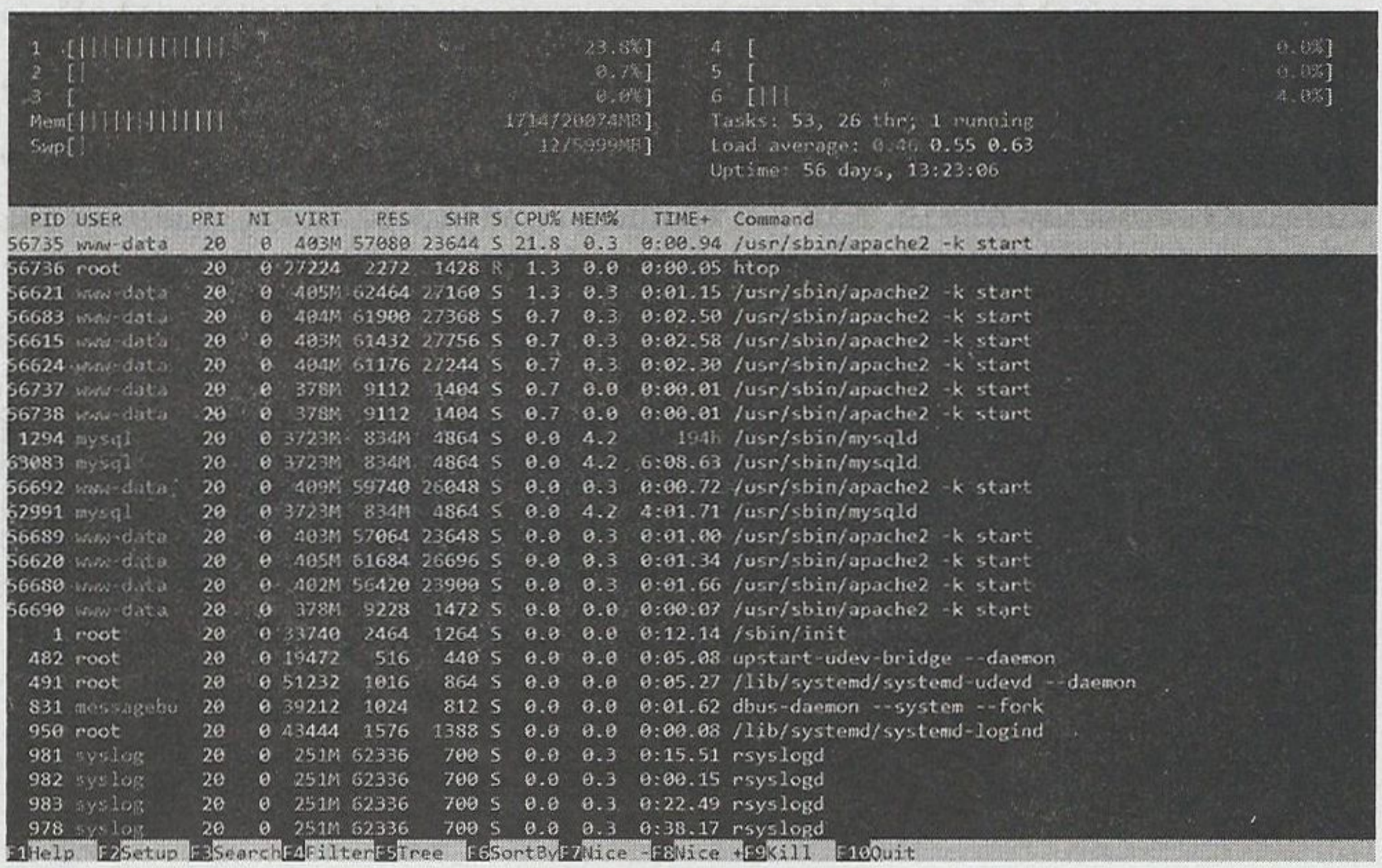


Рис. 16.4. Команда htop

Отличительная особенность программы htop - возможность использования функциональных клавиш по образу и подобию файлового менеджера mc. Так, нажав F2, можно настроить, что и где выводить. Клавиша F3 используется для поиска нужного вам процесса, F4 - позволяет отфильтровать ненужные процессы. На рис. 16.5 я отфильтровал список процессов, оставил только процессы mysqld.

Клавиша <F5> отображает дерево процессов, а клавиша <F6> позволяет отсортировать список процессов по другому столбцу (для сортировки нужно выйти из режима дерева, если вы его активировали). По умолчанию со-

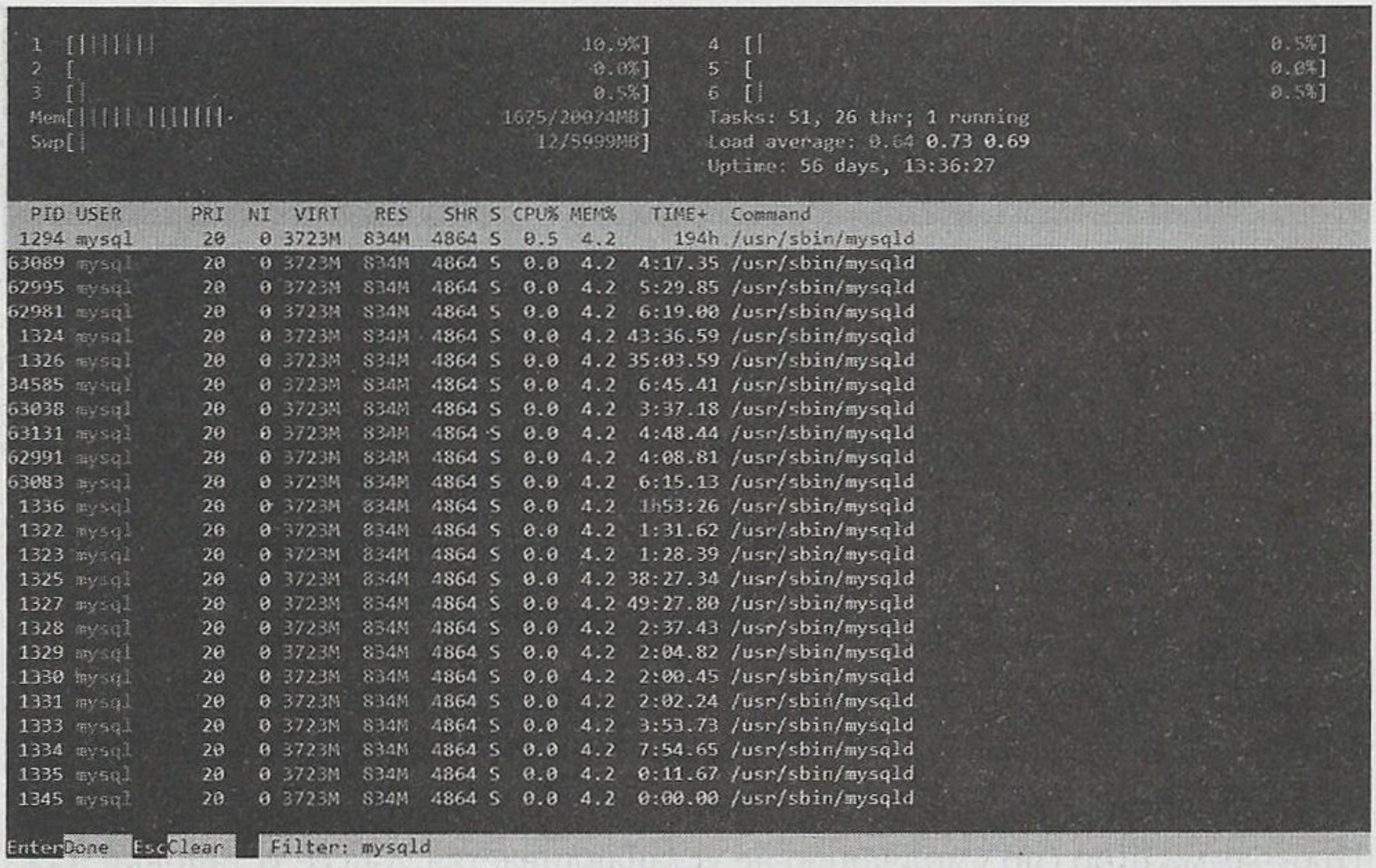


Рис. 16.5. Фильтр процесса

ртировка осуществляется по столбцу CPU% (процессорное время), но вы можете отсортировать по любому другому, например, по Mem (память).

Клавиши <F7> (Nice -) и <F8> (Nice +) позволяют изменять значение Nice, то есть приоритет процесса. Для изменения приоритета выделите нужный вам процесс и нажмите соответствующую клавишу (<F7> или <F8>). Клавиша F9 завершает процесс, а F10 - завершает работу программы.

16.4. Зарегистрированные пользователи

Команда users позволяет получить информацию о том, кто вошел в систему. Посмотрите на рис. 16.6. Команда users просто возвращает список зарегистрированных в системе пользователей (сейчас активен только один пользователь). Команда w (кроме информации об uptime) выводит информацию о том, откуда он вошел (IP-адрес), как именно (по ssh), когда именно (LOGIN), информацию об использовании процессора. Команда who выводит только имя пользователя, название консоли, время и дату входа, а также IP-адрес.

```
root@hosting:/# users
denis
root@hosting:/# w
 16:32:57 up 56 days, 13:29,  1 user,  load average: 0,86, 0,77, 0,68
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
denis    pts/1    168.163.24.170   12:09    1.00s  0.11s  0.01s sshd: denis [priv]
root@hosting:/# who
denis    pts/1        2017-04-20 12:09 (168.163.24.170)
root@hosting:/#
```

Рис. 16.6. Информация о пользователях

16.5. Мониторинг Интернет-канала

16.5.1. Программа iftop

Существуют различные утилиты для мониторинга Интернет-канала. Они позволяют узнать уровень загрузки канала, что помогает при диагностике проблем с ним.

