

<b>I. ОПЕРАЦИОННАЯ СИСТЕМА (LINUX + KDE)</b> .....	<b>5</b>
1. ИСТОРИЯ LINUX И ПСПО.....	5
1.1. История UNIX, Linux .....	5
1.2. Знакомство с линейкой продуктов Linux .....	6
1.3. Свободное программное обеспечение.....	9
1.4. История ПСПО .....	10
1.5. Основные понятия и принцип работы ОС .....	13
2. РАБОТА В KDE. РАБОТА С ПСПО .....	14
2.1. Основы работы с ПК .....	14
2.1.1. Персональный компьютер .....	14
2.1.2. Аппаратное обеспечение, программное обеспечение .....	15
2.1.3. Последовательность включения ПК.....	23
2.2. Пользовательский интерфейс.....	24
2.2.1. Запуск графического интерфейса пользователя, завершение работы.....	24
2.2.2. Обзор графической среды.....	29
2.2.2.1. Оконная система X и её реализации.....	29
2.2.2.2. Менеджеры окон .....	30
2.2.2.3. Интегрированные графические среды .....	33
2.2.2.4. KDE.....	36
2.2.3. Рабочий стол KDE, панель задач, главное меню, объекты на рабочем столе, контекстное меню.....	40
2.2.4. Работа с окнами: перемещение, изменение размеров, понятие активного окна, переход из одного окна в другое.....	44
2.2.5. Справочная система .....	47
2.3. Работа с файловой системой .....	50
2.3.1. Организация данных .....	50
2.3.2. Иерархическая структура файлов и папок.....	54
2.3.3. Работа с файлами и папками: создание файлов, папок, копирование, перемещение, переименование, удаление, восстановление, поиск.....	58
2.3.4. Популярные файл-менеджеры .....	72
2.3.5. Форматирование дискет.....	74
2.3.6. Запись CD-R и CD-RW .....	76
2.4. Стандартные программы (обзор) .....	80
2.5. Настройка настольной системы с помощью графических утилит .....	98
2.5.1. Настройка меню графической среды KDE .....	98

2.5.2. Запуск программ.....	101
2.5.3. Панель управления KDE.....	101
2.5.4. Получение информации о системе .....	106
3. ИНСТАЛЛЯЦИЯ LINUX .....	108
3.1. Введение.....	108
3.2. Выбор программного обеспечения для начальной установки.....	109
3.3. Установка дистрибутива.....	110
3.4. Точки монтирования и их использование.....	111
3.5. Файловые системы .....	113
3.6.       Дополнительная настройка, конфигурирование и проверка оборудования в процессе установки ОС.....	114
3.6.1. Начало установки: загрузка системы .....	114
3.6.2. Другие источники установки .....	116
3.7. Конфигурация системного загрузчика.....	137
3.8. Начальные установки (сеть, пользователи, сообщения диагностики).....	138
4. УСТАНОВКА И НАСТРОЙКА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .....	144
4.1. Установка пакетов.....	146
4.2. Удаление пакетов .....	148
4.3. Установка доступных обновлений .....	148
4.4. Обновление всей системы .....	149
5. АДМИНИСТРИРОВАНИЕ LINUX.....	150
5.1.       Начало работы с операционной системой Linux.....	150
5.1.1 Варианты входа пользователя в систему .....	150
5.1.2. Основы работы с командными интерпретаторами .....	152
5.1.3. Справочные системы (man, info, HowTo, FAQs).....	154
5.1.4. Обзор основных команд интерпретатора.....	159
5.2.       Понятие системного администрирования.....	166
5.3.       Система авторизации и управление правами доступа пользователей.....	167
5.3.1       Средства аутентификации/авторизации пользователей Linux. Учетные записи пользователей и группы .....	167
5.3.2. Настройки операционной системы и среды пользователя. Утилиты администрирования учетных записей пользователей.....	168
5.3.3.       Управление правами доступа к файлам .....	173
5.4.       Процесс загрузки системы и управление процессами.....	178

5.4.1. Процесс загрузки ОС.....	178
5.4.2. Стартовый виртуальный диск .....	179
5.4.3.       Запуск и остановка системных служб .....	184
5.4.4. Основные характеристики процессов и управление процессами.....	185
5.4.5.       Регистрация событий в системе .....	187
5.4.6.       Обзор планировщика выполнения заданий cron .....	188
5.5.       Структура ядра. Добавление/удаление устройств .....	189
5.5.1       Внутреннее устройство ядра Linux.....	189
5.6. Система печати. ....	194
5.6.1. Организация системы печати ОС Linux. Возможности систем печати .....	194
5.6.2. Настройка системы печати CUPS .....	196
5.7. Файловая система Linux - возможности и использование. ....	199
5.7.1. Типы файловых систем. Особенности и преимущества использования файловых систем. Структура файловой системы .....	199
5.7.2. Типы файлов в файловой системе .....	202
5.7.3. Монтирование файловых систем различных типов. Создание и проверка файловых систем различных типов .....	205
5.7.4. Настройка квот .....	209
5.8. Сетевые ресурсы. Клиентская часть.....	212
5.8.1. Подключение рабочей станции в существующую сеть.....	212
5.8.2. Удалённый доступа к компьютеру. Работа с программами удалённого доступа (telnet, ssh, ftp).....	215
5.8.3. Настройка клиента NFS .....	217
5.9.       Настройка серверной части Linux.....	218
5.9.1.       Сетевые службы Linux – установка, настройка и администрирование стандартных сетевых служб DNS, Apache, SMB, DHCP.....	218
5.9.2. FTP     227	
5.10.       Дополнительные возможности администрирования Linux.....	228
5.10.1. Резервное копирование и восстановление .....	228
5.10.2. Анализ ошибок и устранение неисправностей.....	231
6.   РАБОТА В СЕТИ .....	236
6.1. Локальные сети.....	236
6.1.1. Топология типа «звезда» .....	237
6.1.2 Кольцевая топология.....	238
6.1.3 Шинная топология .....	239

6.1.4 Древовидная топология .....	240
6.2. Глобальная, мировая сеть .....	241
6.3. Адрес компьютера. Имя компьютера .....	244
6.4. Процедура входа в сеть. Общий диск (дисковое пространство) .....	247
6.5. Общий принтер (сетевой принтер) .....	250
6.6. Информационные источники (веб-сайты) .....	252
6.7. Хранилища файлов (FTP узлы) .....	252
<b>II. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ СЖАТИЯ И АРХИВИРОВАНИЯ</b>	
<b>ФАЙЛОВ (ARK) .....</b>	<b>253</b>
1. ВВЕДЕНИЕ .....	253
2. СЖАТИЕ И АРХИВИРОВАНИЕ ФАЙЛОВ С ИСПОЛЬЗОВАНИЕМ ARK .....	254
2.1. Открытие архивов .....	254
2.2. Работа с файлами архива .....	255
2.3. Создание архивов и добавление файлов .....	256
2.4. Настройка Ark .....	257
2.5. Работа с архивами из файлового менеджера .....	258

# ***I. Операционная система (Linux + KDE)***

## **1. История Linux и ПСПО**

### **1.1. История UNIX, Linux**

Принято начинать историю создания Linux начинать с 1957 с создания BESYS. Но мы начнем ее намного раньше. В США в 1890 году был акт Шермана, первый шаг в создании антимонопольного законодательства. За ним последовали другие законы, акты которые обеспечивали конкуренцию и боролись с монополиями. В 50 годах прошлого века корпорация AT&T (глобальный монополист на рынке телефонной связи США) согласно различным судебным постановлениям не имела права продавать программное обеспечение и обязана была предоставлять его всем желающим.

В середине 50-х гг. исследовательский отдел корпорации Bell System (AT&T) приобрел для своего компьютерного центра несколько новых компьютеров. Компьютеры, купленные у IBM, предназначались для проведения разработок в пока еще мало изученной компьютерной области. Но когда ученые Bell освоились на установленных мейнфреймах, стало ясно, что идущее в поставке программное обеспечение не подходит для полноценной исследовательской работы. А из-за отсутствия операционной системы, все приходилось делать вручную, что отнимало много времени и сил. Так как большинство сотрудников отдела состояло из опытных программистов, они приняли решение разработать ОС своими силами.

Общими усилиями коллектива разработка системы BESYS заняла менее года и в 1957 г. она была установлена на всех машинах компьютерного центра. Авторы BESYS не собирались распространять за пределы Bell – программа предназначалась исключительно для собственных нужд. Но, когда некоторые исследовательские институты проявили интерес к разработке, сотрудники крупнейшей телефонной компании выслали всем желающим копии на магнитных лентах.

В 1964 г. Bell Labs приобрела новое дорогостоящее оборудование, включая более мощные компьютеры, установленные в исследовательском отделе. Программисты компании снова столкнулись со старой проблемой. BESYS была написана под конкретную платформу и не работала на новых машинах.

Идея создать Multics (Multiplexed Information and Computing Service) – операционную систему с поддержкой CTSS – пришла профессору Джеку Дэннису из Массачусетского Технологического Института. Ее подхватили студенты и в 1963 г. вместе они разработали спецификации будущей ОС.

В апреле 1969 г., команда из Bell заявила о своем решении выйти из состава разработчиков ОС. Через полгода, в октябре 1969 г., сильно сокращенная и явно недоработанная Multics была представлена общественности. Только в середине 1970 г. первая система с разделением времени обрела стабильную форму.

Некоторые программисты из числа Bell-разработчиков Multics не оставили своего желания создать гибкую систему, пригодную для серьезного программирования. Среди них были Кен Томпсон, Деннис Речи, Джои Осанна и Рад Кеннедей.

В начале 1970 г., система могла полностью функционировать самостоятельно. Название UNICS (UNiplexed Information and Computing Service) подсказал Брайан Керниган – работник компании, все это время с интересом следивший за проектом. А через несколько месяцев ОС стала более известна как UNIX.

После того, она стала быстро набирать популярность. Этому во многом способствовало ее умение легко адаптироваться к самым разным компьютерным платформам. В 1973 г. UNIX была практически полностью переписана на языке C, что сделало ее еще привлекательнее. В большинстве исследовательских институтов эта ОС стала стандартом де-факто, причем многие старались как-то улучшить ее возможности. В результате, с 70-х гг. по 90-е вышло множество UNIX клонов (FreeBSD, OpenBSD, NetBSD, Ultrix, Xenix, Irix, HP-UX, Solaris, Unixware и т.д.), среди которых были как коммерческие, так и не коммерческие. Но ни одна из этих систем не получила такой популярности и такого признания, как Linux.

Линус Торвальдс создал ядро и предоставил добровольцам исходные коды ядра по лицензии GPL. Поверх этого ядра стали использоваться программы, разработанные при участии Фонда свободного программного обеспечения. Так в 1991 году родилась Linux. В октябре 1991 г. вышла Linux 0.02, а в ноябре – 0.03. Однако по настоящему завоевывать популярность Linux начала, когда в конце ноября стала полностью автономной. К тому времени, как вышла версия Linux 1.0, о системе уже знал весь мир.

## **1.2. Знакомство с линейкой продуктов Linux**

Так как Linux разрабатывается независимыми группами (ядро разрабатывается одной группой) возникает проблема с выбором версии Linux. На текущий момент число версий сборок Linux перевалило за 400. В их числе как коммерческие реализации, научные и прочие версии. Вопросы совместимости между ними стали не простой задачей. Причем программы от одной версии Linux могут не работать на другой. Идет время к движению Linux примыкают крупные корпорации IBM, Oracle. Другие открывают исходные коды некоторых продуктов SUN, Microsystems открыла Open Office - аналог закрытого решения от Microsoft. Развитие Linux спонсируется крупными компаниями.

Коммерческие версии Linux уверенно шагают на корпоративный рынок, разрабатывают учебные курсы, системы сертификации и партнерство с учебными заведениями. Не коммерческие версии поддерживаются сообществом энтузиастов и производителями коммерческих версий (служат полигоном новых технологий и бесплатными тест площадками, например REDHAT-Fedora). Есть также и научные версии, например, Scientific Linux, пересобранная версия из исходных кодов REDHAT с добавлением научного программного обеспечения (используется в CERN,FERMILAB,ДУБНА и прочее). Не обходимо отметить, что не только Linux свободна и доступна. Есть еще не FreeBSD, OpenBsd, OpenSolaris, OpenSuse и другие, которые тоже используются в компьютерном мире место Linux.

Общеизвестно, что Linux - бесплатен. Ядро Linux, и немалое количество модулей к этой системе – ПО с открытым исходным кодом; но некоторые дистрибутивы, такие как коммерческий Red Hat Enterprise Linux, требуют оплаты.

При выборе дистрибутива в первую очередь рассматриваются продукты коммерческих производителей, имеющих хорошую репутацию. Эти производители предлагают дорогостоящие контракты с достаточно высоким уровнем поддержки, официальной сертификацией специалистов. Поставщики коммерческих прикладных программ (таких, как Oracle или SAP) чаще всего официально обеспечивают их работу именно с конкретным дистрибутивом. Вокруг этих дистрибутивов существуют достаточно серьезные технические сообщества. Поэтому их можно применять и без заключения контракта с производителем.

Рассмотрим некоторые популярные ОС Linux.

**DEBIAN GNU/Linux.** Проект Debian запущен в 1993 году открытым сообществом. Один из самых надёжных дистрибутивов в мире, за которым не стоит коммерческая компания. Особенность – весьма совершенная система управления пакетами (dpkg и apt). Другие дистрибутивы, многое перенимают у Debian. Официальные “стабильные” выпуски Debian происходят достаточно редко. Так, последняя версия Debian 3.1 (условное название sarge) выпущена в июне 2005 года, а предыдущая (3.0, woody) – в июле 2002 года. Ошибки (особенно связанные с безопасностью системы) в Debian исправляются весьма оперативно благодаря большому количеству разработчиков– участников сообщества. В “стабильных” выпусках зачастую содержатся немного устаревшие версии приложений. Постоянные обновления безопасности гарантированы именно для “стабильного” дистрибутива. Существуют дистрибутивы, основанные на Debian, но тем или иным образом модифицированные. Например, Ubuntu.