Программа iftop (по умолчанию не устанавливается, для ее установки нужно установить одноименный пакет) позволяет произвести мониторинг Интернет-канала. Помимо информации о загрузке канала программа выводит даже IP-адреса узлов, с которыми осуществляется обмен данными (см. рис. 16.7).

```
                19,1Mb           38,1Mb           57,2Mb           76,3Mb           95,4Mb
10.10.1.20                   => 217.76.195.210                  0b      92,7kb      42,2kb
                             <=                                 0b      10,4kb      5,04kb
10.10.1.20                   => 195.200.244.102                 44,5kb  84,5kb      118kb
                             <=                                 6,42kb  12,1kb      16,7kb
10.10.1.20                   => 46.229.168.66                   0b      16,8kb      7,66kb
                             <=                                 0b      840b        423b
10.10.1.20                   => 198.204.243.138                 0b      12,7kb      11,5kb
                             <=                                 0b      738b        630b
10.10.1.20                   => 10.10.2.16                      2,26kb  2,04kb      1,54kb
                             <=                                 2,28kb  2,10kb      1,60kb
10.10.1.20                   => 188-163-24-170.broadband.kyivstar.net  3,42kb  3,36kb  3,31kb
                             <=                                 160b    160b        247b
10.10.1.20                   => 204.12.208.10                   0b      479b        218b
                             <=                                 0b      604b        275b
10.10.1.20                   => GW1.zarina.ua                   0b      116b        340b
                             <=                                 0b      221b        630b
10.10.1.20                   => 77-239-176-241.static.vega-ua.net  0b   32b         8,85kb
                             <=                                 0b      64b         459b
10.10.1.20                   => 46.229.168.69                   0b      0b          5,76kb
                             <=                                 0b      0b          367b
10.10.1.20                   => c80-216-219-141.bredband.comhem.se  0b   0b          131b
                             <=                                 0b      0b          175b
10.10.1.20                   => 46.229.168.65                   0b      0b          15b
                             <=                                 0b      0b          15b

TX:             cum:   550kB   peak:   546kb          rates:   50,2kb  213kb   200kb
RX:                    72,8kB          66,1kb                   8,86kb  27,2kb  26,5kb
TOTAL:                 623kB           612kb                    59,1kb  240kb   226kb
```

Рис. 16.7. Программа iftop

Все соединения вашего сервера (или рабочей станции - без разницы) как на ладони. Видно направление передачи информации, IP-адреса, объемы передаваемых данных.

На сервере может быть несколько интерфейсов. Выбрать нужный можно с помощью опции -i:

```
iftop -i ens33
```

Другие полезные параметры этой программы указаны в таблице 16.1.

Таблица 16.1. Полезные параметры программы iftop

Параметр	Описание
-P	Показывать порт
-n	Не разрешать IP-адреса в имена узлов
-B	Отображать график в байтах/сек, а не в битах/сек.
-F сеть/маска	Трафик будет считываться по указанному диапазону адресов, например, iftop -F 111.11.11.11/24.

Программа iftop по умолчанию не устанавливается, для ее установки нужно установить одноименный пакет.

16.5.2. Программа nload

Программа nload - инструмент, показывающий загрузку канала в псевдографическом режиме (рис. 16.8). Клавиши вверх/вниз и влево/вправо позволяют выбрать другой сетевой интерфейса, а клавиша F2 показывает меню настроек.

Программа nload не устанавливается по умолчанию, для ее установки нужно установить пакет nload.

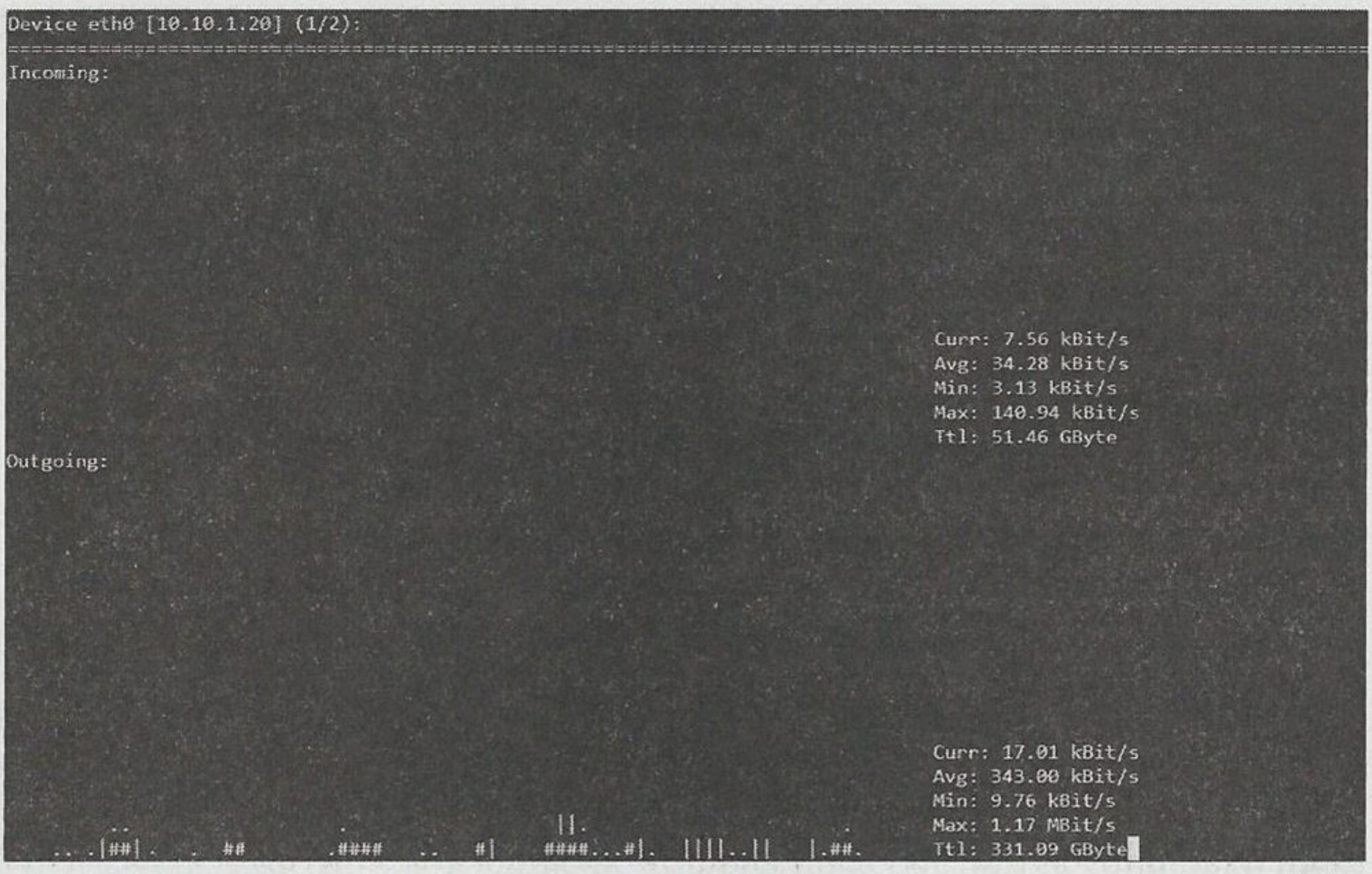


Рис. 16.8. Утилита nload

16.5.3. Программа iptraf-ng

Программа iptraf-ng предназначена для мониторинга сетевой активности в режиме реального вида. Программа довольно проста в использовании, поэтому особо нет смысла ее описывать. Главное, что вы знаете, что она есть. При запуске программы вы можете указать следующие параметры:

- -i - задает интерфейс для мониторинга. Можно выбрать все интерфейсы, указав all (iptraf -i all);
- -g - мониторинг общей статистики интерфейсов;
- -d - подробный мониторинг определенного интерфейса;
- -s - подробный мониторинг активности по протоколам TCP/UDP, на выбранном интерфейсе.

В принципе, вы можете и не указывать параметры, поскольку при запуске программы все равно отображается меню (рис. 16.9), в котором нужно выбрать информацию, которую вы хотите просмотреть. Программа также предоставляет вам выбрать интерфейс, который нужно мониторить (рис.

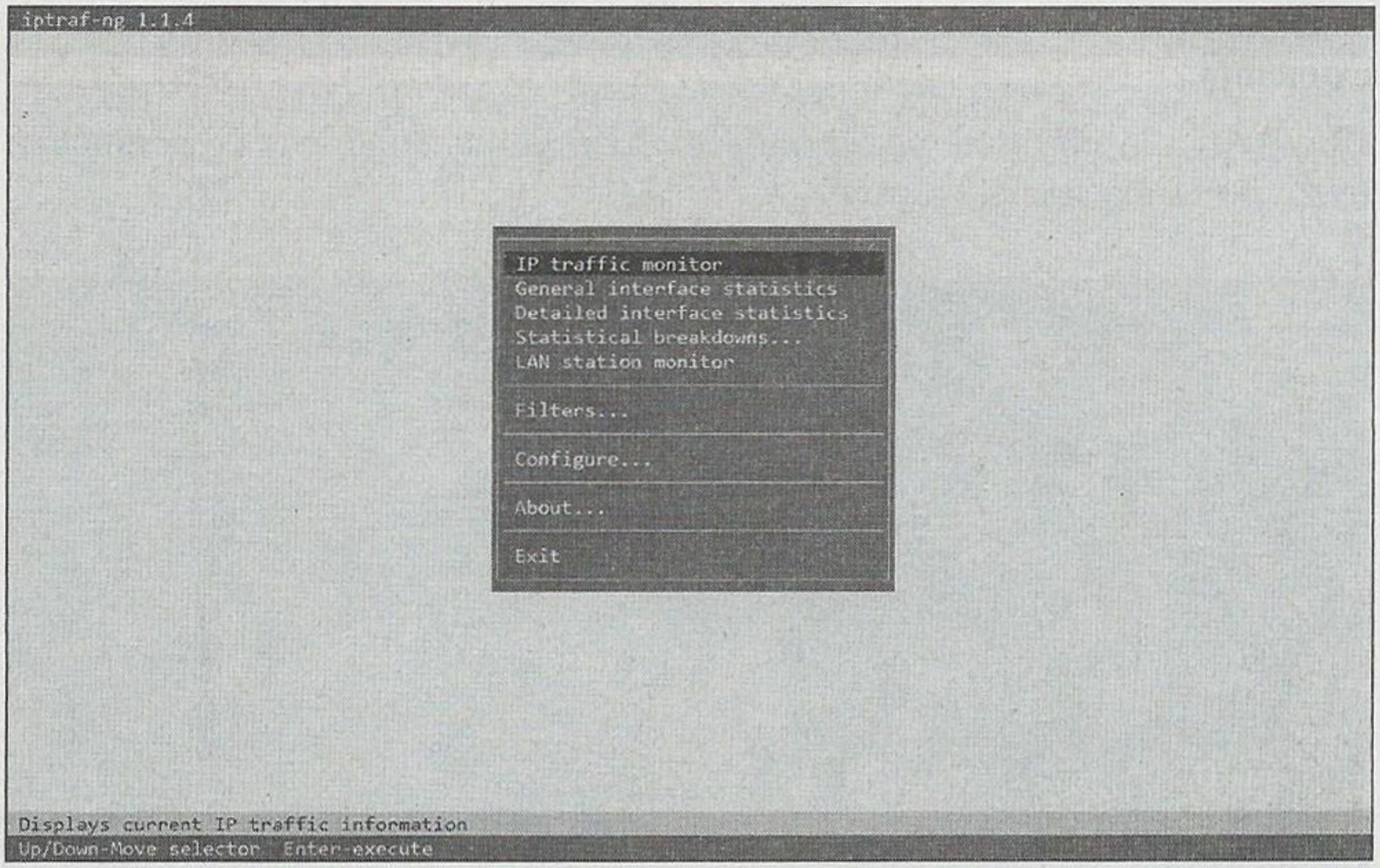


Рис. 16.9. Основное меню iptraf-ng (при запуске программы)

16.10). Далее программа показывает IP-адреса пользователей, соединенных с вашим компьютером (в данном случае IP-адрес 10.10.1.20 - это адрес сер-

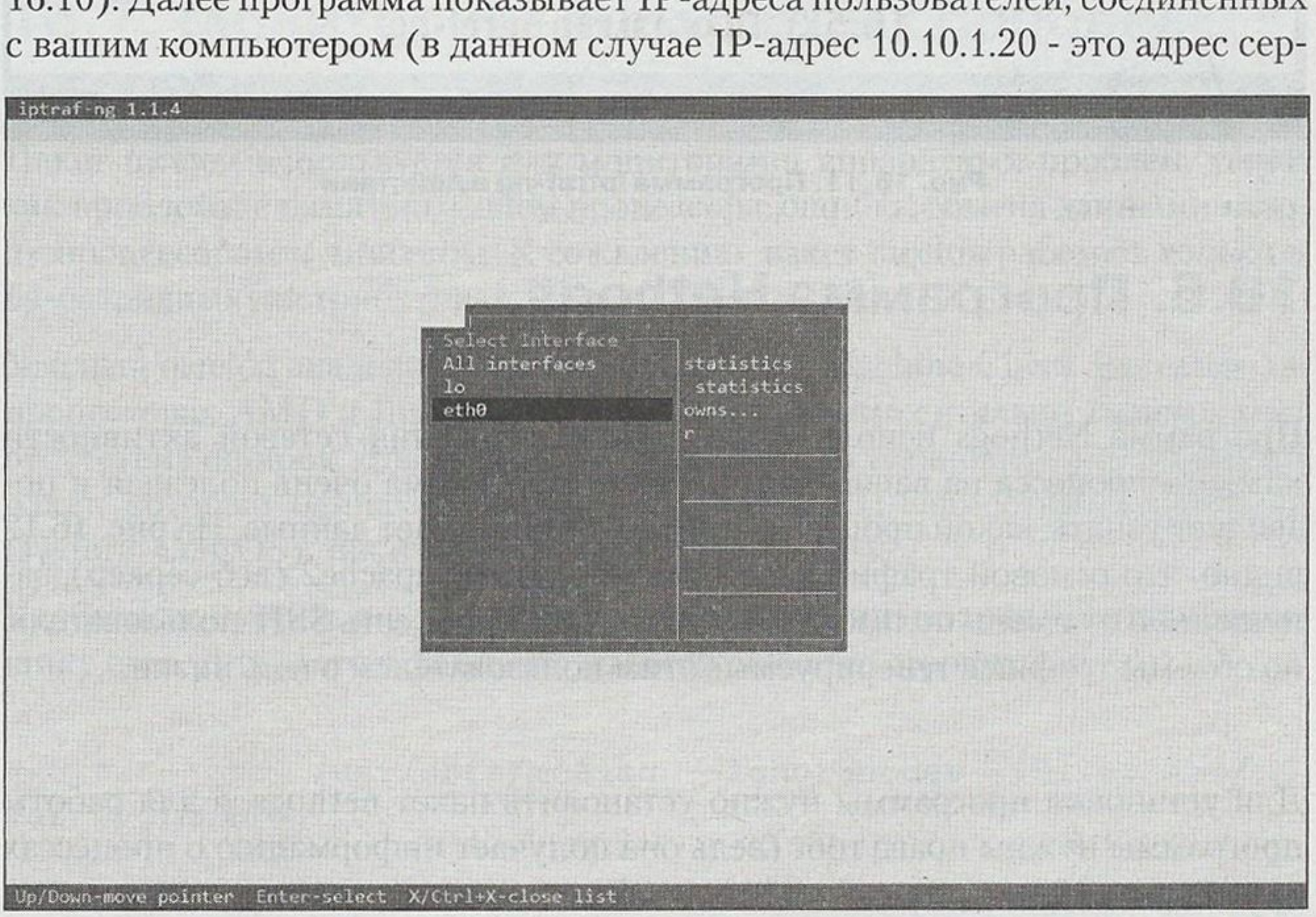


Рис. 16.10. Выбор сетевого интерфейса (при выборе пункта меню IP traffic monitor)

вера), номера портов, выводит количество байтов и пакетов для каждого соединения.

Программа iptraf-ng не устанавливается по умолчанию, для ее установки нужно установить пакет iptraf-ng.

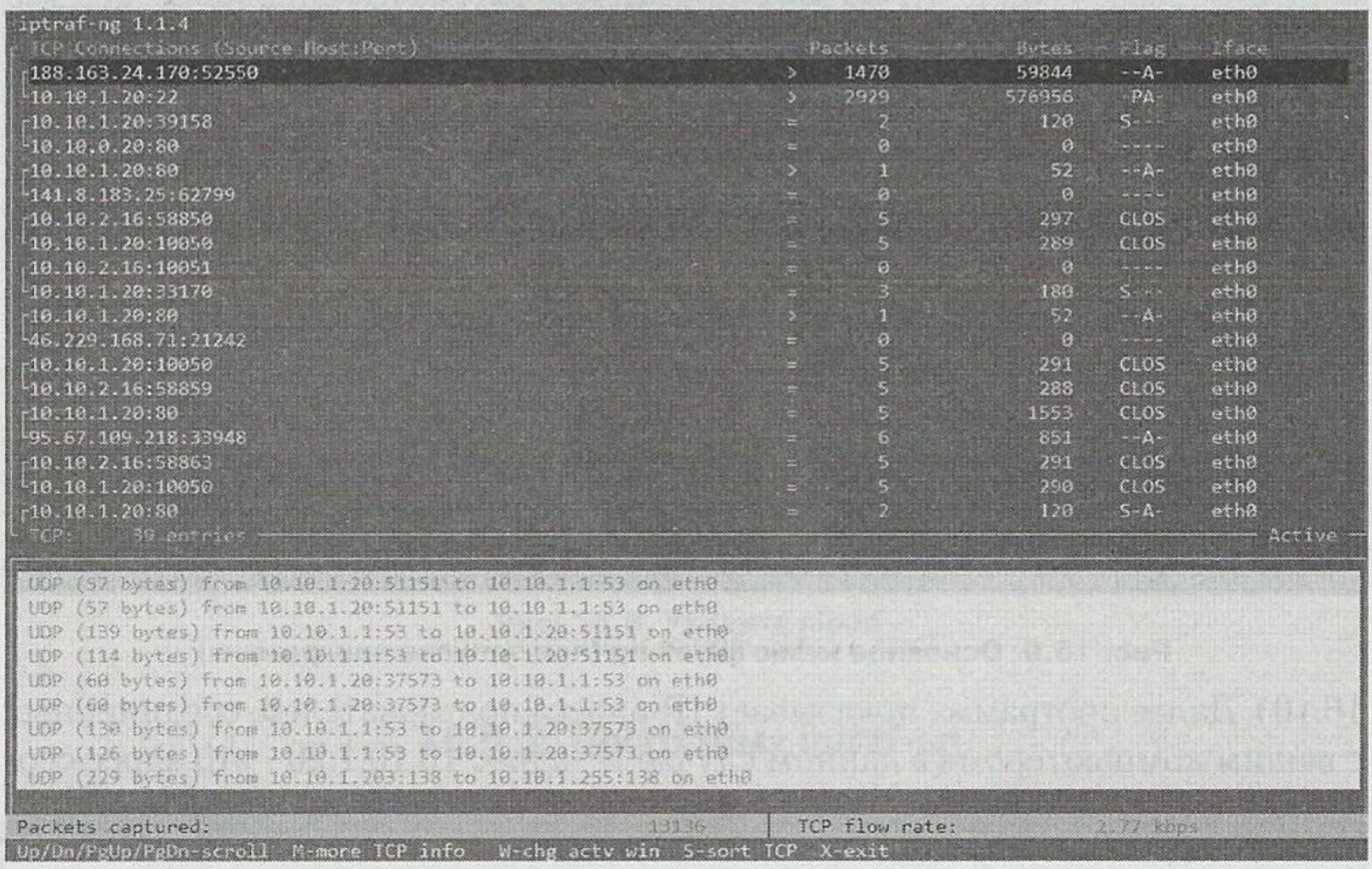


Рис. 16.11. Программа iptraf-ng в действии

16.6. Программа Nethogs

Программа Nethogs используется для исследования сетевой активности каждого процесса на вашем компьютере. Программа очень полезная и позволяет узнать, какой процесс отправляет/принимает данные. На рис. 16.12 видно, что основой трафик адресован программе apache2 (веб-сервер), запущенной от имени пользователя www-data. Также есть SSH-пользователь, но объемы трафика, генерируемые этим пользователем очень низки.