**RED HAT.** Компания Red Hat – известнейший поставщик дистрибутивов Linux. Эта компания предлагала свои продукты с 1994 года, и завоевала ведущие позиции на корпоративном рынке США. Коммерческий продукт – это линейка дистрибутивов Red Hat Enterprise Linux (RHEL). Среди них есть варианты для различных видов серверов и рабочих станций. Дистрибутивы предлагаются вместе с контрактами на поддержку. Гарантируются критичные обновления в течение нескольких лет. Свободно эти дистрибутивы не распространяются. Но их исходные тексты доступны Сборки идентичных версий, распространяемых свободно (Scientific Linux - русифицированную версию). Red Hat Linux поддерживает разработку дистрибутивов серии Fedora, в которой принимает участие техническое сообщество.

**NOVELL.** Направление деятельности Novell, имеющее отношение к Linux, началось с двух приобретений в 2003 г. – небольшой компании Ximian, которая вела открытые разработки, и германского разработчика дистрибутива SUSE Linux. Novell предлагает ряд дистрибутивов Linux, от высокопроизводительной системы для серверов Novell Open Enterprise Server до решения для рабочих станций Novell Linux Desktop и универсального, бесплатного дистрибутива SUSE Linux Professional.

**ASP LINUX.** Дистрибутивы ASP разработаны на базе проекта Fedora (открытая разработка Red Hat). Существует универсальный дистрибутив ASPLinux и специализированный вариант для серверов ASPLinux Server. Отличительная особенность – базовые услуги технической поддержки. ASP Linux требует минимального администрирования. Достаточно всего лишь запустить, правильно сконфигурировать – после этого он сможет работать без перезагрузки и обслуживания годы.

**ALT LINUX.** Реализовано несколько систем, по различному назначению: системы для домашних и офисных компьютеров - ALT Linux 4.1 Desktop и ALT Linux 4.0 Lite, сертифицированные системы - ALT Linux 4.0 Desktop Professional и ALT Linux 4.0 Server Edition, серверные системы - ALT Linux 4.0 Server и ALT Linux 4.0 Office Server. Именно, на базе дистрибутивов Alt Linux разрабатывался Пакет свободного программного обеспечения, о котором скажем далее.

**MANDRIVA (MANDRAKE).** Известная французская фирма – производитель дистрибутивов Mandrake Linux недавно объединилась с бразильской Conectiva, и сменила название на Mandriva. Претендует на вхождение в круг коммерческих производителей Linux. Известны в первую очередь лёгкостью установки и управления, во многих случаях они наиболее корректно определяют подключённое к компьютеру оборудование.

Несколько зарубежных фирм активно продвигают свои дистрибутивы Linux, предназначенные для «особо лёгкого» использования на настольных компьютерах.



Таковы, например, Lycoris и Linspire (прежнее название – Lindows). Однако без поддержки со стороны фирм-производителей (которые в России не представлены) никакого реального смысла в применении данных дистрибутивов нет.

Некоторые дистрибутивы предназначены для максимальной гибкой настройки со стороны системного администратора. Как правило, в них нет жёсткой системы управления пакетами, как в остальных дистрибутивах. Специалист, используя такую систему, нередко добивается максимальной производительности. К сожалению, может пострадать надёжность. Кроме того, система может быть достаточно серьёзно изменена, и смысл этих изменений знает только тот, кто их производил. Если же он по какой-то причине окажется вне досягаемости, другому специалисту (даже очень квалифицированному) будет сложно разобраться в системе. Кроме того, не слишком опытные специалисты иногда переоценивают свои возможности и настаивают на применении подобных дистрибутивов – а затем не справляются с большим количеством работы, которая требуется для поддержки и обновления ПО на серверах. К данному классу относится Slackware Linux, а также дистрибутивы, основанные на сборке ПО из исходных кодов (наиболее распространённый из них – Gentoo Linux). Получить информацию о других дистрибутивах Linux можно, перейдя по ссылке [http://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA\\_%D0%B4%D0%B8%D1%81%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D0%B8%D0%B2%D0%BE%D0%B2\\_Linux](http://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B4%D0%B8%D1%81%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D0%B8%D0%B2%D0%BE%D0%B2_Linux).

### **1.3. Свободное программное обеспечение**

#### **Типы программного обеспечения**

Все программное обеспечение на сегодняшний день можно разделить на две большие категории:

- **Открытое** — программное обеспечение, исходные тексты которого можно свободно получить, изучать и модифицировать. Разработчики открытого ПО обладают авторскими правами на него, но при этом все желающие могут свободно его использовать, распространять, копировать и модифицировать без каких-либо лицензионных отчислений.
- **Проприетарное** — программное обеспечение, разработанное некоторой (коммерческой) компанией, которая обладает на него авторскими правами и правом собственности, и распространяемое на коммерческой основе. Исходные тексты такого ПО недоступны.

#### **Определение свободного программного обеспечения**

В соответствии с «Концепцией развития разработки и использования свободного программного обеспечения в Российской Федерации», принятой Мининформсвязи России 12 марта 2008 года (далее по тексту — Концепция Мининформсвязи), под свободным программным обеспечением (СПО) понимается разновидность программ для ЭВМ, лицензионным договором (свободная лицензия) на право использования которых, предоставляются следующие права:

1. Использовать программу для ЭВМ в любых, не запрещенных законом целях.
2. Получать доступ к исходным текстам программы как в целях изучения, адаптации и использования, так и в целях переработки программы для ЭВМ.
3. Распространять программу (бесплатно или за плату, по своему усмотрению).
4. Вносить изменения в программу для ЭВМ (перерабатывать) и распространять экземпляры измененной (переработанной) программы с учетом возможных требований наследования лицензии.

**Свободные лицензии** — особый вид лицензий, предназначенный для обеспечения юридической защиты прав («свобод») пользователя (общественности) на неограниченные *воспроизведение, изучение, распространение и изменение* (модификацию или совершенствование) различных продуктов интеллектуальной деятельности. Примерами свободных лицензий являются: GNU GPL, GNU LGPL, BSD, GNU FDL и т.д.

Дополнительную информацию о СПО можно получить <http://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5>

## 1.4. История ПСПО

Рассмотрим исторический аспект ПСПО. В октябре 2007 года вышло Распоряжение Председателя Правительства РФ № 1447-р., согласно которому Министерству образования и науки и Министерству связи предписывалось:

1. закупить лицензию на 3 года на проприетарное программное обеспечение для всех школ;
2. обеспечить разработку и внедрение в школы свободного программного обеспечения по перечню программ, входящих в стандартный (базовый) пакет программного обеспечения.

Закупка трёхлетних лицензий на проприетарное ПО осуществлена для того, чтобы снять накал проблем, связанных с пиратским ПО, и за это время подготовить систему образования к переходу на СПО.

В ноябре 2007 года федеральное агентство по образованию провело открытый конкурс на право заключения государственного контракта по созданию пакета свободного программного обеспечения (ПСПО) и пилотному апробированию этого пакета программ в образовательных учреждениях (ОУ) Российской Федерации. На основе требований стандартного базового пакета программного обеспечения был разработан пакет свободного программного обеспечения (ПСПО) для нужд ОУ, не требующий лицензионных отчислений.

Был проведен тендер, который выиграла группа компаний «РБК-ЦЕНТР»-Армада, и в результате на настоящий момент мы имеем 2 независимые ветки дистрибутивов Linux (Alt-Linux, HayLinux – фактический Scientific Linux). Эти решения были протестированы в нескольких пилотных регионах (Республика Татарстан, Пермский край и Томская область), потом частично доработаны.

В рамках апробации ПСПО, компания НОУДПО «Институт «АйТи» (бренд – Академия АйТи) провела обучение учителей в пилотных регионах с применением дистанционных технологий.

2009 год. Проект **«Обеспечение поддержки внедрения ПСПО в общеобразовательных учреждениях Российской Федерации»**, в рамках проекта «Образование», включает:

- «Развитие и модернизация пакета свободного программного обеспечения для общеобразовательных учреждений Российской Федерации».
- "Оказание услуг технической поддержки общеобразовательных учреждений Российской Федерации при внедрении и использовании пакета свободного программного обеспечения“.
- "Обучение специалистов сферы образования внедрению пакета свободного программного обеспечения и использованию свободного программного обеспечения в учебном процессе“.

В рамках проекта «Обеспечение поддержки внедрения ПСПО в общеобразовательных учреждениях Российской Федерации» продекларированы следующие положения:

I. Цели и задачи проекта

1. Основной целью проекта является создание условий для массового внедрения специально разработанного и прошедшего апробацию пакета свободного

программного обеспечения (ПСПО) в общеобразовательных учреждениях Российской Федерации (ОУ РФ).

2. Для достижения указанной цели в рамках проекта требуется решить следующие задачи:

- Обеспечить ввод в эксплуатацию единого государственного репозитория свободного программного обеспечения, как средства разработки, тестирования и распространения ПСПО.
- Обеспечить сопровождение и развитие программных продуктов ПСПО в соответствии с требованиями, предъявляемыми образовательными учреждениями.

II. Целевой индикатор и показатель направления деятельности в рамках направления «Внедрение современных образовательных технологий» приоритетного национального проекта «Образование»:

1. Повышение удельного веса образовательных учреждений, входящих в сетевое взаимодействие.
2. Удельный вес численности занятого населения, прошедшего повышение квалификации и профессиональную переподготовку.
3. Увеличение объема трафика, приходящегося на обращение к информационно-образовательным ресурсам.

III. Область практического использования и применения результатов выполнения работ (оказания услуг) в системе образования с указанием конкретных категорий специалистов, конкретных уровней и ступеней образования, пользователей результатов работ (услуг)

1. Федеральный, региональный, муниципальный уровни управления образованием.
2. Общеобразовательные учреждения Российской Федерации.

В проекте «Обучение специалистов сферы образования внедрению ПСПО и использованию СПО в учебном процессе» основными этапами являются:

- Актуализация учебно-методической базы для системы непрерывного повышения квалификации административно-педагогических и педагогических работников образовательных учреждений по применению пакета свободного программного обеспечения (ПСПО) в профессиональной деятельности по результатам развития и модернизации ПСПО.
- Проведение обучения административно-педагогических и педагогических работников образовательных учреждений применению ПСПО в формате конференций.

- Обучение административно-педагогических и педагогических кадров внедрению и использованию СПО с применением дистанционных технологий с использованием актуализированной учебно-методической базы.
- Контрольные мероприятия по результатам дистанционного обучения административно-педагогических и педагогических кадров внедрению и использованию СПО с применением дистанционных технологий.
- Повышение квалификации представителей региональных образовательных учреждений и формирование регионального кадрового резерва для консультаций и методической поддержки образовательных учреждений региона, осуществляющих переход на свободное программное обеспечение.
- Мониторинг использования ПСПО в образовательных учреждениях Российской Федерации.

## 1.5. Основные понятия и принцип работы ОС

Прежде чем идти дальше, разберемся в некоторых терминах.

*Операционная система* (operating system) — это программное обеспечение (т.е. инструкции), которое заставляет аппаратное обеспечение (т.е. компьютер) выполнять какие-либо действия.

О Linux часто говорят, как о «просто очередной операционной системе». Попробуйте заглянуть глубже, и вы поймете, что это не так. Linux является флагманом в нынешнем море компьютерной индустрии, движущимся по направлению к системам с открытым и свободно распространяемым исходным кодом. Как надежная и перспективная операционная система, Linux включает в себя все самое лучшее, что только было в развитии программного мира.

Два фундаментальных отличия позволяют выделить Linux среди остальных операционных систем.

- Linux получила уникальную и гениально простую общедоступную лицензию (General Public License — GPL).
- Linux разработана и постоянно обновляется командой программистов-добровольцев из многих стран мира, которые работают совместно через Интернет.

Linux великолепна по многим причинам. С самого начала создатели системы планировали наделить ее следующими свойствами:

- **Многопользовательский режим.** Несколько пользователей могут подключиться к одному компьютеру одновременно.
- **Многозадачность.** Реальный приоритетный многозадачный режим дает

возможность ядру операционной системы эффективно выполнять по несколько программ

одновременно. Это крайне важно для работы многочисленных служб, выполняющихся одновременно на одном и том же компьютере.

- **Многоплатформенность.** В настоящее время Linux работает на платформах Intel, Digital/Compaq Alpha, PowerPC (Apple Macintosh), Sun Sparc, Amiga и StrongARM.

- **Способность к взаимодействию.** Linux хорошо взаимодействует с большинством сетевых протоколов и операционных систем, включая Microsoft Windows, Unix, Novell и Mac OS (для компьютеров Apple).

- **Масштабируемость.** Производительность Linux может расти по мере увеличения потребностей ее владельца. Эта операционная система одинаково хорошо работает как на настольных, так и на очень больших, промышленных компьютерах.

- **Переносимость.** Большая часть Linux написана на языке программирования C. Это язык среднего уровня, созданный специально для написания системного программного обеспечения, которое можно переносить на компьютеры с новейшим оборудованием.

- **Гибкость.** Операционную систему Linux можно настроить таким образом, чтобы она была сетевым узлом, маршрутизатором, графической рабочей станцией, офисным компьютером, домашним развлекательным центром, файловым сервером, Web-сервером, кластером или выполняла любую другую задачу.

- **Надежность.** Ядро Linux достигло достаточно совершенно. Нередки случаи, когда серверы под управлением Linux годами работают без каких-либо сбоев.

- **Эффективность.** Модульная архитектура Linux позволяет настраивать систему таким образом, чтобы подключать к ней только те компоненты, которые вам нужны.

## 2. Работа в KDE. Работа с ПСПО

### 2.1. Основы работы с ПК

#### 2.1.1. Персональный компьютер

**Персональный компьютер** в своей базовой конфигурации состоит из системного блока, монитора, клавиатуры и мыши. Также к компьютеру можно подключить разнообразные периферийные (внешние) устройства: принтер, сканер, графопостроитель (плоттер), модем, микрофон, акустика, веб-камера и т.д.