Для установки программы нужно установить пакет nethogs, а для работы программы нужны права root (ведь она получает информацию о процессах других пользователей).

```
NetHogs version 0.8.0

  PID USER     PROGRAM                                          DEV        SENT      RECEIVED
    ? root     10.10.1.20:80-194.44.244.130:53869                          58.770      2.893 KB/sec
    ? root     10.10.1.20:80-46.219.45.103:57548                           10.748      0.753 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21776                            3.132      0.690 KB/sec
61781 www-data /usr/sbin/apache2                                  eth0       3.070      0.463 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21777                            3.286      0.452 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21778                            1.650      0.357 KB/sec
    ? root     10.10.1.20:80-194.44.244.130:53874                           5.432      0.341 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21775                            2.041      0.335 KB/sec
    ? root     10.10.1.20:80-194.44.244.130:53871                           5.057      0.325 KB/sec
    ? root     10.10.1.20:80-194.44.244.130:53870                           5.746      0.287 KB/sec
    ? root     10.10.1.20:80-194.44.244.130:53873                           4.888      0.248 KB/sec
61946 www-data /usr/sbin/apache2                                  eth0       4.660      0.203 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21774                            1.357      0.179 KB/sec
    ? root     10.10.1.20:80-194.44.244.130:53872                           3.173      0.171 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59880                            0.070      0.071 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59871                            0.071      0.071 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59868                            0.071      0.069 KB/sec
27765 denis    sshd: denis@pts/1                                  eth0       2.250      0.042 KB/sec
    ? root     10.10.1.20:80-46.229.168.73:49410                            0.014      0.027 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59855                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59847                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-46.229.168.74:37158                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59841                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59836                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-46.219.45.103:57564                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59830                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-95.132.77.185:21771                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59823                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-46.229.168.67:60832                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-46.229.168.67:23462                            0.000      0.000 KB/sec
    ? root     10.10.1.20:10050-10.10.2.16:59821                            0.000      0.000 KB/sec
    ? root     10.10.1.20:80-46.219.45.103:57563                            0.000      0.000 KB/sec
```

Рис. 16.12. Программа Nethogs

16.7. Мониторинг аппаратных проблем

Пакет mcelog используется для мониторинга аппаратных проблем, таких как проблемы с памятью, кэшем процессора, ошибки обмена данными между процессором и чипсетом. К сожалению, пакет mcelog работает только в 64-разрядных дистрибутивах.

Утилита mcelog анализирует состояние MCE (Machine Check Exception) в процессорах AMD и Intel и делает заключение относительно наличия или отсутствия ошибок. Для установки mcelog нужно установить одноименный пакет, который одинаково называется, как в RH-совместимых системах (Fedora, CentOS), так и в Debian (Debian, Ubuntu).

Чтобы не запускать mcelog вручную, лучше поручить эту задачу планировщику cron. Для этого в файл /etc/crontab добавьте строку:

```
*/5 * * * *    /usr/sbin/mcelog --ignorenodev --filter >> /
var/log/mcelog
```

Осталось научиться проверять журнал. В нем нужно искать строку «hardware error», которую добавляет mcelog. Для этого используйте команду:

```
sudo grep -i "hardware error" /var/log/mcelog
```

Также желательно прописать параметры ядра в конфигурационном файле загрузчика Linux (для процессоров Intel):

```
mce=3
```

А для процессоров AMD:

```
mce=3 mce=bootlog
```

Если у вас есть аппаратные ошибки, пока не поздно, нужно задуматься о замене «железа». В этом случае остановка сервера будет плановой и вы сможете корректно завершить работу сервера без потери данных. Не нужно ждать, пока ваш сервер сломается.

16.8. Проверка производительности диска

Некоторые не очень честные хостеры предлагают серверы с SSD-дисками, но на самом деле устанавливают самые обычные жесткие диски или же ограничивают производительность дисковой подсистемы так, что SSD работает как обычный HDD. Как проверить производительность диска?

В Windows есть прекрасная утилита CrystalDisk Mark, позволяющая в различных режимах протестировать производительность жесткого диска. В Linux можно протестировать производительность штатными средствами. Да, результаты будут не такими качественными, зато нам ничего не нужно устанавливать. Просто введите команду:

```
dd if=/dev/zero of=temp bs=1M count=2048
```

Здесь мы берем устройство /dev/zero и заполняем ним (то есть нулем) файл temp, размер блока 1Мб, количество блоков - 2048, то есть мы создаем файл

размером 2 Гб. Программа сообщит скорость операции и время, затраченное на ее выполнение. На рис. 16.13 приведен результат с одного из моих серверов - 1.4 ГБ/с, файл размером 2 Гб был создан за 1.56 секунды.

```
root@hosting:/var/www# dd if=/dev/zero of=temp bs=1M count=2048
2048+0 записей получено
2048+0 записей отправлено
скопировано 2147483648 байт (2,1 GB), 1,56953 с, 1,4 GB/с
root@hosting:/var/www#
```

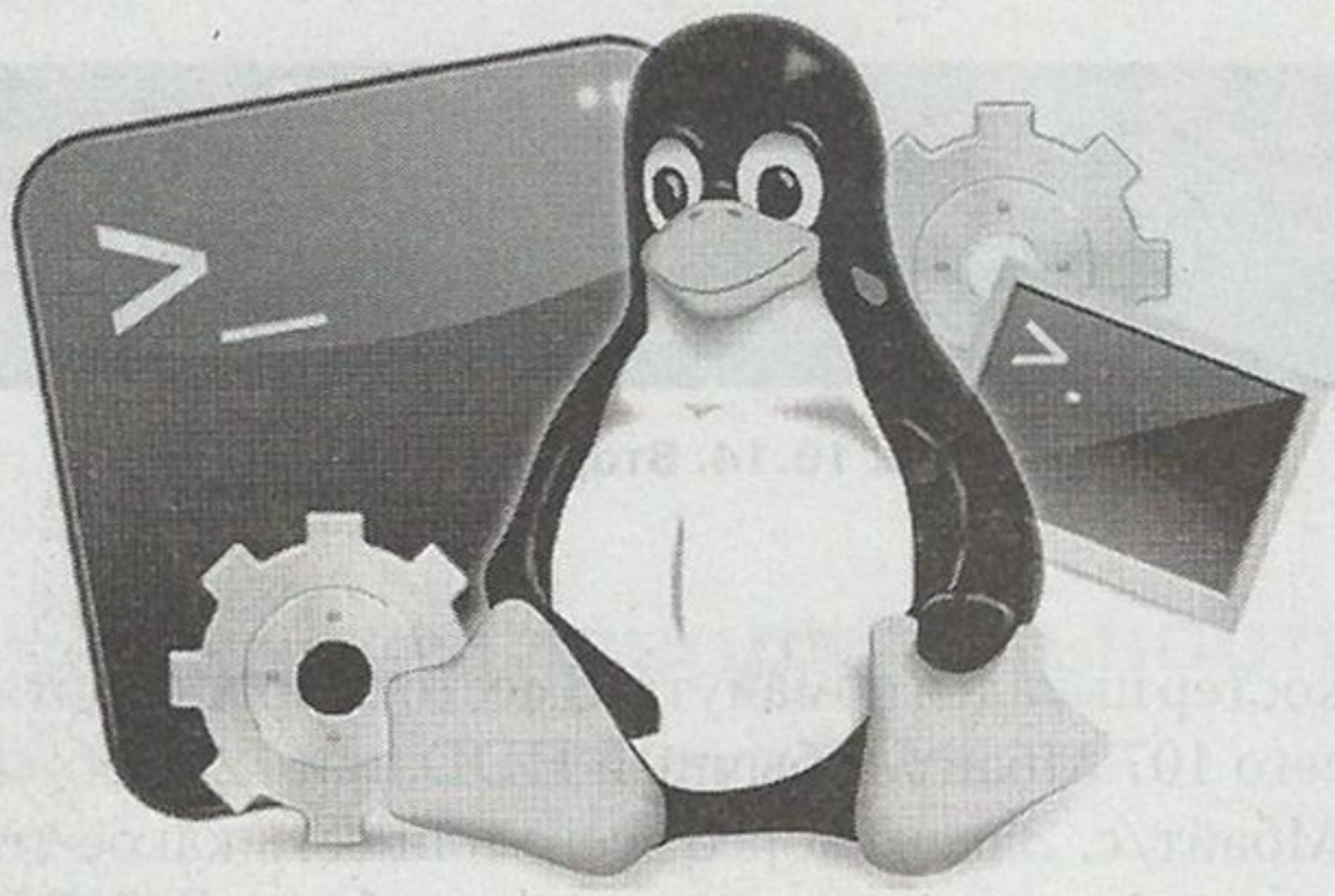
Рис. 16.13. Скорость выполнения операции

А на рис. 16.14 приведен результат одного из не очень честных хостеров, которые обещали производительность на уровне SSD-диска (а это как минимум должно быть 250 Мбайт/с при SATA II).

```
[den@v188134 /]$ sync; dd if=/dev/zero of=~/temp bs=1M count=2048; sync
2048+0 записей считано
2048+0 записей написано
 скопировано 2147483648 байт (2,1 GB), 20,1382 с; 107 MB/с
[den@v188134 /]$
```

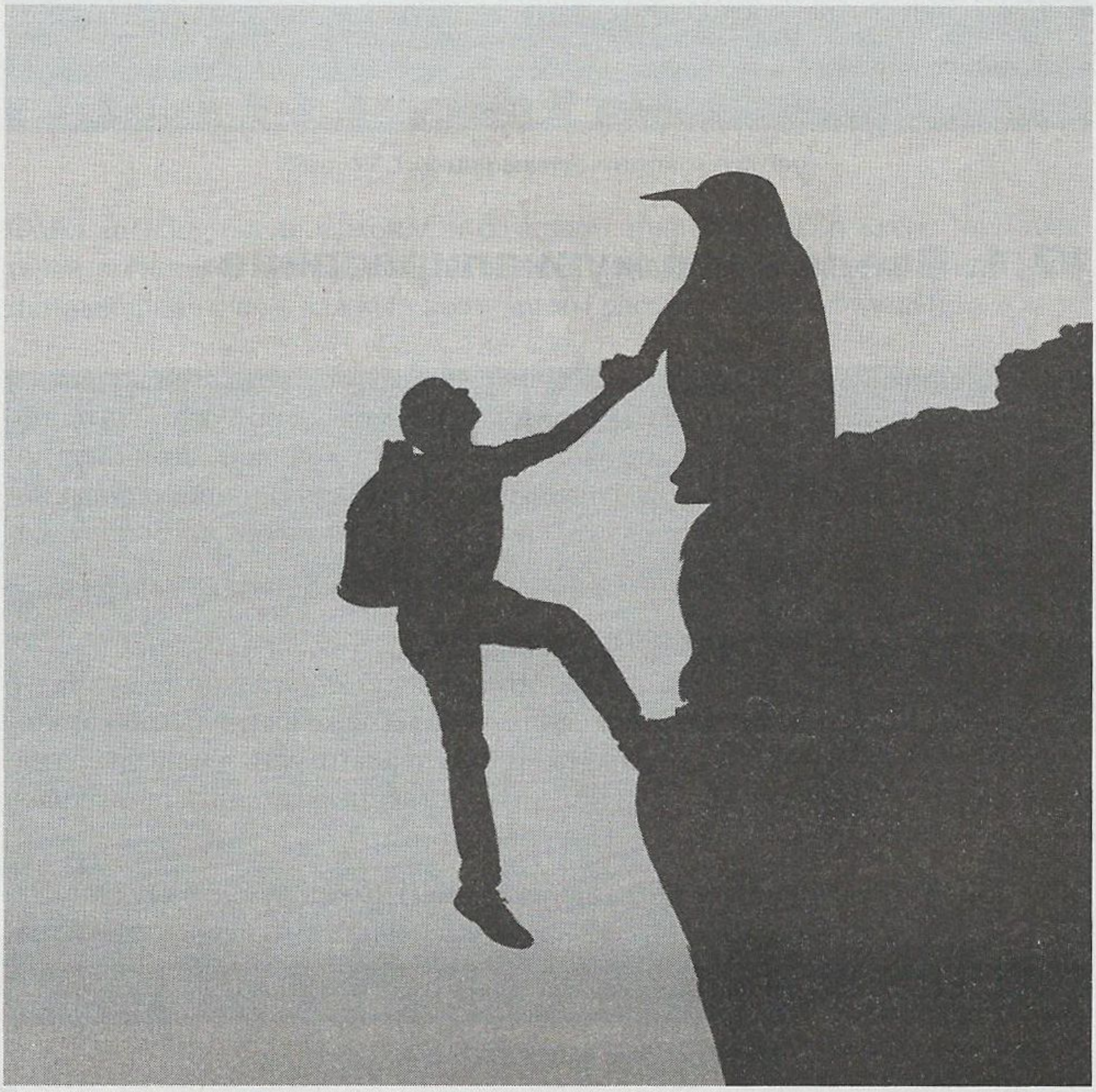
Рис. 16.14. Второй результат

Как видите, хостер пытался обмануть. Вместо скорости хотя бы 250 Мбайт/с я получил всего 107 Мбайт - обычный HDD при 7200 об/мин. может выдавать до 150 Мбайт/с. Здесь скорее всего или есть какое-то ограничение на использование накопителя или установлен обычный HDD. Вот так и верь рекламе!



ГЛАВА 17.

Журналирование



В этой главе:

- *Что такое журналирование*
- *Основные системные журналы*
- *Исследуем сообщения ядра*
- *Команда journalctl*
- *Ротация журналов*

17.1. Введение в журналирование

Операционная система Linux и все ее службы ведут множество журналов (протоколов, логов) своей работы. Протоколируется все - от какой-то ошибки до входа того или иного пользователя в систему. Журналы помогают понять причину сбоя, помочь в расследовании попыток несанкционированного доступа или просто убедиться, что система работает в обычном порядке.

Журналы хранятся в каталоге /var/log. На рис. 17.1 приведено содержимое этого каталога. Давайте разберемся, что есть что.

Некоторые большие сервисы вроде Apache ведут собственные журналы. Так, веб-сервер Apache2 загружает свои журналы даже в отдельный каталог - /var/log/apache2. Собственный каталог с журналами есть и менеджера пакетов apt - /var/log/apt. Также видно, что собственные каталоги протоколов есть у системы печати CUPS и FTP-сервера proftpd.

Некоторые подсистемы не имеют собственного каталога, поэтому записывают журналы прямо в каталог /var/log. Так, сообщения о неполадках в графической подсистеме нужно искать в файле Xorg.0.log, в данном случае 0 - номер монитора.

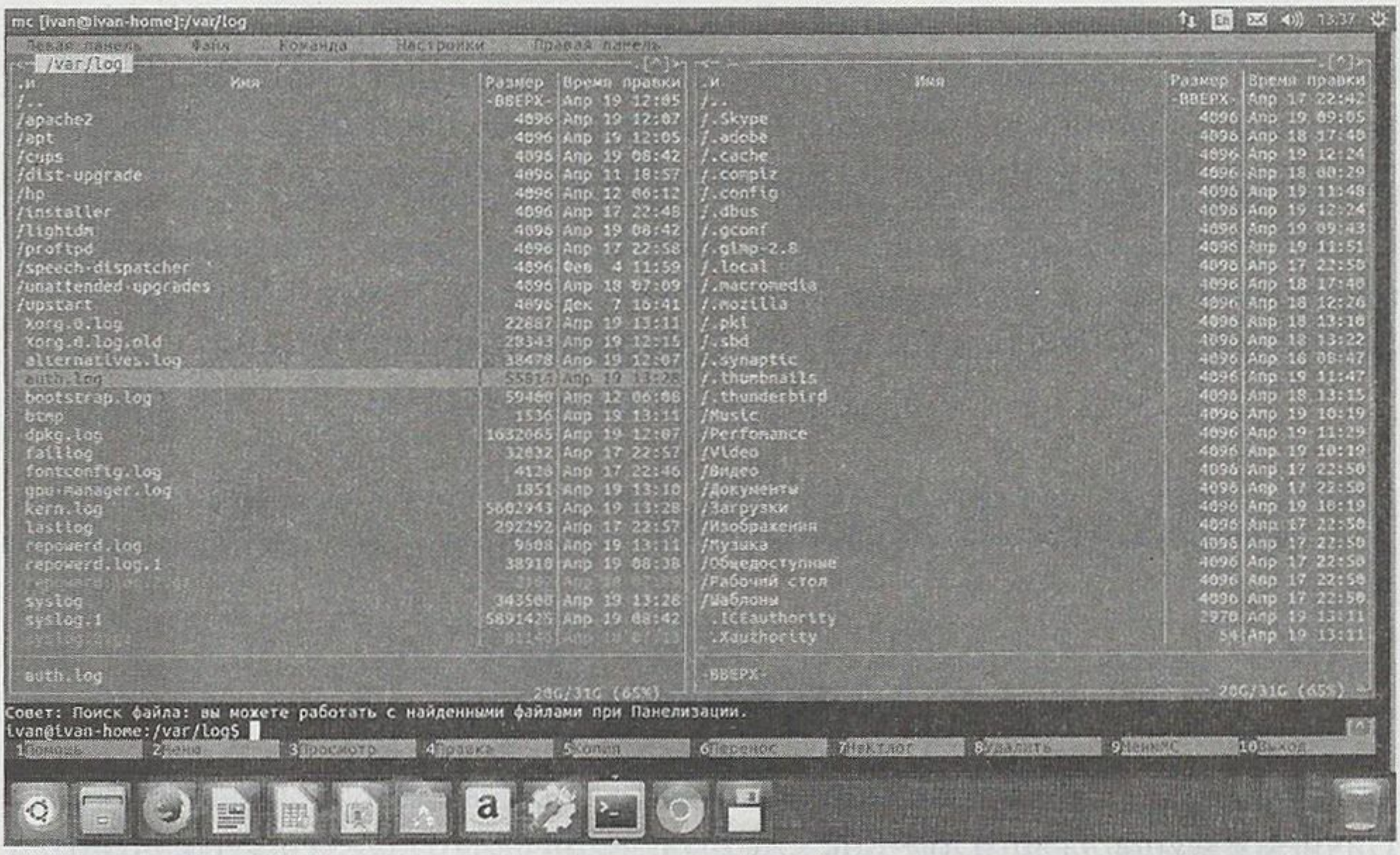


Рис. 17.1. Содержимое каталога /var/log

Файл auth.log очень важен. Он содержит информацию о входе пользователей в систему, о неправильно введенном пароле, о вводе команды sudo. Данный файл очень важный поэтому мы рассмотрим его отдельно.

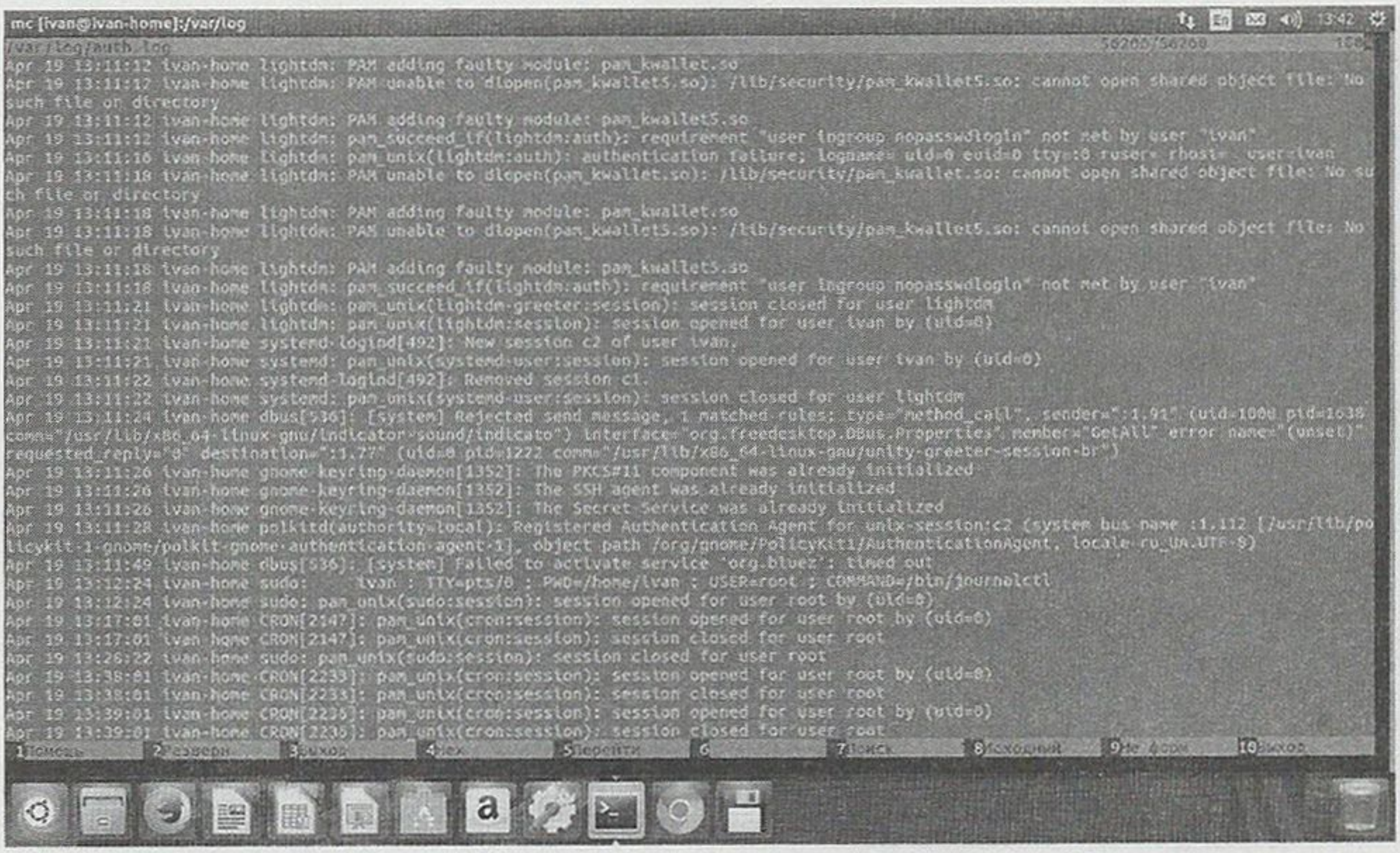


Рис. 17.2. Файл auth.log

Сообщения ядра записываются в kern.log. Поскольку при загрузке ядро генерирует много сообщений, размер у этого файла будет очень солидный. В этот журнал записывают сообщения и некоторые системные компоненты, например, NetworkManager. О нем мы также поговорим отдельно.