1. Монитор
2. Материнская плата
3. Процессор
4. IDE-слот

5. Оперативная память
6. Платы расширения (видео, звуковая и т.д.)
7. Блок питания
8. Привод для дисков (CD/DVD)
9. Винчестер
10. Клавиатура
11. Мышь

### **2.1.2. Аппаратное обеспечение, программное обеспечение**

Аппаратное обеспечение – система взаимосвязанных технических устройств, выполняющих ввод, хранение, обработку и вывод информации.

*Системный блок* – это основная деталь компьютера. С формальной точки зрения это и есть компьютер, а всё остальное – периферийные устройства, то есть устройства, подключаемые к нему и обеспечивающие конкретные функции. Системный блок персонального компьютера содержит корпус и находящиеся в нем источник питания, материнскую (системную, или основную) плату с центральным процессором и оперативной памятью, различные накопители (жесткий диск, дисководы, приводы CD-ROM или DVD-ROM), платы расширения (графическая плата, звуковая плата, сетевая плата, модем), TV-тюнер, дополнительные устройства и т.д. Для примера: множество специализированных компьютеров – серверов – не комплектуются ни мониторами, ни клавиатурами, ни мышами, они лишь выполняют свою основную функцию – вычисления, а доступ и управление ими осуществляется с помощью другого компьютера – удалённого терминала.

На корпусе системного блока всегда есть кнопка включения и иногда перезагрузки компьютера, индикаторы включённого питания и активности жёсткого диска. На задней панели находятся отверстия для портов материнской платы.

Все *кабели* можно разделить на две большие группы: сигнальные кабели, предназначенные в основном для передачи информационных сигналов, и кабели питания (power cord), обеспечивающие только электропитание соответствующего устройства.

*Сетевые фильтры* предназначены для эффективной защиты компьютера от импульсных и высокочастотных помех (скачков напряжения), приходящих из сети питания. Плавкий предохранитель отключает фильтр при коротком замыкании.

*Источники бесперебойного питания (ИБП)* предназначены для питания компьютера в течение небольшого промежутка времени при отсутствии электроэнергии, для того чтобы можно было сохранить открытые документы и правильно завершить работу компьютера.

*Материнская плата* – печатная плата, на которой осуществляется монтаж большинства компонентов компьютерной системы. Название происходит от английского motherboard, иногда используется сокращение MB или слово mainboard – главная плата.

Материнская плата обеспечивает связь между всеми устройствами ПК, посредством передачи сигнала от одного устройства к другому. На ней размещаются микропроцессор; оперативная память; набор управляющих микросхем, или чипсетов(chipset); ПЗУ с системной BIOS(базовой системой ввода/вывода); слоты расширения; разъемы для подключения интерфейсных кабелей жестких дисков, дисководов; разъемы питания; разъемы последовательного (COM) и параллельного (LPT) портов; универсальной последовательной шины USB; разъем PS/2 для подключения клавиатуры и мыши и ряд других компонентов. На материнской плате также могут находиться микросхемы видеоадаптера, звуковой платы и сетевой карты.

*Центральный процессор, или центральное процессорное устройство (ЦПУ)* (англ. central processing unit – CPU) – основная микросхема компьютера, в которой и производятся все вычисления. ЦПУ устанавливается на материнской плате. На процессоре установлен большой радиатор, охлаждаемый вентилятором (cooler). Конструктивно процессор состоит из ячеек, в которых данные могут не только храниться, но и изменяться. Внутренние ячейки процессора называют регистрами. Важно также отметить, что данные, попавшие в некоторые регистры, рассматриваются как команды, управляющие обработкой данных в других регистрах. Таким образом, управляя засылкой данных в разные регистры процессора, можно управлять обработкой данных. На этом и основано исполнение программ.

С остальными устройствами компьютера, и в первую очередь с оперативной памятью, процессор связан несколькими группами проводников, называемых шинами. Основных шин три: *адресная шина, шина данных и командная шина.*

*Оперативная память (ОЗУ – оперативное запоминающее устройство).* Существует два типа оперативной памяти – память с произвольным доступом (RAM – Random Access Memory) и память, доступная только на чтение (ROM – Read Only Memory). Процессор ЭВМ может обмениваться данными с оперативной памятью с очень высокой скоростью, на несколько порядков превышающей скорость доступа к другим носителям информации, например, дискам.

*Оперативная память с произвольным доступом (RAM)* служит для размещения программ, данных и промежуточных результатов вычислений в процессе работы компьютера. Данные могут выбираться из памяти в произвольном порядке, а не строго последовательно, как это имеет место, например, при работе с магнитной лентой.



*Память, доступная только на чтение (ROM)*, используется для постоянного размещения определенных программ, например, программы начальной загрузки ЭВМ – BIOS (basic input-output system – базовая система ввода-вывода). В процессе работы компьютера содержимое этой памяти не может быть изменено.

Оперативная память – энергозависимая, т.е. данные в ней хранятся только до выключения ПК. Для долговременного хранения информации служат дискеты, винчестеры, компакт-диски и т.д.

*Носители информации:* жесткий диск, дисковод, накопители на компакт-дисках, накопители на DVD-дисках, флэш-память

*Жесткий диск* - накопитель на жёстких магнитных дисках, жёсткий диск или винчестер (англ. Hard Disk Drive, HDD) – энергонезависимое, перезаписываемое компьютерное запоминающее устройство. Является основным накопителем данных практически во всех современных компьютерах.

*Дискета* – портативный магнитный носитель информации, используемый для многократной записи и хранения данных сравнительно небольшого объема. Этот вид носителя был особенно распространён в 1970-х – начале 1990-х гг. Вместо термина «дискета» иногда используется аббревиатура ГМД – «гибкий магнитный диск» (соответственно, устройство для работы с дискетами называется НГМД – «накопитель на гибких магнитных дисках»).

*Накопители на компакт-дисках* Цифровая информация представляется на компакт-дисках (CD) чередованием впадин (не отражающих пятен) и отражающих свет островков. Компакт-диск имеет всего одну физическую дорожку в форме непрерывной спирали, идущей от наружного диаметра диска к внутреннему. Считывание информации с компакт-диска происходит при помощи лазерного луча, который, попадая на отражающий свет островок, отклоняется на фотодетектор, интерпретирующий это как двоичную единицу. Луч лазера, попадающий во впадину, рассеивается и поглощается: фотодетектор фиксирует двоичный ноль.

*DVD (Digital Versatile Disc, цифровой многоцелевой, или универсальный, диск)* – это оптические диски большой емкости, которые применяются для хранения полнометражных фильмов, музыки высокого качества, компьютерных программ. Существует несколько вариантов DVD, отличающихся по емкости: односторонние и двухсторонние, однослойные и двухслойные.

*Флэш-память (flash)* – разновидность полупроводниковой энергонезависимой перезаписываемой памяти.

*Платы расширения:* графическая плата, звуковая плата, сетевая плата, модем, TV-тюнер

*Графическая плата (известна также как графическая карта, видеокарта, видеоадаптер) (англ. videocard) – устройство, преобразующее изображение, находящееся в памяти компьютера, в видеосигнал для монитора.*

*Звуковая плата (также называемая звуковая карта, аудиоадаптер) используется для записи и воспроизведения различных звуковых сигналов: речи, музыки, шумовых эффектов.*

*Сетевая плата (также известная как сетевая карта, сетевой адаптер, Ethernet card, NIC (англ. network interface card)) – печатная плата, позволяющая взаимодействовать компьютерам между собой посредством локальной сети.*

*Модем – это устройство, способное осуществлять модуляцию и демодуляцию информационных сигналов; как правило, используется для подключения домашнего компьютера к сети Интернет по телефонной линии.*

*ТВ-тюнер (англ. TV tuner, TV-тюнер) – устройство, предназначенное для приёма телевизионного сигнала в различных форматах вещания (PAL, S?CAM, NTSC) с показом на компьютере или просто на отдельном мониторе. Tune означает «настраивать» (на длину волны).*

*Мониторы электронно-лучевые (CRT). ЭЛТ – электронно-лучевая трубка, CRT – Cathode Ray Tube. Изображение на экране ЭЛТ-монитора получается в результате облучения люминофорного покрытия остронаправленным пучком электронов, разогнанных в вакуумной колбе. Для получения цветного изображения люминофорное покрытие имеет точки или полосы трех типов, светящиеся красным, зеленым и синим цветом.*

*Мониторы жидкокристаллические (LCD). ЖК – жидкокристаллические, LCD – Liquid Crystal Display. ЖК-монитор состоит из двух слоев стекла с нанесенными на них тонкими бороздками и электродами, заключенного между ними слоя жидких кристаллов, осветителя и поляризаторов. Жидкие кристаллы под действием электрического поля поворачивают плоскость поляризации света на определенный угол. Далее свет проходит через поляризатор, который пропускает его с интенсивностью, зависящей от угла поворота плоскости поляризации. Цвет получается в результате использования трех цветных фильтров, разделяющих белый свет на составляющие RGB.*

*PDP – Plasma Display Panel. Как и в ЭЛТ-мониторе, в плазменной панели светится люминофор, но не под воздействием потока электронов, а под воздействием плазменного*

разряда. Каждая ячейка плазменного дисплея – флуоресцентная мини-лампа, которая способна излучать только один цвет из схемы RGB.

*Клавиатура (keyboard)* содержит 101 или 104 клавиши. Стандартом расположения символьных клавиш является раскладка QWERTY (ЙЦУКЕН) по названию клавиш верхнего символьного ряда слева направо.

*Манипуляторы*, или координатные устройства ввода информации, являются неотъемлемой частью современного компьютера. Наиболее известны следующие типы манипуляторов: мышь, трекбол, графические планшеты, устройства ввода, применяемые в ноутбуках – тачпад и трэкпойнт, а также джойстики. Изначально для подключения мыши к компьютеру использовался провод (в обиходной речи «хвост»), который подключался в один из портов компьютера. Первым из широко применяемых стандартных портов стал СОМ-порт, впоследствии его сменил порт PS/2, который в настоящее время всё больше вытесняется портом USB. Провод часто являлся помехой при работе с мышью, поэтому от него неоднократно пытались избавиться. Первыми попытками было внедрение инфракрасной связи между мышью и специальным приёмным устройством, которое, в свою очередь, подключалось к порту компьютера. Но оптическая связь, как показала практика, тоже не лишена недостатка, любое препятствие между мышью и датчиком мешало работе. Радиосвязь между мышью и приёмным устройством, подключённым к компьютеру, позволила избавиться от недостатков инфракрасной связи. Сейчас для связи стало всё более широко применяться Bluetooth-соединение, это позволяет избавиться от приёмного устройства, так как некоторые компьютеры уже оснащены Bluetooth-адаптером.

*Графический планшет (дигитайзер, диджитайзер)* – это устройство для ввода рисунков от руки непосредственно в компьютер. Состоит из пера и плоского планшета, чувствительного к нажатию пера. Также к планшету может прилагаться специальная мышь.

*Принтер (от англ. printer – печатник)* – устройство печати информации на твердый носитель, обычно на бумагу. Процесс печати называется выводом на печать, а результат – распечаткой. Принтеры, в зависимости от вида печати, разделяют на цветные и монохромные, в зависимости от способа нанесения изображения – на матричные, струйные, лазерные.

*Матричный принтер* является старейшим из ныне применяемых типов принтеров, его механизм был изобретён в 1964 г. компанией Seiko Epson. Основными недостатками данного типа принтеров являются низкая скорость работы и высокий шум, однако благодаря дешевизне копии (расходным материалом, по сути, является только красящая

лента) и возможности работы с непрерывной (рулонной, фальцованой) и копировальной бумагой они незаменимы, когда требуется печать на непрерывной бумаге (лаборатории, промышленность, бухгалтерия, ведение отчетов, печать чеков в магазинах, банкоматах и т.п.), многослойных бланках (например, авиабилеты), или минимальная стоимость печати.

Первый *струйный принтер* появился в 1976 г. Это был принтер от компании IBM. Принцип печати последовательный, безударный. Изображение формируется из микрокапель (~ 50 мкм) чернил, которые выдуваются из сопел картриджа. Засорение сопел, а точнее засыхание чернил в соплах – это существенный конструктивный недостаток струйных принтеров.

*Лазерные принтеры* менее требовательны к бумаге, чем, например, струйные, а стоимость печати одной страницы текстового документа у них в несколько раз ниже. Большинство представленных на рынке лазерных принтеров предназначены для черно-белой печати; цветные лазерные принтеры пока дороги и рассчитаны на корпоративных пользователей. Лазерные принтеры печатают на бумаге плотностью от 60 г/м<sup>3</sup> со скоростью от 8 до 24 листов в минуту (ppm – page per minutes), при этом разрешение может быть 1200 dpi и более. Качество текста, напечатанного на лазерном принтере с разрешением 300 dpi, примерно соответствует типографскому. Однако если страница содержит рисунки, содержащие градации серого цвета, то для получения качественного графического изображения потребуется разрешение не ниже 600 dpi. При разрешающей способности принтера 1200 dpi отпечаток получается почти фотографического качества. Если необходимо печатать большое количество документов (например, более 40 листов в день), лазерный принтер представляется единственным разумным выбором.

*Графопостроитель, плоттер* – устройство для автоматического вычерчивания с большой точностью рисунков, схем, сложных чертежей, карт и другой графической информации на бумаге размером до A0 или кальке. Графопостроители рисуют изображения с помощью пера (пишущего блока). Распространенное заблуждение: широкоформатные струйные принтеры иногда неверно называют плоттерами.

*Сканер (англ. scanner)* – устройство, которое создаёт цифровое изображение сканируемого объекта. Полученное изображение может быть сохранено как графический файл, или, если оригинал содержал текст, распознано посредством программы распознавания текста и сохранено как текстовый файл.

*Цифровой фотоаппарат* – это устройство для фотографической фиксации изображений. В плёночном фотоаппарате изображение получается при попадании на пленку света, отраженного от объекта в момент открытия затвора. В цифровом фотоаппарате роль фиксирующего свет материала вместо пленки выполняет небольшая

пластина со светочувствительными датчиками, называемыми «сенсорами» или «пикселями». Матрица состоит из множества светочувствительных ячеек – пикселей. Ячейка при попадании на нее света вырабатывает электрический сигнал, пропорциональный интенсивности светового потока. Т.к. используется информация только о яркости света, картинка получается в оттенках серого.