Самый важный системный журнал называется syslog. В него записывают сообщения многие системные компоненты - тот же NetworkManager, dbus, графическая оболочка и т.д. Мы его также рассмотрим отдельно.

17.2. Важные журналы

17.2.1. Журнал auth.log

Начнем с журнала auth.log. Как уже было отмечено ранее, в нем содержится информация о попытках входа в систему - как успешных, так и не очень, а также информация об использовании команды sudo. Давайте посмотрим подробнее.

17 апреля в 22:57 на компьютере ivan-home был создан (useradd) новый пользователь с именем ftp. Чуть позже был изменен его пароль. Очевидно, это произошло при установки FTP-сервера.

```
Apr 17 22:57:38 ivan-home useradd[3956]: new user: name=ftp,
UID=124, GID=65534, home=/srv/ftp, shell=/bin/false
Apr 17 22:57:38 ivan-home usermod[3961]: change user 'ftp'
password
```

Если не совсем очевидно, то пролистайте журнал и несколькими строчками выше будет примерно такая строка:

```
Apr 17 22:57:29 ivan-home sudo:      ivan : TTY=pts/0 ; PWD=/
home/ivan ; USER=root ; COMMAND=/usr/bin/apt install proftpd
```

Здесь пользователь ivan ввел команду sudo для получения root-прав для выполнения команды /usr/bin/apt install proftpd.

А вот так выглядит ошибка входа:

```
Apr 19 13:10:40 ivan-home lightdm: pam_unix(lightdm:auth):
authentication failure; logname= uid=0 euid=0 tty=:0 ruser=
rhost=  user=ivan
```

Нормальный вход в систему через графический менеджер выглядит так (19 апреля в 13:11 на компьютер ivan-home вошел пользователь ivan):

```
Apr 19 13:11:21 ivan-home lightdm: pam_unix(lightdm:session):
session opened for user ivan by (uid=0)
```

А вот пользователь ivan опять вошел в систему, но этот раз уже через FTP - с использованием сервиса profptd:

```
Apr 19 14:07:33 ivan-home proftpd: pam_unix(proftpd:session):
session opened for user ivan by (uid=0)
```

Отфильтровать журнал можно с помощью команды grep. Например, если вас интересуют неуспешные попытки входа, то можно ввести команду:

```
cat /var/log/auth.log | grep failure
```

Результат выполнения этой команды приведен на рис. 17.3. Как видно, 19 апреля было четыре неуспешных попытки входа в систему.

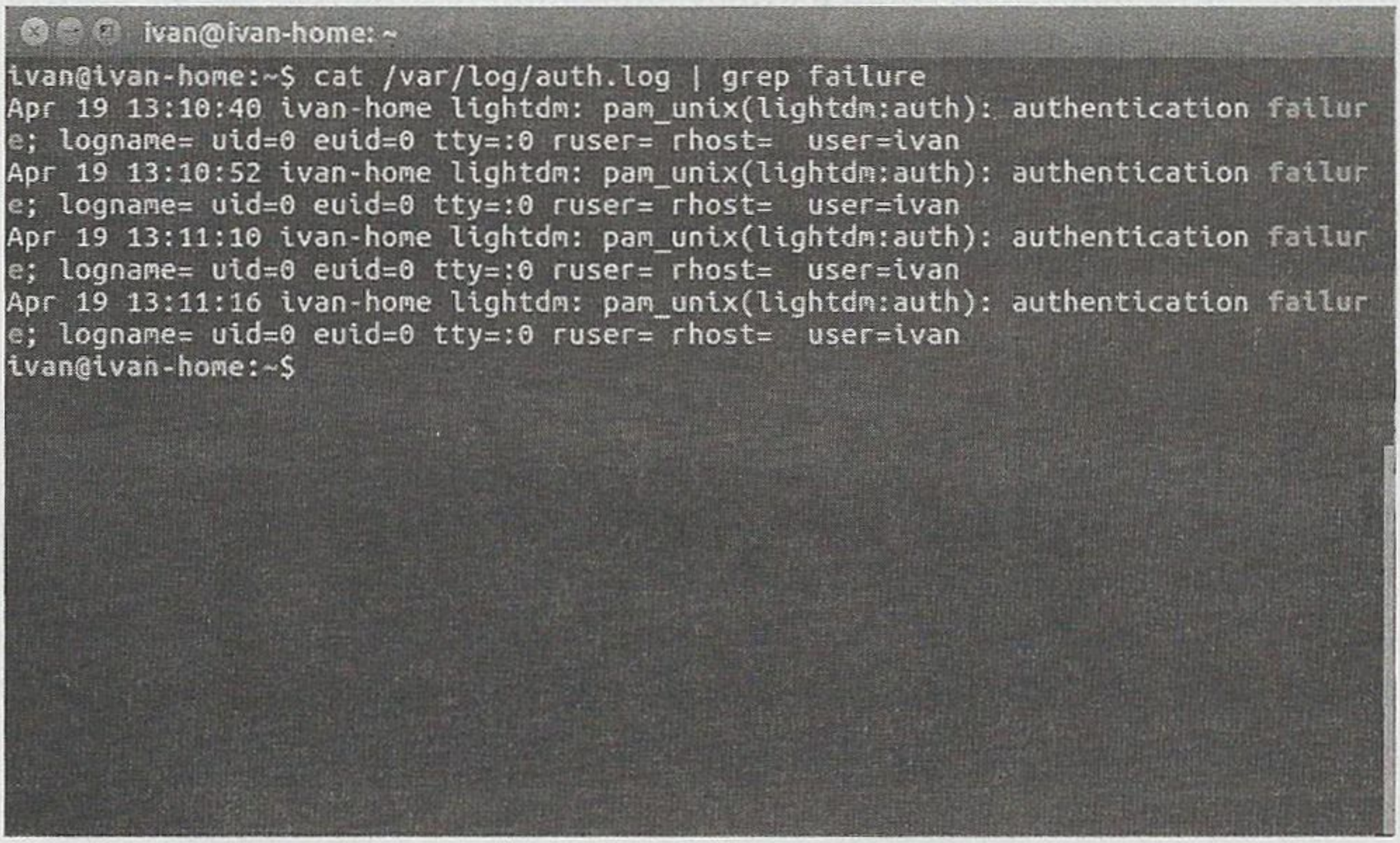


Рис. 17.3. Неуспешные попытки входа в систему

Если вас интересуют только FTP-сессии, отфильтруем журнал так (при использовании сервера proftpd:

```
cat /var/log/auth.log | grep proftpd:session
```

Все попытки входа по FTP за определенное число, например, за 19 апреля, можно вывести так:

```
cat /var/log/auth.log | grep proftpd:session | grep "Apr 19"
```

Можно вместо команды cat использовать команду tac - она обеспечит вывод строк журнала в обратном порядке, то есть сначала будут отображаться самые свежие записи. Тут как кому удобно - решайте сами.

17.2.2. Сообщения ядра - kern.log

Сообщения ядра, а также некоторых системных компонентов записываются kern.log. Поскольку при каждой загрузке ядро выводит очень много сообщений, размер файла kern.log будет немаленьким.

Просмотреть сообщения ядра, которые были выведены при последней загрузке можно командой dmesg | less. Команда dmesg выводит сообщения ядра без даты и времени записи этих сообщений - считается, что вы хотите получить информацию о последней загрузке системы. А команда less обеспечивает удобный просмотр этих сообщений.

По сообщениям ядра можно много чего узнать, например, версию ядра (в нашем случае - 4.10), а также параметры, которые были переданы ядру (в нашем случае - ro quiet splash):

```
Apr 17 22:50:34 ivan-home kernel: [    0.000000] Linux
version 4.10.0-19-generic (buildd@lcy01-13) (gcc version
6.3.0 20170321 (Ubuntu 6.3.0-10ubuntu1) ) #21-Ubuntu SMP
Thu Apr 6 17:04:57 UTC 2017 (Ubuntu 4.10.0-19.21-generic
4.10.8)
Apr 17 22:50:34 ivan-home kernel: [    0.000000] Command
line: BOOT_IMAGE=/boot/vmlinuz-4.10.0-19-generic
root=UUID=0a23abd6-d721-4def-a56c-c7a7457160ab ro quiet
splash
```

Можно также узнать тип установленного процессора:

```
Apr 17 22:50:34 ivan-home kernel: [    0.265061] smpboot:
CPU0: Intel(R) Celeron(R) CPU 1000M @ 1.80GHz (family: 0x6,
model: 0x3a, stepping: 0x9)
```

Конфигурацию, полученную сетевым адаптером от DHCP-сервера:

```
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6517] dhcp4 (ens33):   address 192.168.52.214
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6518] dhcp4 (ens33):   plen 24 (255.255.255.0)
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6519] dhcp4 (ens33):   gateway 192.168.52.2
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6519] dhcp4 (ens33):   server identifier
192.168.52.254
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6519] dhcp4 (ens33):   lease time 1800
Apr 17 23:02:24 ivan-home NetworkManager[595]:
<info>  [1492459344.6519] dhcp4 (ens33):   nameserver
'192.168.52.2'
Apr 17 23:02:24 ivan-home NetworkManager[595]:
<info>  [1492459344.6519] dhcp4 (ens33):   domain name
'localdomain'
Apr 17 23:02:24 ivan-home NetworkManager[595]: <info>
[1492459344.6519] dhcp4 (ens33):   wins '192.168.52.2'
```

Если у вас будет немного свободного времени, проанализируйте вывод dmesg - вы найдете много интересных фактов о вашей системе.

17.2.3. Системный журнал syslog

Этот журнал очень важен - в него записывают все, кому не лень, поэтому он содержит много полезной информации о работе системы. Давайте посмотрим, что в нем есть интересного.

Например, есть информация о том, что сетевому интерфейсу ens33 был назначен IP-адрес 192.168.52.214:

```
Apr 19 08:50:46 ivan-home dhclient[977]: DHCPREQUEST
of 192.168.52.214 on ens33 to 192.168.52.254 port 67
(xid=0x6f3e48b)
Apr 19 08:50:46 ivan-home NetworkManager[743]: <info>
[1492581046.6943] dhcp4 (ens33): address 192.168.52.214
```

Формат логов, кстати, такой же, как и везде:

```
Дата Время Имя_компьютера Название_компонента Сообщение
```

Перед сообщением может выводиться его тип, например, <info> для информационных сообщений.

Сообщения NetworkManager также дублируются в этот журнал. Удобно, что можно получить сразу всю информацию о конфигурации сети из одного файла:

```
# IP-адрес
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5864] dhcp4 (ens33): address 192.168.52.214
# Маска сети
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5865] dhcp4 (ens33): plen 24 (255.255.255.0)
# IP-адрес шлюза
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5865] dhcp4 (ens33): gateway 192.168.52.2
# ID сервера
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5866] dhcp4 (ens33): server identifier
192.168.52.254
```

```
# Время аренды IP-адреса (в секундах)
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5866] dhcp4 (ens33):   lease time 1800
# Сервер имен (DNS)
Apr 19 09:05:11 ivan-home NetworkManager[743]:
<info>  [1492581911.5866] dhcp4 (ens33):   nameserver
'192.168.52.2'
# Имя домена
Apr 19 09:05:11 ivan-home NetworkManager[743]:
<info>  [1492581911.5866] dhcp4 (ens33):   domain name
'localdomain'
# WINS-сервер
Apr 19 09:05:11 ivan-home NetworkManager[743]: <info>
[1492581911.5866] dhcp4 (ens33):   wins '192.168.52.2'
```

Информация о подключении внешнего USB накопителя выглядит так:

```
# Обнаружен USB-накопитель
Apr 19 14:42:53 ivan-home kernel: [ 5553.837012] usb-storage
1-1:1.0: USB Mass Storage device detected
Apr 19 14:42:53 ivan-home kernel: [ 5553.839580] scsi host3:
usb-storage 1-1:1.0
Apr 19 14:42:53 ivan-home kernel: [ 5553.839957] usbcore:
registered new interface driver usb-storage
Apr 19 14:42:53 ivan-home kernel: [ 5553.846180] usbcore:
registered new interface driver uas
# Производитель накопителя - StoreJet Transcend
Apr 19 14:42:54 ivan-home kernel: [ 5554.879053] scsi 3:0:0:0:
Direct-Access     StoreJet Transcend               PQ: 0 ANSI:
2 CCS
Apr 19 14:42:54 ivan-home kernel: [ 5554.882278] sd 3:0:0:0:
Attached scsi generic sg2 type 0
# Размер - 320 Гб, назначено имя устройства /dev/sdv
Apr 19 14:42:54 ivan-home kernel: [ 5554.910834] sd 3:0:0:0:
[sdb] 625142272 512-byte logical blocks: (320 GB/298 GiB)
```

```
Apr 19 14:42:54 ivan-home kernel: [ 5554.923832] sd 3:0:0:0:
[sdb] Write Protect is off
Apr 19 14:42:54 ivan-home kernel: [ 5554.923836] sd 3:0:0:0:
[sdb] Mode Sense: 28 00 00 00
Apr 19 14:42:54 ivan-home kernel: [ 5554.931760] sd 3:0:0:0:
[sdb] No Caching mode page found
Apr 19 14:42:54 ivan-home kernel: [ 5554.931769] sd 3:0:0:0:
[sdb] Assuming drive cache: write through
# Обнаружен один раздел - /dev/sdb1
Apr 19 14:42:54 ivan-home kernel: [ 5554.992950]  sdb: sdb1
Apr 19 14:42:54 ivan-home kernel: [ 5555.092817] sd 3:0:0:0:
[sdb] Attached SCSI disk
Apr 19 14:42:57 ivan-home colord-sane: io/hpmud/pp.c 627:
unable to read device-id ret=-1
# Опознана файловая система, NTFS, в действие вступил драйвер
ntfs-3g
Apr 19 14:43:02 ivan-home ntfs-3g[2577]: Version 2016.2.22AR.1
integrated FUSE 28
# Монтируется /dev/sdb1
Apr 19 14:43:02 ivan-home ntfs-3g[2577]: Mounted /dev/sdb1
(Read-Write, label "", NTFS 3.1)
# Опции монтирования
Apr 19 14:43:02 ivan-home ntfs-3g[2577]: Cmdline options: rw
,nodev,nosuid,uid=1000,gid=1000,uhelper=udisks2
Apr 19 14:43:02 ivan-home ntfs-3g[2577]: Mount
options: rw,nodev,nosuid,uhelper=udisks2,allow_
other,nonempty,relatime,default_permissions,fsname=/dev/
sdb1,blkdev,blksize=4096
Apr 19 14:43:02 ivan-home ntfs-3g[2577]: Global ownership
and permissions enforced, configuration type 7
# Точка монтирования - /media/ivan/BED2C6A4D2C6606F
Apr 19 14:43:02 ivan-home udisksd[1904]: Mounted /dev/sdb1
at /media/ivan/BED2C6A4D2C6606F on behalf of uid 1000
```

Исследуйте свой файл syslog и я более чем уверен, вы найдете в нем много интересного.

17.3. Ротация журналов

Файлы журналов бывают очень большими, особенно большими бывают журналы некоторых сетевых служб, например, веб-сервера. Работать с большими журналами неудобно - и администратору нужно больше времени, чтобы найти нужную информацию, и система тормозит, поскольку приходится работать с огромными файлами.

Именно поэтому во всех современных дистрибутивах используется служба ротации журналов logrotated. Чтобы logrotated выполнял ротацию того или иного журнала, об этом журнале нужно сообщить демону (службе) ротации. Для этого в каталоге /etc/logrotate.d нужно создать файл с произвольным (но понятным для вас) именем, например, в файле /etc/logrotate.d/apache2 содержатся правила ротации журнала веб-сервера Apache 2. Давайте рассмотрим их (лист. 17.1)

Листинг 17.1. Файл /etc/logrotate.d/apache2

```
# Путь поиска журналов. В нашем случае - все файлы с
расширением .log
# из каталога /var/log/apache2
/var/log/apache2/*.log {
# Ротация ежедневная
   daily
# Отсутствие файла не является ошибкой
   missingok
# Количество хранимых сжатых фрагментов
   rotate 14
# Фрагменты хранить в сжатом виде
   compress
# Не откладывать сжатие файла на следующий цикл
   delaycompress
# Не обрабатывать пустые файлы
   notifempty
# Права доступа к создаваемым фрагментам
   create 640 root adm
   sharedscripts
# Что нужно сделать после ротации - заставить Apache
перечитать
```

```
# файл конфигурации
   postrotate
                  if /etc/init.d/apache2 status > /dev/null ;
then \
                      /etc/init.d/apache2 reload > /dev/null; \
                  fi;
   endscript
# Что нужно сделать перед ротацией
   prerotate
      if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
         run-parts /etc/logrotate.d/httpd-prerotate; \
      fi; \
   endscript
}
```

17.4. Просмотр журналов с помощью journalctl

17.4.1. Что такое journald.service

Ранее протоколированием, то есть занесением сообщений системы и ее компонентов в журналы занимались службы syslogd и rsyslogd. Сейчас же протоколированием системы занимается сама система инициализации, то есть systemd, а точнее ее сервис systemd-journald.service. А для запроса системного лога используется утилита journalctl.

Новая система инициализации используется во всех современных дистрибутивах. Так, в Fedora она используется, начиная с версии 20, а в Ubuntu - начиная с версии 15.04. На данный момент даже дистрибутивы Debian и CentOS перешли на systemd-journald.service.

Некоторые особенности протоколирования в современных дистрибутивах:

- Все журналы по-прежнему хранятся в /var/log;
- Серверы (WWW, FTP и т.д.) могут создавать собственные каталоги/файлы журналов в каталоге /var/log;
- Для просмотра системных журналов используется утилита journalctl. Привычные файлы вроде /var/log/messages уже недоступны. Теоретически можно параллельно установить rsyslogd и он будет прекрасно работать в паре с journalcrl. Однако у journalctl гораздо больше возможностей, например, вы можете просмотреть логи текущей или предыдущей загрузки с помощью опции -b. Можно отфильтровывать сообщения по сервису, указав его в качестве параметра journalctl, при этом вам не нуж-

но знать имя журнала этого сервиса, не нужно использовать grep и т.д. Далее сами все увидите.

Первое, что нужно сделать - это установить ваш часовой пояс. Основным недостатком syslog/rsyslog было то, что эти демоны сохраняли записи в журнале без учета часового пояса и было непонятно, когда именно произошло то или иное событие. В journalctl этот недостаток был устранен: можно использовать как местное время, так и UTC.

Для выбора часового пояса используется команда:

```
$ timedatectl set-timezone <часовой пояс>
```

Вывести список часовых поясов можно командой:

```
$ timedatectl list-timezones
```

Отобразить информацию о текущем часовом поясе можно командой:

```
$ timedatectl status
```

17.4.2. Команда journalctl

Теперь приступим к просмотру журналов. Введите команду journalctl. Без параметров она выведет очень много записей, тем не менее, она же предлагает и возможности удобного просмотра этих записей. Вам больше не нужно использовать команды more или less.