**Конфигурацией (или спецификацией) компьютера** называют характеристики устройств, которые в этот компьютер включены.

Например, в прайс-листе компьютерной фирмы указана такая конфигурация:

Intel Core2 Duo – 3,0GHz/ 1Gb/ 400Gb/ 128Mb GeForce PCX6600/ DVD±RW (22xW/8xRW/16xR/48xW/32xRW/48xR)/ FDD/ LAN 1Gb/ SB/ kbd/ M&P/ 19.0» Samsung SyncMaster 970P black (DVI,1280×1024–6ms, 250cd/m<sup>2</sup>, 1000:1, 178°/178°)

Это следует читать так:

- процессор Intel Core 2 Duo двухъядерный с тактовой частотой 3,0 гигагерца;
- емкость оперативной памяти – 1 гигабайт;
- жесткий диск (винчестер) емкостью 400 гигабайт;
- графическая плата GeForce PCX 6600 со 128 мегабайтами видеопамяти;
- привод дисков DVD, у которого
  - скорость чтения: DVD - 16x, CD - 48x
  - скорость записи: DVD+R 22x, DVD+R DL 16x, DVD-R 22x, DVD-R DL 12x, CD-R 48x
  - скорость перезаписи: DVD+RW 8x, DVD-RW 6x, DVD-RAM 12x, CD-RW 32x
- дисковод для гибких дисков (FDD);
- сетевая плата со скоростью 1 гигабит (LAN1Gb);
- звуковая карта (SB);
- клавиатура (kbd – keyboard);
- манипулятор мышь и коврик для мыши (M&P – mouse and pad);
- жидкокристаллический 19-дюймовый монитор Samsung SyncMaster 970P с «родным» разрешением 1280×1024, с разъемом DVI для ЖК-мониторов, временем отклика 6 миллисекунд, яркостью 250 кд/м<sup>2</sup>, контрастность 1000:1, с углами обзора 178°/178°.

**Программное обеспечение** – наряду с аппаратными средствами, важнейшая составляющая информационных технологий, включающая компьютерные программы и данные, предназначенные для решения определённого круга задач и хранящиеся на машинных носителях.

Программное обеспечение (ПО) представляет собой алгоритм, реализованный в виде последовательности инструкций для процессора. В компьютерном жаргоне часто используется слово «софт» от английского software. По назначению программное обеспечение разделяется на системное, прикладное и инструментальное.

*Инструментальное ПО* – интегрированные среды разработки, SDK, компиляторы, интерпретаторы, линковщики, ассемблеры, отладчики и т.д.

К *прикладному ПО* относятся банковские и бухгалтерские программы, игры, текстовые и графические редакторы, интернет-клиенты и т.д.

*Системное программное обеспечение.*

Конкретные виды системного программного обеспечения включают загрузчики, операционные системы, драйверы устройств, утилиты (сервисные программы). Наиболее общая часть системного программного обеспечения – операционная система.

Операционная система, ОС (OS – operating system) – базовый комплекс компьютерных программ, обеспечивающий управление аппаратными средствами компьютера, работу с файлами, ввод и вывод данных, а также выполнение прикладных программ и утилит. При включении компьютера операционная система загружается в память раньше остальных программ и затем служит платформой и средой для их работы. С 1990-х гг. наиболее распространёнными операционными системами являются ОС семейства Microsoft Windows и системы класса UNIX (особенно Linux).

Основные функции ОС:

- загрузка приложений в оперативную память и их выполнение;
- стандартизированный доступ к периферийным устройствам (устройствам ввода-вывода);
- управление оперативной памятью;
- управление энергонезависимой памятью (жесткий диск, компакт-диски и т.д.), как правило, с помощью файловой системы;
- пользовательский интерфейс.

Дополнительные функции ОС:

- параллельное или псевдопараллельное выполнение задач (многозадачность);
- взаимодействие между процессами;
- межмашинное взаимодействие (компьютерная сеть);
- защита самой системы, а также пользовательских данных и программ от вредоносных действий пользователей или приложений;
- разграничение прав доступа и многопользовательский режим работы (аутентификация, авторизация).

### 2.1.3. Последовательность включения ПК

Итак, в предыдущих разделах Вы познакомились с внешним видом компьютера и периферийных устройств. Однако перед началом работы с компьютером необходимо включить его в электросеть. Процесс включения компьютера следующий:

#### *Подготовительные операции.*

1. Проверьте, нет ли повреждения корпуса у компьютера и периферийных устройств. Устройства со сломанными корпусами использовать нельзя! В случае неисправности корпуса вызовите специалиста.
2. Заправьте принтер бумагой (если Вы его используете).
3. Проверьте, нет ли каких-нибудь «висящих» проводов, не подключенных к компьютеру.
4. Если Вы используете компьютерную сеть, проверьте подключения кабелей к адаптеру сети.

#### *Порядок включения.*

Компьютерные устройства вообще включаются «от периферии к центру». Это значит, что необходимо включать устройства в следующей последовательности:

1. Включите все USB- и SCSI-устройства (USB – универсальная последовательная шина, SCSI – стандартный интерфейс малых компьютеров.) К USB-устройствам, как правило, относятся недорогие сканеры, цифровые фотоаппараты и видеокамеры, иногда мониторы, клавиатура, принтер и мышь. К SCSI-устройствам относят различные сканеры и дополнительные дисководы.
2. Включите приборы, подключенные к последовательным и параллельным портам компьютера (принтеры, модемы и т.д.)
3. Включите монитор (кнопкой на его передней панели)
4. Удалите дискету из дисковода, если она в нем есть

*Включение в сеть.* Включите кнопку Сеть на передней панели компьютера. После включения компьютера начинается внутреннее тестирование компьютера. В процессе тестирования проверяется:

1. Источник питания.
2. Видеокарта.
3. Процессорный блок.
4. Память.
5. Подключенные дисководы.
6. Внешние устройства.

Если хотя бы одно из этих устройств не работает нормально, компьютер издаст несколько длинных писков и дальше загружаться не будет. В этом случае необходимо позвать мастера. Если тест прошел успешно, то после короткого писка на экране появляется информация о конфигурации компьютера, и начинает загружаться операционная система. Далее при загрузке операционной системы на экране будет отображаться заставка. Заставка может прерываться запросом на выбор конфигурации операционной системы.

Если в устройстве чтения компакт-дисков находился диск, то система может начать загружаться с этого диска. В этом случае, удалите компакт-диск и перезагрузите компьютер кнопкой на корпусе.

*Вам важно знать, что на этапе загрузки программ:*

- у Вас могут запросить пароль на вход в систему;
- Вас могут попросить выбрать конфигурацию компьютера. Выбирайте конфигурацию, заданную по умолчанию (либо ничего не делайте, либо нажмите клавишу Enter);
- у Вас могут запросить идентификационное имя («логин») и пароль для входа в сеть. Введите свой логин и пароль, а если Вы его не знаете или Вам его не давали, в поле имени наберите «guest» (без кавычек и строчными буквами!), игнорируйте пароль и нажмите Enter.

## **2.2. Пользовательский интерфейс**

### **2.2.1. Запуск графического интерфейса пользователя, завершение работы**

Итак, включение компьютера, в нашем случае, приводит к автоматической загрузке Linux, вы увидите на мониторе окно загрузки Линукс Мастер. Для быстрой загрузки следует нажать Enter, либо система загрузится сама после небольшой паузы. Остальные пункты меню нужны для восстановления системы в случае сбоев и для других специфических задач администрирования.



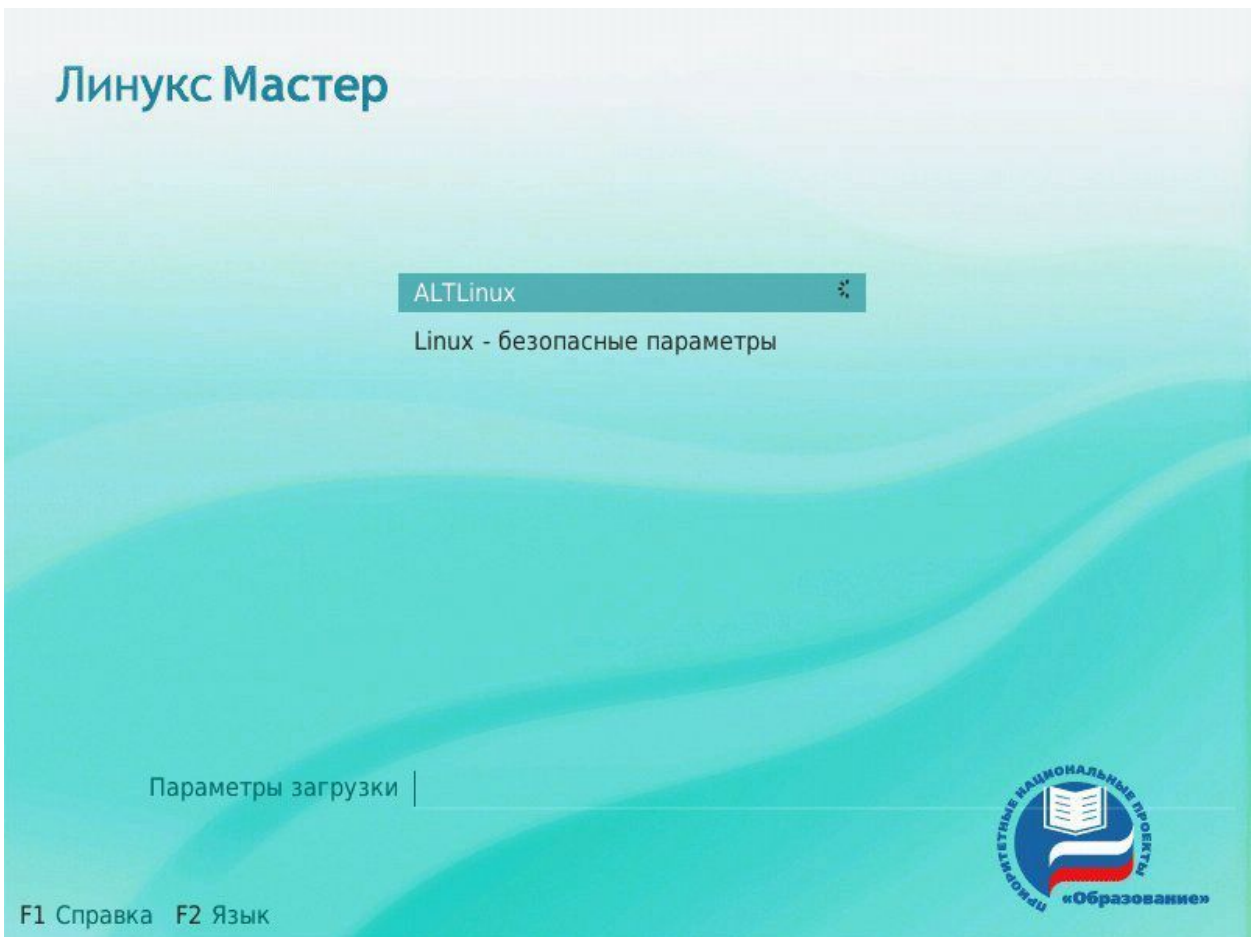


Рис. 1. Окно загрузки системы

Linux (в нашем случае Линукс Мастер)— это многопользовательская система. Один пользователь может работать непосредственно за компьютером, где установлен Linux, остальные пользователи могут работать удаленно, по сети. На практике это означает, что для работы в системе нужно в ней зарегистрироваться, т. е. дать понять системе, кто именно находится за монитором и клавиатурой. Вместо формального «зарегистрироваться в системе» часто используют выражение «войти в систему». Операционная система представляется чем-то вроде замкнутого помещения, внутри которого можно оказаться, только успешно проникнув через «дверь» — пройдя процедуру регистрации. Наиболее распространённый способ регистрации на сегодняшний день — использование системных имён (login name) и паролей (password). Это надёжное средство убедиться, что с системой работает тот, кто нужно, если пользователи хранят свои пароли в секрете и если пароль достаточно сложен и не слишком короток (иначе его легко угадать или подобрать).

После небольшой паузы, дальнейшая загрузка операционной системы заканчивается интерфейсом входа в систему: выводится приглашение ввести системное имя пользователя (**login:**) и **пароль**. Если загрузка произошла в графическом режиме, то можно не вводить системное имя вручную, а нажать на кнопку с нужным именем, однако

пароль при этом всё равно нужно ввести самостоятельно. Имя пользователя и пароль вам должен сказать администратор, иначе войти с систему не удастся.

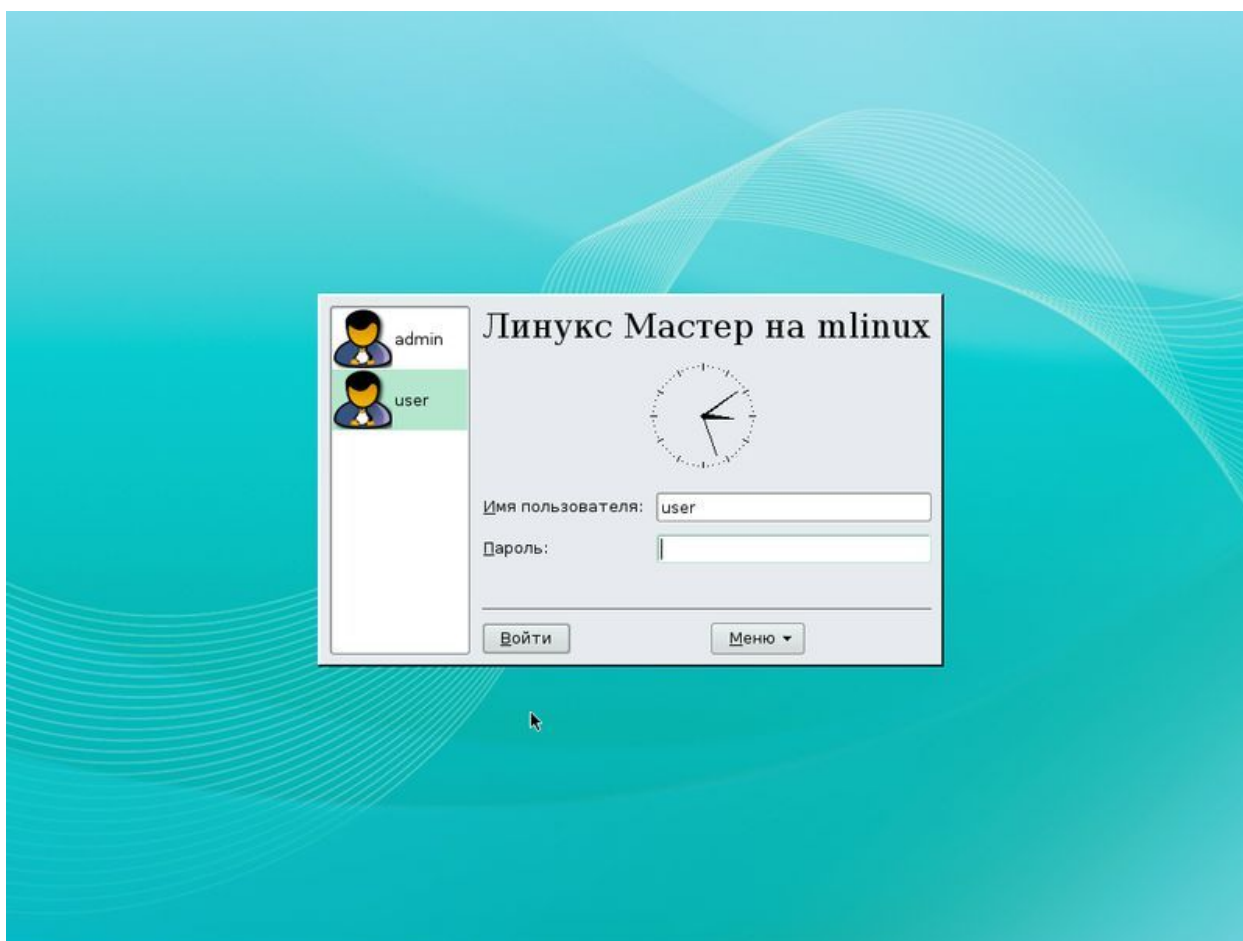


Рис. 2. Интерфейс входа в систему

Пользователи обычно создаются непосредственно в процессе установки системы, однако всегда можно добавить новых пользователей или удалить существующих при помощи стандартных средств управления пользователями.

При установке системы был создан специальный пользователь с неограниченными правами в системе для выполнения административных действий. Имя этого пользователя – **root**.

Из соображений безопасности не следует входить в систему в качестве пользователя root. Для выполнения обычных администраторских задач (изменение настроек системы) обычный пользователь может временно получить права администратора.

В Linux у каждого пользователя обязательно есть свой домашний каталог, предназначенный для хранения всех собственных данных пользователя. Именно с этого каталога пользователь начинает работу после регистрации в системе. Домашние каталоги пользователей обычно собраны в каталоге /home, их название чаще всего совпадает с

учётным именем пользователя в системе, например, для пользователя `test` домашним каталогом будет `/home/test`.

Пользователь является полным хозяином внутри своего каталога, однако остальная часть файловой системы доступна ему только для чтения, но не для записи. Доступ других пользователей к чужому домашнему каталогу ограничен: наиболее типична ситуация, когда пользователи могут читать содержимое файлов друг друга, но не имеют права их изменять или удалять.

В операционной системе Линукс Мастер пользователю доступны два режима работы: графический и текстовый. В текстовом режиме недоступны возможности графических интерфейсов: рисование окон произвольной формы и размера, поддержка миллионов цветов, отрисовка изображений. Все возможности текстового режима ограничены набором текстовых и псевдографических символов и несколькими десятками базовых цветов. Тем не менее, в Linux в текстовом режиме можно выполнять практически любые действия в системе (кроме тех, которые требуют непосредственного просмотра изображений). Текстовый режим в Linux — это полнофункциональный способ управления системой благодаря интерфейсу командной строки. В Linux существует огромное множество программ (включая даже игры), предназначенных для работы в текстовом режиме.

Бывают ситуации, когда графический режим недоступен или неработоспособен (удалённый доступ по сети, проблемы с поддержкой видеокарты, сбои системы и др.). В таких случаях всегда остаётся возможность работать в текстовом режиме, поскольку его возможности поддерживаются непосредственно графическим оборудованием и не требуют специальных драйверов или настройки.

В процессе работы Linux активно несколько виртуальных консолей. Каждая виртуальная консоль доступна по одновременному нажатию `Alt` и функциональной клавиши с номером этой консоли. На первых шести виртуальных консолях (`Alt+F1` — `Alt+F6`) пользователь может зарегистрироваться и работать в текстовом режиме. 12-ая виртуальная консоль (`Alt+F12`) выполняет функцию системной консоли — на неё выводятся сообщения о происходящих в системе событиях.

Если загрузка системы по каким-то причинам не дошла до графического режима и завершилась приглашением к регистрации (`login:`) на текстовой виртуальной консоли, то можно попробовать запустить графический режим вручную. Для этого следует войти в систему (ввести имя пользователя и пароль) и ввести команду **startx**. Эта команда запускает графическую подсистему X11.

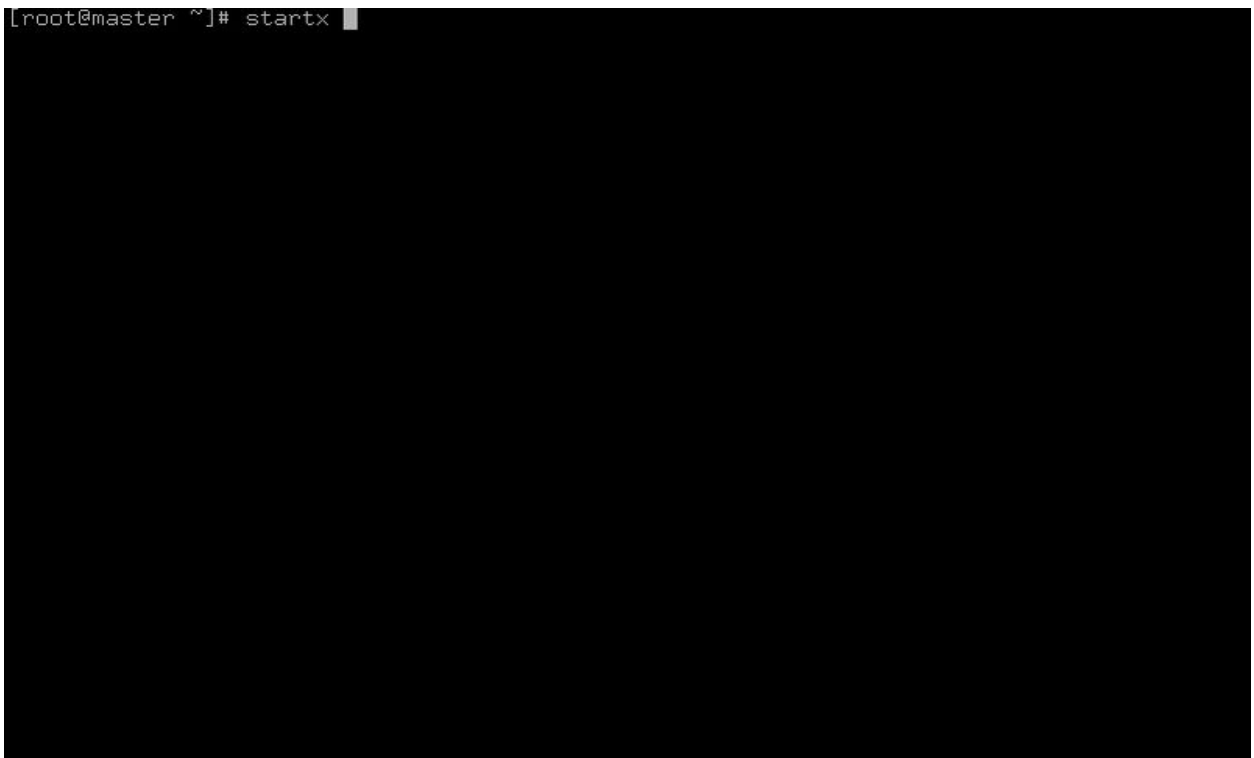


Рис. 3. Запуск графического режима вручную

Чтобы переключиться из графического режима на другую консоль, следует нажимать одновременно Ctrl, Alt и функциональную клавишу с номером нужной консоли.

Благодаря виртуальным консолям каждый компьютер, на котором работает Linux, предоставляет возможность зарегистрироваться и получить доступ к системе одновременно нескольким пользователям. Даже если в распоряжении всех пользователей есть только один монитор и одна системная клавиатура, эта возможность небесполезна: можно переключаться между виртуальными консолями так, как если бы вы переходили от одного монитора с клавиатурой к другому, подавая время от времени команды и следя за выполняющимися там программами. Более того, ничто не препятствует зарегистрироваться в системе несколько раз под одним и тем же системным именем — это один из способов организовать параллельную работу над несколькими задачами.

### **Завершение работы**

Linux нельзя выключать, просто лишив компьютер электропитания. Множество информации, которая должна располагаться на диске, система держит в оперативной памяти для повышения быстродействия. Неожиданное выключение питания приводит к потере этой информации. Если вы работаете в графической среде, то для завершения работы нужно выбрать в главном меню пункт *Завершить сеанс* и выбрать далее один из трех вариантов: *Завершить текущий сеанс*, *Выключить компьютер* или *Перезагрузить компьютер*.

Если перед вами — графический интерфейс входа в систему, то там также есть кнопка меню, в котором имеется пункт «выключить компьютер». В командной строке (например, на текстовой виртуальной консоли), можно выполнить команду `halt` (требует привилегий администратора). Процедура выключения завершится автоматическим отключением питания компьютера, если это возможно. Если автоматическое отключение невозможно, на экран выведется соответствующее сообщение, и питание можно отключить кнопкой `Power`. Если компьютер поддерживает протокол работы ACPI, нажатие на кнопку `Power` приводит не к выключению электропитания, а к передаче системе ACPI-сообщения о том, что кнопка `Power` была нажата. При получении такого сообщения система выполняет ту же процедуру выключения. Только в этом случае допустимо выключение кнопкой `Power`.

## **2.2.2. Обзор графической среды**

### **2.2.2.1. Оконная система X и её реализации**

Графический интерфейс не является неотъемлемой частью Linux — это просто одна из её компонент, такая же необязательная с точки зрения архитектуры системы, как, например, программа для рисования изображений. Но для тех программ, которые используют графические ресурсы, эта компонента предоставляет возможность работать с графическими объектами (линиями, прямоугольниками, цветами), ничего не зная о деталях работы конкретных устройств графического вывода (видеокарты и монитора). Это похоже на то, как ядро скрывает от программ детали работы с конкретным оборудованием, например, жёстким диском, предоставляя им работать с файлами. Поэтому комплекс программ, предоставляющий доступ к графическим ресурсам, называют графической подсистемой. В Linux функции графической подсистемы выполняет оконная система Икс.

Графическая подсистема с точки зрения операционной системы представляет собой группу обычных процессов, управление которыми производится общесистемными средствами. Точно так же, общесистемными средствами производится и управление процессами, запускаемыми «из-под» этой графической среды. Графическая подсистема отнюдь не монополизирует использование компьютера; параллельно с её работой продолжает исполняться множество служебных системных процессов; с других терминалов (если система многотерминальная) могут запускаться другие программы или даже другие графические подсистемы.

*Оконная система Икс (от англ. X window system, далее — просто X)* — один из самых больших и успешных проектов в истории компьютерной техники — восходит к 1984 г., когда разработчики двух систем компьютерной графики, претендующих на

универсальность — проектов Athena (Массачусетский технологический институт) и W Windowing (Стэнфордский университет) — решили объединить свои усилия.

Тогда перед ними стояла задача создать систему компьютерной графики, позволяющую совместно использовать самые разные компьютерные платформы. Решением стало создание специального протокола X, который позволял разделить программы-клиенты и сервер, предоставляющий графические ресурсы, отсюда и возможность исполнять программу-клиент на одном компьютере, сервер на другом, а данные между ними передавать по сети.

Проект этот был настолько наукоёмок и настолько полно охватывал тогдашнюю область задач, связанных с графикой, что серьёзных альтернатив ему так и не возникло. С тех пор X прошла через одиннадцать основных релизов (отсюда другое название — *X11*, представляющее собой название и текущую версию протокола) и множество версий. И возникновение, и вся история развития X тесно связаны с ОС UNIX, а теперь, естественно, и Linux. Тем не менее, реализации X доступны и для нескольких альтернативных архитектур ОС, включая Windows.

#### **2.2.2.2. Менеджеры окон**

Для одновременной и поочерёдной работы с разными программами, требуется возможность управлять окнами (с помощью клавиатуры или мыши), т. е. возможность изменять «на лету» их геометрию (положение и размеры), а также (обычно не относимое к геометрии) положение — от этого зависит, какое из окон будет «верхним» (видимым полностью), если окна перекрывают друг друга на плоскости экрана.

Управление окнами и составляет основную функцию оконного менеджера. Устоявшийся англоязычный термин *window manager*, относящийся к этому классу программ, мы будем передавать далее словосочетанием - калькой «оконный менеджер».

Базовая (а также расширенная) функциональность оконных менеджеров доступна пользователю прежде всего за счёт введения в интерфейс так называемых виджетов (от англ. *widgets*, сокращение от *window gadgets*, «оконные приспособления»). Виджеты — это рамки, кнопки, меню и пр., которые служат «органами управления» окна. Технически (в терминах оконной системы X) виджеты представляют собой отдельные окна, примыкающие к окну прикладной программы и, как правило, перемещающиеся вместе с ним.

Оконные менеджеры обычно добавляют очень удобную для работы возможность — работу с «виртуальным» рабочим столом (от англ. *virtual desktop*, также рабочим столом), по размеру превышающим физический экран. Такие окна позволяют группировать запущенные приложения, по желанию переключаясь между виртуальными окнами.

Количество таких окон настраивается, часто их делают четыре. Enlightenment позволяет создавать до 64 экранов на рабочем столе, KDE до 20.