Чтобы просмотреть системные журналы с момента текущей загрузки, используйте параметр -b:

```
$ journalctl -b
```

А вот просмотреть логи за предыдущую загрузку немного сложнее. Просмотреть список предыдущих загрузок можно командой (для каждой загрузки выводится ее идентификатор и дата):

```
$ journalctl --list-boots
```

Чтобы просмотреть журнал предыдущей загрузки, нужно указать ее номер, например (0 - текущая загрузка, -1 - предыдущая и т.д.):

```
$ journalctl -b -1
```

```
ivan@ivan-home: ~
-- Logs begin at Wed 2017-04-19 13:10:24 EEST, end at Wed 2017-04-19 15:17:55 EEST. --
Апр 19 13:10:24 ivan-home kernel: Linux version 4.10.0-19-generic (buildd@lcy01-13) (gcc version 6.3.0 20170321 (U
Апр 19 13:10:24 ivan-home kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.10.0-19-generic root=UUID=0a23abd6-d721
Апр 19 13:10:24 ivan-home kernel: KERNEL supported cpus:
Апр 19 13:10:24 ivan-home kernel:   Intel GenuineIntel
Апр 19 13:10:24 ivan-home kernel:   AMD AuthenticAMD
Апр 19 13:10:24 ivan-home kernel:   Centaur CentaurHauls
Апр 19 13:10:24 ivan-home kernel: Disabled fast string operations
Апр 19 13:10:24 ivan-home kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
Апр 19 13:10:24 ivan-home kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
Апр 19 13:10:24 ivan-home kernel: x86/fpu: Enabled xstate features 0x3, context size is 576 bytes, using 'standard
Апр 19 13:10:24 ivan-home kernel: e820: BIOS-provided physical RAM map:
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009efff] usable
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x000000000009f000-0x000000000009ffff] reserved
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x00000000000ca000-0x00000000000cbfff] reserved
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x00000000000dc000-0x00000000000fffff] reserved
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x0000000000100000-0x000000007fedffff] usable
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x000000007fee0000-0x000000007fefefff] ACPI data
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x000000007feff000-0x000000007fefffff] ACPI NVS
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x000000007ff00000-0x000000007fffffff] usable
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x00000000e0000000-0x00000000efffffff] reserved
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec0ffff] reserved
Апр 19 13:10:24 ivan-home kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved
lines 1-23
```

Рис. 17.4. Команда journalctl

17.4.3. Фильтр журналов по дате

В большинстве случаев удобно отфильтровывать сообщения по дате. Для этого используются опции -since и -until. Так, для просмотра логов, начиная с 19.04.17 5:00 введите команду:

```
$ journalctl --since «2017-04-19 05:00:00»
```

Если вы указали опцию --since, но не указали дату, будет использоваться текущая. Если указана дата, но не указано время, будет использоваться время 00:00:00.

Еще несколько примеров:

```
$ journalctl ---since yesterday
$ journalctl --since 05:00 --until now
$ journalctl --since 05:00 --until "2 hour ago"
```

Первая команда показывает журналы, начиная со вчерашнего дня и по текущий момент. Вторая команда отображает журналы, начиная с 5 утра и по текущий момент. Третья - начиная с 5 утра и до двух предыдущих часов (то есть, если сейчас 18:00, то до 16:00).

17.4.4. Фильтр по службе

Выполнить фильтрацию можно и по определенному сервису (когда нужно просмотреть не все журналы, а только определенной службы), например:

```
$ journalctl -u proftpd.service
```

Тип фильтрации можно комбинировать, например, следующая команда выводит журналы profptd, начиная со вчерашнего дня:

```
$ journalctl -u profptd.service --since today
```

```
ivan@ivan-home: ~
ivan@ivan-home:~$ sudo journalctl -u proftpd.service
-- Logs begin at Wed 2017-04-19 13:10:24 EEST, end at Wed 2017-04-19 15:23:49 EEST. --
Anp 19 13:10:40 ivan-home systemd[1]: Starting LSB: Starts ProFTPD daemon...
Anp 19 13:10:40 ivan-home proftpd[1241]:  * Starting ftp server proftpd
Anp 19 13:10:40 ivan-home proftpd[1241]:    ...done.
Anp 19 13:10:40 ivan-home systemd[1]: Started LSB: Starts ProFTPD daemon.
Anp 19 14:03:10 ivan-home proftpd[2337]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
Anp 19 14:07:28 ivan-home proftpd[2337]: pam_unix(proftpd:session): session closed for user ivan
Anp 19 14:07:33 ivan-home proftpd[2355]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
Anp 19 14:22:33 ivan-home proftpd[2451]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
Anp 19 14:34:10 ivan-home proftpd[2470]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
Anp 19 15:03:30 ivan-home proftpd[2625]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
Anp 19 15:18:24 ivan-home proftpd[2709]: pam_unix(proftpd:session): session opened for user ivan by (uid=0)
ivan@ivan-home:~$
```

Рис. 17.5. Пример фильтра по сервису

17.4.5. Фильтр по уровню ошибки

Возможно отфильтровать сообщения по уровню ошибки:

- 0 – EMERG (система неработоспособна);
- 1 – ALERT (требуется немедленное вмешательство);
- 2 – CRIT (критическое состояние);
- 3 – ERR (ошибка);
- 4 – WARNING (предупреждение);
- 5 – NOTICE (просто обратите внимание);
- 6 – INFO (информационное сообщение);
- 7 –DEBUG (отложенная печать).

Пример:

```
$ journalctl -p err -b
```

Команда выводит все ошибки при текущей загрузке.

Вместо заключения

Linux - очень универсальная и гибкая операционная система, что и было показано в этой книге. Надеюсь, она пролила свет на многие моменты, связанные с повседневным использованием этой замечательной операционной системы.

Книжный магазин

издательства «Наука и Техника»
приглашает за покупками

Предлагаем широкий ассортимент технической литературы ведущих издательств (более 2000 наименований):

- Компьютерная литература
- Радиоэлектроника
- Телекоммуникации и связь
- Транспорт, строительство
- Научно-популярная медицина, педагогика, психология

Чем привлекателен наш магазин:

- низкие цены;
- ежедневное пополнение ассортимента;
- поиск книг под заказ;
- обслуживание за наличный и безналичный расчет;
- гибкая система скидок;
- комплектование библиотек;
- обеспечение школ учебниками по информатике;
- возможна доставка.

Наш адрес: г. Санкт-Петербург
пр. Обуховской Обороны д. 107
ст. метро Елизаровская

Справки о наличии книг по тел. 412-70-26

E-mail: admin@nit.com.ru
(рассылка ассортиментного прайс-листа по запросу)

Мы работаем с 10 до 19 часов без обеда и выходных
(в субботу и воскресенье до 18 час)

Издательство «Наука и Техника»

Книги по компьютерным технологиям, медицине, радиоэлектронике

Уважаемые читатели!

Книги издательства «Наука и Техника» вы можете:

- **заказать в нашем интернет-магазине**

  www.nit.com.ru (более 100 пунктов выдачи на территории РФ)

- **приобрести в магазине издательства по адресу:**

  Санкт-Петербург, пр. Обуховской обороны, д.107 (ежедневно с 10.00 до 18.30),
  тел. **(812) 412-70-26**

- **приобрести в Москве:**

«Новый книжный» Сеть магазинов	тел. (495) 937-85-81, (499) 177-22-11
ТД «БИБЛИО-ГЛОБУС»	ул. Мясницкая, д. 6/3, стр. 1, ст. М «Лубянка», тел. (495) 781-19-00, 624-46-80
Московский Дом Книги, «ДК на Новом Арбате»	ул.Новый Арбат, 8, ст. М «Арбатская», тел. (495) 789-35-91
Московский Дом Книги, «Дом технической книги»	Ленинский пр., д.40, ст. М «Ленинский пр.», тел. (499) 137-60-19
Московский Дом Книги, «Дом медицинской книги»	Комсомольский пр.,д. 25, ст. М«Фрунзенская» тел. (499) 245-39-27
Дом книги «Молодая гвардия»	ул. Б. Полянка, д. 28, стр. 1, ст. М «Полянка», тел. (499) 238-50-01

- **приобрести в Санкт-Петербурге:**

Санкт-Петербургский Дом Книги	Невский пр. 28, тел. (812) 448-23-57
Буквоед. Сеть магазинов	тел. (812) 601-0-601

- **приобрести в регионах России:**

г. Воронеж, «Амиталь» Сеть магазинов	тел. (473) 224-24-90
г. Екатеринбург, «Дом книги» Сеть магазинов	тел. (343) 289-40-45
г. Нижний Новгород, «Дом книги» Сеть магазинов	тел. (831) 246-22-92
г. Владивосток, «Дом книги» Сеть магазинов	тел. (423) 263-10-54
г. Иркутск, «Продалить» Сеть магазинов	тел. (395) 298-88-82
г. Омск, «Техническая книга» ул. Пушкина, д.101	тел. (381) 230-13-64

Мы рады сотрудничеству с Вами!

Группа подготовки издания:

Зав. редакцией компьютерной литературы: *М. В. Финков*

Редактор: *Е. В. Финков*

Корректор: *А. В. Громова*

ООО «Наука и Техника»

Лицензия №000350 от 23 декабря 1999 года.

198097, г. Санкт-Петербург, ул. Маршала Говорова, д. 29.

Подписано в печать 28.04.2017. Формат 70x100 1/16.

Бумага газетная. Печать офсетная. Объем 22 п. л.

Тираж 1400 . Заказ 4127.

Отпечатано с готовых файлов заказчика
в АО «Первая Образцовая типография»
филиал «УЛЬЯНОВСКИЙ ДОМ ПЕЧАТИ»
432980, г. Ульяновск, ул. Гончарова, 14.

Донцов В. П., Сафин И. В.

Linux на примерах

Практический курс по работе в Linux

Данная книга является отличным практическим руководством по работе в операционной системе Linux и ее администрированию. На большом количестве практических примеров показано выполнение огромного количества действий и задач, которые могут понадобиться пользователю Linux.

Книга может использоваться как практический справочник и как самоучитель для начинающих. Изложение ведется с учетом самых разных дистрибутивов Linux. Лучший выбор для эффективного изучения и использования Linux.

www.nit.com.ru

ISBN 978-5-94387-742-1

978-5-94387-7421

Издательство "Наука и Техника"
г. Санкт-Петербург

Для заказа книг:
(812) 412-70-26
e-mail: nitmail@nit.com.ru
www.nit.com.ru