Переключение между виртуальными окнами осуществляется либо с использованием функциональных кнопок на клавиатуре, либо часто с помощью специального графического приложения, так называемого пейджера (pager). На пейджере представлена миниатюрная копия экрана, обновляющаяся в режиме реального времени, причём, если подвести курсор к изображению отдельного окна, оно увеличивается и рядом высвечивается название приложения, запущенного в нём.

Менеджер окон, который помимо управления окнами обладает рядом дополнительных функций, может использоваться в качестве операционной графической среды пользователя, предоставляющей полный спектр возможностей для параллельной работы с несколькими задачами. Наиболее часто такими дополнительными функциями являются следующие:

*Минимизация/сворачивание окон и управление свёрнутыми окнами.* Существует несколько относительно распространённых способов отображения свёрнутых окон. Например, «на столе» может оставаться полоса заголовка свёрнутого окна, по щелчку на которой оно вновь разворачивается. Свёрнутым окнам могут соответствовать пиктограммы (иконки, значки) на поверхности рабочего стола или в специально отведённом для этого окне (панели управления). Свёрнутые окна могут отображаться как пункты общего или специального меню (см. ниже).

*Управление несколькими рабочими столами.* Практика показывает, что для многих продвинутых пользователей, которые осваивают открытые системы, уже имея опыт работы в характерных для ПК альтернативных системах, именно возможность работать на нескольких рабочих столах оказывается решающим плюсом оконной системы X.  
*Быстрый запуск команд.* Возможность быстрого запуска предуготовленных команд обычно ассоциируется с общим меню (главным меню), вызываемым щелчком мыши на особом виджете, не связанном с прикладными окнами, или в свободной от прикладных окон области экрана.

*Настройка внешнего вида и поведения среды.* *Поведение* — реакция отдельных виджетов на операции с ними, модель фокусировки (способ переключения активного в данный момент окна, с которым связан ввод с клавиатуры и мыши) и т. п. *Поведение* и внешний вид оформления окон, а также наличие на экране общих виджетов, не связанных с конкретными прикладными окнами, обои (цвет фона или изображение в корневом окне) и т. п. могут варьировать в очень широких пределах. Иногда возможности такой настройки считают некими «архитектурными излишествами», однако более взвешенной

является точка зрения, согласно которой в хорошем визуальном дизайне (так же, как и в хорошей архитектуре) ничто не является излишеством.

Менеджеров окон существует превеликое множество — под любой набор задач, которые может решать графическая многооконная система. Их настолько много, что выбрать какой-нибудь в качестве «типичного представителя семейства» затруднительно.

Перечислим наиболее часто используемые.

*Enlightenment* «Просвещение» (англ. *enlightenment*). До 2000г. он был основным менеджером окон в популярной среде GNOME, затем уступив это место менее функциональной, но более быстрой «Рыбе-пиле» (Sawfish).

*BlackBox* — один из самых компактных и быстродействующих оконных менеджеров. Он позволяет эффективно организовать работу на рабочем столе, не «захламляя» его ненужными ссылками и не расходуя экранное пространство на отображение громоздких элементов оформления.

Наряду с базовой функциональностью, *BlackBox* предоставляет (факультативно) панель, содержащую кнопки переключения между рабочими столами (по умолчанию их четыре) и заголовки открытых окон. Общее меню вызывается щелчком правой кнопкой мыши на свободном от окон месте рабочего стола. Меню (или любое из вложенных в него меню) щелчком по заголовку может быть превращено в окно, остающееся на экране до явного его закрытия щелчком на соответствующей кнопке.

На основе *BlackBox* созданы два более развитых оконных менеджера — *OpenBox* и более популярный *FluxBox*. Внешний вид *BlackBox*, *FluxBox* и *OpenBox* легко настраивается с помощью механизма тем рабочих столов.

*WindowMaker (WM)* — это свободная реализация (в рамках проекта GNUStep) концепций NextSTEP — первой получившей более или менее широкую известность универсальной графической среды пользователя. За недоступностью оригинальной NextSTEP для современных платформ, познакомиться с *WM* полезно и поучительно вне зависимости от того, собираетесь ли вы с ним работать. Основным автономным виджетом *WM*, как и NextSTEP, является пирс прикладных программ, представленный при запуске пиктограммой со скрепкой. При запуске любой корректной (с точки зрения *WM*), а также некоторых некорректных программ, кроме её окна на экране появляется её пиктограмма. *WM* позволяет работать с несколькими рабочими столами. *WM* очень гибко настраивается, как в отношении внешнего вида, так и в отношении поведения, причём большая часть настроек доступна из специальной графической утилиты, запускаемой по щелчку на пиктограмме с изображением ступеньки.



### 2.2.2.3. Интегрированные графические среды

Существует два подхода к тому, как можно достроить оконную систему до полнофункциональной среды, позволяющей пользователю решать все (или почти все) его практические задачи. Во-первых, можно расширить функциональность менеджера окон, добавив в него недостающие возможности. Во-вторых, можно добавить менеджер рабочего стола — работающий «поверх» менеджера окон и использующий функциональность последнего. Этим путем идут команды разработчиков **GNOME** и **KDE**.

С точки зрения пользователя нет чёткой границы между менеджерами окон с расширенной функциональностью и менеджерами рабочего стола, работающими «поверх» менеджера окон, поскольку они обеспечивают одну и ту же функциональность и нередко даже графически организованы сходным образом. Оба варианта предоставляют пользователю возможность работать в графической среде (desktop environment).

Интегрированная графическая среда предполагает не только единство оформления, но и трактовку объектов в рабочем пространстве (окон, файлов, пунктов меню и т. п.) как физических объектов, которые можно перемещать, выбрасывать в «корзину» и т. д. Однако сколько-нибудь последовательной теории интегрированных графических сред не существует. Изучая отдельные среды в динамике их развития, можно, тем не менее, выделить несколько общих черт.

1. Они опираются на определённый интерфейс разработчика (API), состоящий из библиотек, доступных также разработчикам прикладных программ.
2. Они реализуют элементы объектной метафоры: файлы, процессы (их потоки ввода-вывода) изображаются как отдельные объекты, на них можно фокусироваться и выполнять с ними различные действия, их состояния и изменения этих состояний также могут визуализироваться или озвучиваться.
3. Они реализуют единообразные элементы управления (виджеты), зачастую не только в оформлении отдельных окон, но и в их содержимом.
4. Они содержат те или иные элементы управления, не привязанные к отдельным окнам прикладных программ (общие меню, панели управления, поверхность стола и т.п.).
5. Они позволяют согласованно изменять свойства интерфейса образующих среду программ (менеджера окон, менеджера рабочего стола, приложений, разработанных специально для данной среды).
6. Они реализуют буфер обмена, позволяющий передавать типизованные данные от программы программе (оконная система X содержит буфер, позволяющий передавать данные лишь простого текстового типа).

7. Они реализуют возможность «перетаскивания» при помощи мыши (drag'n'drop) объектов или данных между окнами одной программы или разных программ.

Однородность опыта при работе в интегрированных средах и связанная с нею привычность позволяют при освоении нового инструмента-программы сосредоточиться на её прикладной логике, не задумываясь и специально не фокусируя внимания на приёмах работы, общих для всех инструментов. Это позволяет новому пользователю гораздо быстрее осваивать прикладные программы.

Основной недостаток работы в интегрированной среде является оборотной стороной основного достоинства: жёстко закреплённые навыки мешают при выходе за её пределы. Конечному пользователю, ограниченному опытом работы в одной среде, недостаёт «стереоскопичности» видения, глубины понимания; элементы эргономической логики могут напрямую ассоциироваться с определёнными визуальными элементами и «жестами», с помощью которых подаются команды.

Общеизвестны сложности, с которыми сталкиваются люди, долгое время работавшие в одной графической среде, при необходимости поработать в другой (пусть даже и весьма схожей). Для преодоления таких сложностей крайне полезным представляется знакомство с разными средами уже на начальном этапе освоения графических интерфейсов. Это не обязательно должны быть разные интегрированные среды, но само представление о том, что один и тот же результат может достигаться с помощью разных интерфейсных средств весьма важно. На сегодня существуют и развиваются две свободные интегрированные графические среды общего назначения: KDE и GNOME. Они входят в поставку большинства стандартных (открытых) ОС, как свободных, так и несвободных.

*GNOME (GNOME, GNU Network Object Model Environment)* — один из самых амбициозных и масштабных проектов в программистском сообществе.

Кроме реализации функционально полной графической среды, GNOME претендует на то, чтобы полностью реализовать спецификации промышленной платформы сетевого взаимодействия CORBA и полностью абстрагировать слой менеджера рабочего стола (или графической среды) от низлежащего слоя управления окнами (менеджера окон).

GNOME поддерживает ряд оконных менеджеров, среди которых: Sawfish (оконный менеджер по умолчанию), Enlightenment, IceWM, WindowMaker, AfterStep и FVWM2, совместимые с GNOME.

Сегодняшняя версия GNOME – полноценная интегрированная среда, включающая реализацию повседневно необходимых функций и позволяющая использовать сторонние решения для реализации функциональности, которая в ней отсутствует.

GNOME использует один из самых развитых интерфейсных пакетов GTK+, реализованный для разных платформ. Над ним надстраивается масса компонентов и библиотек, обеспечивающих сетевую функциональность, интерфейсы к различным языкам программирования, работу со звуком через механизмы ОС и пр. Сам GNOME стремится оставаться мобильным и доступным во всех открытых системах.

С пользовательской точки зрения GNOME предстаёт как набор базовых компонентов интерфейса и апплетов, утилит и прикладных программ. К базовым компонентам относятся менеджер файлов и поверхности стола Наутилус (Nautilus), панели управления и меню GNOME Panel и центр управления (Gnome Control Center).

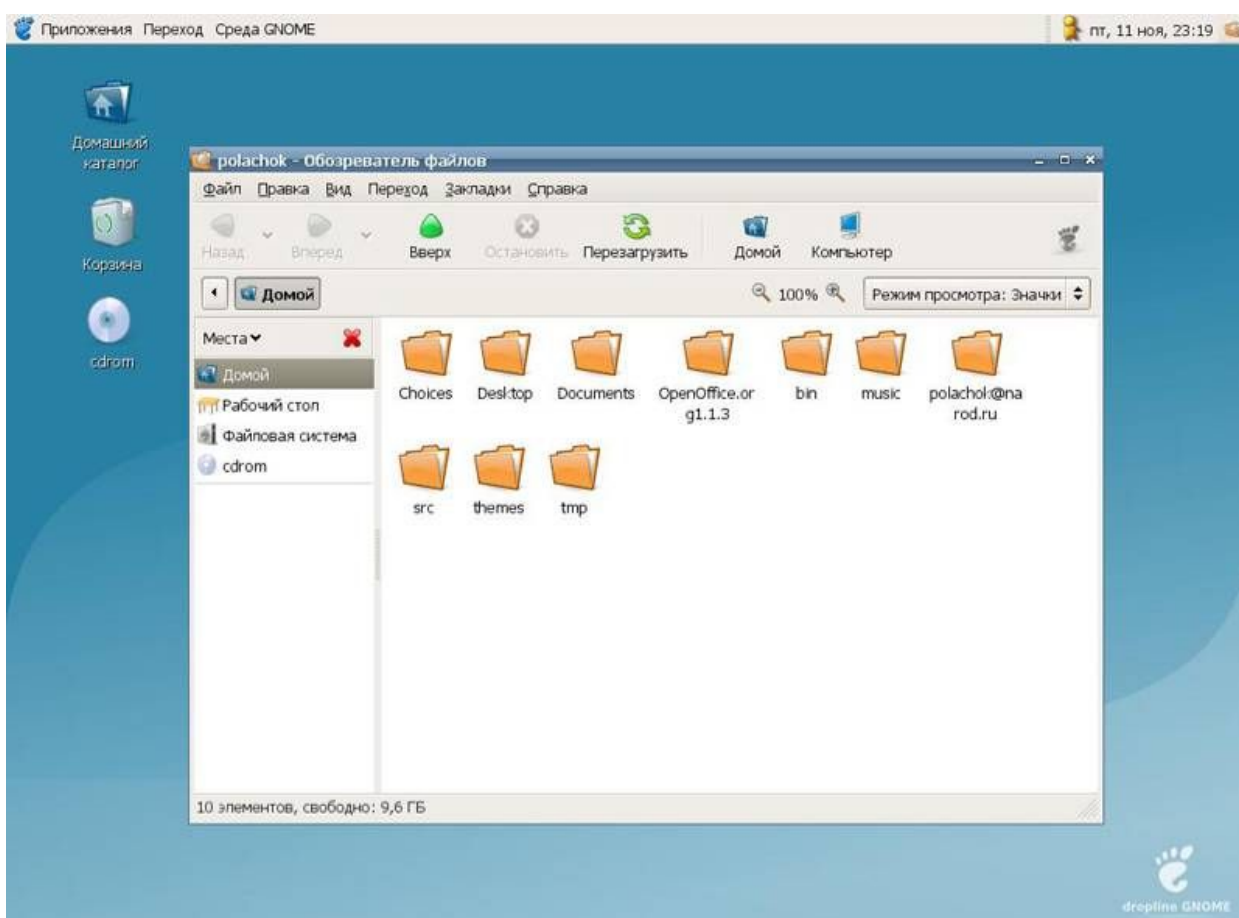


Рис. 4. Интегрированная среда GNOME

Внешний вид и поведение панелей является в высшей степени конфигурируемым. Пользователь может задавать как глобальные предпочтения (анимация движения панелей, отображение панельных объектов и пр.), так и индивидуальные предпочтения для каждой из них (её тип и положение на экране, ширина, возможность автоскрытия и принудительной минимизации, цвет и фоновое изображение и т. п.). Разумеется, пользователь может наполнять панели теми объектами, которые ему нужны.

На панелях могут присутствовать: *апплеты (applet, «приложение»)* – это программа, отображающая какую-нибудь полезную (или забавную) динамическую

информацию, например погоду, текущую загрузку подсистем компьютера (процессора, диска), текущую дату. Часто на панель помещают кнопки запуска, ассоциированные с приложением или командой, которые исполняются по щелчку.

Также постоянно расширяется набор утилит, прикладных программ и апплетов, поставляемых с GNOME — вместе с программами, входящими в большинство дистрибутивов ОС, о которых GNOME «в курсе», их число превышает сотню. Перечислить их здесь нет никакой возможности, но среди них есть интерфейсы для администрирования системы, средства звукозаписи и воспроизведения, сетевые утилиты, игры и многое другое.

GNOME снабжён встроенной системой помощи; кроме того, его разработчиками совместно с Sun Microsystems подготовлено компактное руководство, доступное в разных форматах на сайте проекта.

GNOME и большинство его компонентов соответствуют соглашениям об интернационализации и, соответственно, поддерживают работу с кириллицей, локализацию и перевод интерфейса.

#### **2.2.2.4. KDE**

Подробнее мы остановимся на KDE, поскольку именно эта графическая среда используется в нашей версии Linux.

Само название *KDE* (*KDE, K Desktop Environment* — «Графическая среда К») — явная пародия на CDE (*Common Desktop Environment* — «Общая настольная среда»). CDE была последней попыткой отрасли стандартизовать графическую среду на несвободной основе, предпринятой в конце девяностых годов. Буква «К» в KDE ничего не означает.

KDE — очень серьёзный проект. В KDE любят играть со словами; например, универсальный браузер, входящий в среду, называется Konqueror (от англ. conqueror — «завоеватель», «покоритель»), терминал — Konsole (от console — «консоль») и т.п. .

Если единообразие и однородность графической среды считать достоинством, то KDE — несомненный лидер среди всех (как свободных, так и несвободных) интегрированных графических сред. Основное видимое средство интеграции — это универсальный браузер Konqueror. Функция Konqueror близка к той, которую приобрёл Windows Explorer — он совмещает функции гипермедийного браузера WWW и браузера локальных ресурсов.

Разработчики KDE пошли даже дальше своих коллег из Microsoft и определили ряд дополнительных протоколов, что позволило, в частности, просматривать с помощью браузера в единообразном формате все разнообразие справочной информации, представленное в сегодняшних открытых система (традиционные страницы руководства

**man**, гипертекстовую систему **Info** из проекта GNU, разрозненные файлы документации в текстовом и гипертекстовом формате). В Konqueror интегрирована также возможность предварительного просмотра содержимого большого количества типов файлов.

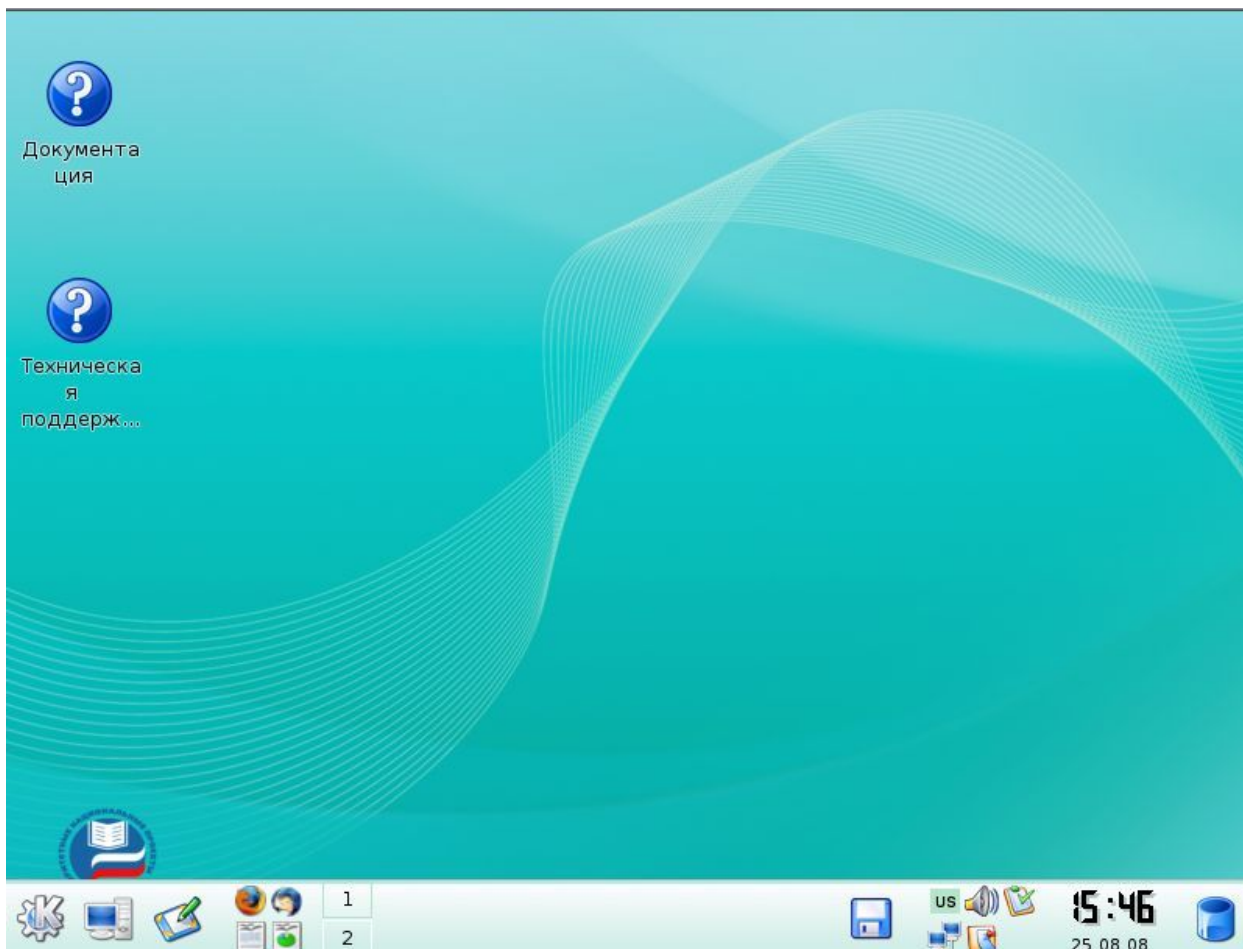


Рис.5. Интегрированная среда KDE

KDE включает также настраиваемую систему панелей и меню и интегрированный центр управления, позволяющий согласованно изменять параметры среды. KDE менее гибка в настройке, чем GNOME, однако её гибкости вполне достаточно для решения любых практических задач (в том числе, имитации вида и поведения других сред). KDE работает только с собственным оконным менеджером KWin.

В поставку KDE входит множество «аксессуаров» и прикладных программ, к тому же рядом с проектом выросла целая группа сопутствующих, ориентированных на те или иные предметные приложения, из которых самым развитым является офисный пакет KOffice.

Более подробно о настройке KDE вы узнаете в следующем разделе.

Следует отметить, что KDE - это постоянно развивающаяся система. Используемая нами версия Linux содержит KDE версии 3.5.9.

В 2008г вышла новая версия графической среды – KDE 4.

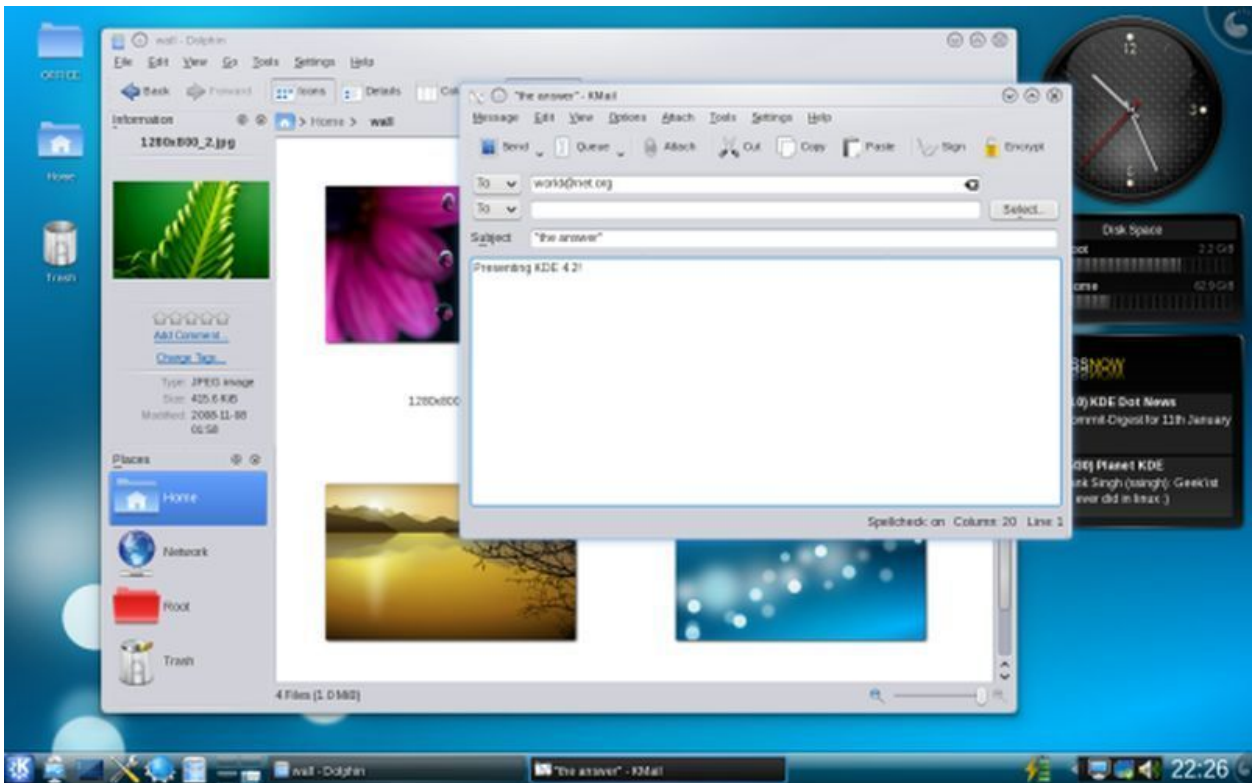


Рис.6. Интегрированная среда KDE 4

*Рабочий стол* в KDE 4 стал гораздо более функциональным. Оболочка Plasma значительно расширила его возможности, выведя панель, главное меню и виджеты рабочего стола на новый уровень и дополнив их приборной доской (dashboard). KWin, менеджер окон KDE, теперь поддерживает графические эффекты.

*Оболочка Plasma* позволяет пользователям просто организовывать своё рабочее пространство. Новые и доработанные старые апплеты - быстрый запуск программ, информация о погоде, ленты новостей, комиксы, быстрая публикация текста и картинок в службах pastebin. Апплеты Plasma теперь можно размещать на хранителе экрана, например, для того чтобы другие люди могли вам оставить записку, пока вас нет. Plasma также может выполнять роль традиционного рабочего стола со значками. Панель Plasma теперь может группировать окна принадлежащие одному приложению и располагать задачи в несколько рядов. Основательно доработанный системный лоток теперь позволяет следить за длительно выполняющимися заданиями, такими как загрузка файлов. Системные уведомления и уведомления программ теперь показываются однородно через системный лоток. Значки системного лотка можно скрывать. Также сама панель может автоматически скрываться для увеличения полезного рабочего пространства. Виджеты могут размещаться как на рабочем столе, так и на панели.

*Улучшена работа с окнами.*

В KDE 4 в анимации учитываются законы физики, что делает старые и новые эффекты более естественными. Трёхмерные эффекты включаются, только если системная конфигурация позволяет их демонстрировать с адекватной производительностью. Упрощён механизм настройки эффектов.

*Обновлены системные утилиты.*

PowerDevil приносит простое и гибкое управление питанием, особенно полезное на ноутбуках. Произведена работа над удобством архиватора Ark, а также механизмом работы с принтерами и заданиями печати.

KDE переводится на ещё большее количество языков, благодаря чему количество людей, которым доступен KDE на родном языке увеличивается до 700 миллионов. Среди новых языков многие - азиатские, что говорит о росте популярности KDE в этом регионе.

*Приложения делают шаг вперёд*

Выполнение типичных задач работы с файлами стало проще и эффективней. В Dolphin появился ползунок для точной настройки размера значков, добавлены всплывающие подсказки с уменьшенными копиями содержимого файлов, а также индикатор свободного места для съёмных носителей. Это касается и диалогов открытия/сохранения во всём KDE.

В KMail включён новый список сообщений, разработанный в рамках Google Summer of Code. Теперь можно более гибко настраивать вывод дополнительной информации. Также улучшена поддержка IMAP и других протоколов -- KMail теперь работает быстрее.

Улучшен веб-браузер Konqueror -- доработана поддержка векторной графики, повышена производительность. Новый механизм поиска делает поиск по веб-страницам более удобным. Konqueror теперь показывает закладки при запуске.

*Фундамент для упрощения разработки приложений*

Расширенная поддержка интерпретируемых языков. Виджеты Plasma теперь можно создавать на JavaScript, Python и Ruby. Их затем можно распространять через специальные сервисы, такие как OpenDesktop.org. Добавлена поддержка GoogleGadgets и улучшена поддержка виджетов Mac OS X dashboard.

Доступны предварительные версии приложений KDE для Windows и Mac OS X. Также ведётся работа над поддержкой OpenSolaris, совершенствованием поддержки FreeBSD.

После выпуска Qt по лицензии LGPL, по которой также распространяются библиотеки KDE, разработчики коммерческого программного обеспечения смогут бесплатно пользоваться наработками сообщества.

### **2.2.3. Рабочий стол KDE, панель задач, главное меню, объекты на рабочем столе, контекстное меню**

Большинство рабочих столов содержат общие базовые компоненты. Этот рабочий стол должен показаться хорошо знакомым всем пользователям Windows.

Рабочая область рабочего стола содержит значки, в данном случае два значка. Чтобы выполнить определенное действие, щелкните мышью на значке, как в Windows. Вы можете настроить ваш рабочий стол, добавляя, удаляя и переставляя значки. Работа со значками описана далее в этой главе.

Внизу рабочего стола, изображенного на рисунке, находится панель. На самом деле вы можете переместить панель в любой край экрана: вверх, вниз, влево, вправо. К тому же вы можете иметь более чем одну панель.

На рис. 8. левая сторона панели содержит несколько значков. Вы можете настраивать набор этих значков, добавляя и удаляя их. Справа от значков расположена секция, содержащая четыре кнопки. Щелчками мыши на кнопках данной секции производится переключение между виртуальными рабочими столами, о чем рассказывается далее в этой главе. Справа от секции переключения рабочих столов находится секция, называемая панелью задач, в данной секции располагаются значки запущенных приложений. Справа от панели задач находятся значки для полезных системных инструментов. Часы показаны в самом правом углу панели. Вы можете реорганизовать панель, добавляя, удаляя и переставляя значки на ней.



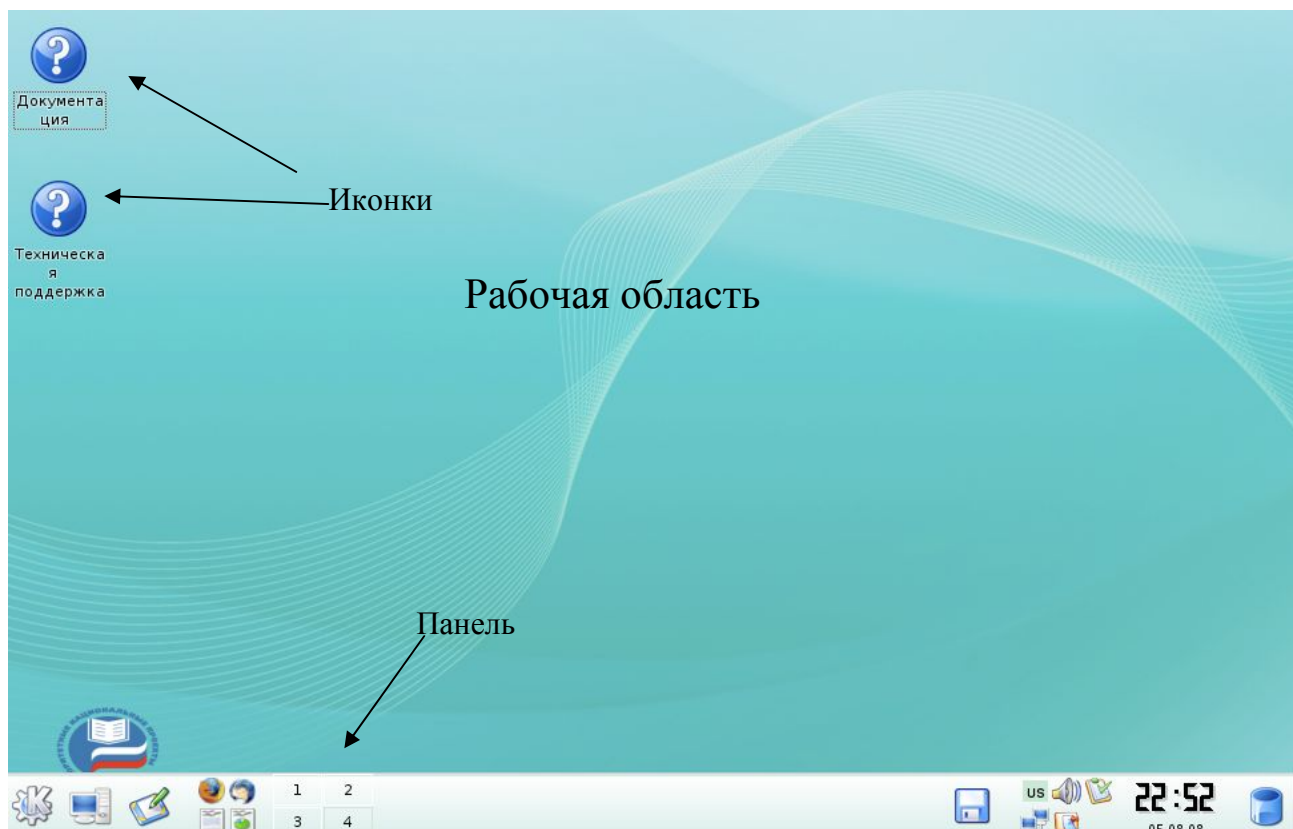


Рис. 7. Рабочий стол

Панель содержит секции:

- *Панель значков:* Секция слева, содержащая значки приложений. Приложения запускаются щелчком мыши на соответствующих значках. При наведении указателя мыши на значок будет показана всплывающая подсказка.
- *Панель переключения рабочих столов:* Секция, включающая, как правило, четыре значка, с помощью которых возможно переключение между виртуальными рабочими столами.
- *Панель задач:* Секция, в которой отображаются значки всех запущенных приложений. Панель рабочего стола KDE показывает одно открытое приложение - X Settings.
- *Системная область уведомлений:* Секция в правой стороне панели, содержащая значки полезных системных приложений, к которым вы имеете доступ прямо на панели. Часы находятся правее. Если вы наведете указатель мыши на часы, текущая дата будет отображена как всплывающая подсказка. Если вы щелкните мышью на часах, будет показан календарь месяца. Вы можете перемещаться в календаре по месяцам и годам. Щелкните правой кнопкой мыши на часах, на экране появится контекстное меню, позволяющее вам выполнить такие задачи, как настройка даты, времени, изменение способа отображения времени. Такой тип приложений обычно называется апплетами (от англ. applet).

- *Кнопка скрытия панели:* Маленькая стрелка на конце или на концах панели. Щелкните мышью на стрелке, чтобы скрыть панель, при этом останется только маленькая стрелка для восстановления панели.
- Обратите внимание на самую левую кнопку на панели. Она открывает главное меню системы.

*Рабочий стол* - это рабочая область с некоторым набором инструментов. Когда Вы хотите выполнить задачу, Вы запускаете приложение, которое открывается в окне на рабочем столе. Вы можете открыть множество окон сразу, в каждом из которых выполняется своя задача с помощью того или иного приложения. Ваш рабочий стол предлагает разные элементы, чтобы помочь вам в работе. Многие из этих элементов знакомы пользователям Windows.

*Значки.* Вы можете щелкнуть мышью на значке рабочего стола или панели, чтобы запустить приложение или открыть какой-либо адрес. В большинстве версий, значки рабочего стола открываются двойным щелчком мыши, а значки панели одинарным. Однако это настраиваемо, и в некоторых версиях значки рабочего стола открываются одним щелчком мыши. Вы можете добавлять, удалять и перестраивать значки на рабочем столе и панели, что обсуждается далее в этой главе.

*Меню.* Операционная система ПСПО предоставляет главное меню и различные контекстные меню. Вы можете добавлять, удалять и изменять элементы меню. Когда вы устанавливаете новое приложение, элемент меню приложения обычно добавляется в главное меню.

*Главное меню.* Содержит ярлыки доступных приложений и утилиты как пункты главного меню или элементы подменю. Главное меню ПСПО аналогично главному меню Windows. Главное меню открывается щелчком мыши на значке панели, обычно это самая левая кнопка. Значок главного меню - это большая буква К для рабочего стола KDE и изображение ступни для GNOME. Однако версии Linux часто используют свои отличительные значки как значки рабочих столов.

*Контекстные меню.* Меню, открываемые щелчком правой кнопки мыши на объекте. Если вы щелкнете правой кнопкой мыши на свободное место рабочего стола, вы увидите одно меню. Если вы щелкнете мышью на панели, это будет уже другое меню. Если вы щелкнете правой кнопкой мыши на значке, меню также будет отличаться. Меню содержат элементы управления, специфичные для выбранного объекта.

Во многих меню одна буква названия элемента управления подчеркнута. Вы можете ввести эту букву, чтобы выбрать элемент управления, нет необходимости выбирать его щелчком мыши. Некоторые меню имеют комбинации клавиш для вызова

содержащихся в них элементов управления, например комбинация **Alt+F9**. Это значит, что Вы можете нажать **Alt+F9** для выполнения элемента управления без открытия меню.

*Всплывающие подсказки.* Информация, показываемая вам, когда вы наводите указатель мыши на объект и удерживаете его непродолжительное время. Вы можете включать и выключать всплывающие подсказки.

*Манипулирование окнами.* Окна можно перемещать, минимизировать, максимизировать, изменять размер, закрывать, полностью сворачивать все окна.

*Перемещение.* Щелкните мышью на верхнюю панель окна и, удерживая нажатой левую кнопку, переместите окно, куда вы хотите. Другой способ - щелкнуть на кнопке в верхнем левом углу окна и в открывшемся контекстном меню выбрать **move** (Переместить).

*Изменение размера.* Щелкните мышью на границе окна и, удерживая нажатой левую кнопку мыши, измените размер, как вам нужно. Чтобы максимизировать окно, щелкните мышью на средней кнопке в правом верхнем углу окна.

*Максимизировать, минимизировать, показать рабочий стол.* Дважды щелкните мышью на заголовке окна, окно будет максимизировано. Чтобы вернуть окну прежний размер, также дважды щелкните мышью на заголовке окна. Щелкните мышью на левой кнопке в правом верхнем углу окна, чтобы минимизировать окно. Если ваша панель содержит значок **Show desktop** (Показать рабочий стол), щелкните на него мышью, чтобы свернуть все открытые окна. Если значок **Show desktop** (Показать рабочий стол) не присутствует на вашей панели, вы можете его добавить, как показано далее в этой главе.

*Закреть.* Щелкните мышью на самой правой кнопке в верхнем правом углу или выберите **close** (Закреть) в открывающемся меню верхнего левого угла.

*Внешний вид.* Внешний вид рабочего стола ПСПО очень хорошо настраивается. Например, заставка рабочего стола по умолчанию зависит от дистрибутива. Однако Вы можете изменить цвет, шаблон, дизайн, изображение на какие-либо другие. Вы можете изменять вид всех элементов рабочего стола, например границ окна, панелей заголовка, панелей. Пользователи Linux часто настраивают рабочие столы с помощью темы - унифицированного набора цветов, изображений и даже звуков, которые объединяют все части рабочего стола в единую систему. Некоторые темы включены в версии ПСПО, Вы можете их использовать. Другие доступны для скачивания из Интернета.

Поведение некоторых элементов рабочего стола также может быть изменено. Например, значки могут запускаться одним щелчком мыши, или двумя. При соответствующей настройке окна могут активизироваться или щелчком мыши или наведением указателя мыши. Всплывающие подсказки - информация, показываемая, когда

указатель мыши наведен на объект - могут быть включены или отключены. Помните, что практически все можно настроить.

## 2.2.4. Работа с окнами: перемещение, изменение размеров, понятие активного окна, переход из одного окна в другое

Каждое приложение, запускаемое в KDE, открывает одно или несколько окон. Вы можете управлять этими окнами. Обычное окно выглядит так:

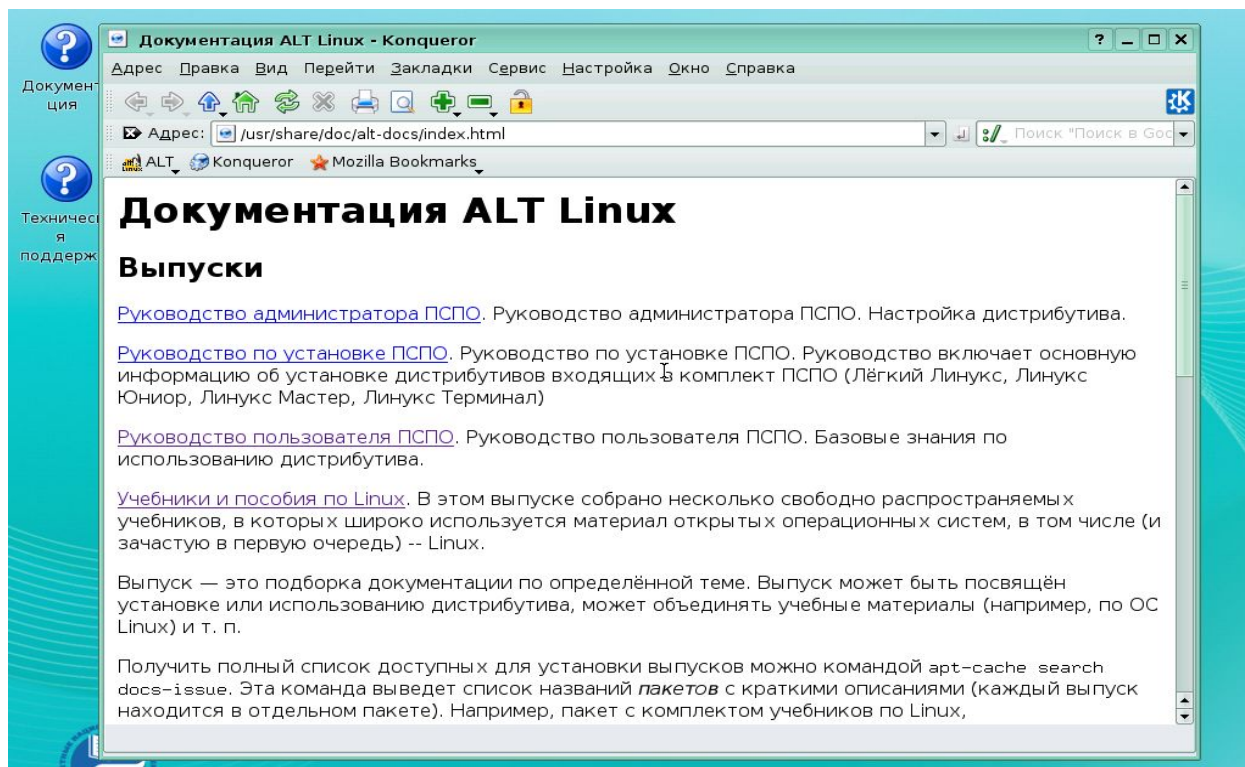


Рис. 8. Обычное окно в KDE

### *Переключение между окнами*

Непосредственно работать можно только с активным окном. Каждое новое окно становится активным автоматически, что позволяет сразу же начать работу с приложением. Активному окну передаются нажатия клавиш. Активным может быть только одно окно, которое можно отличить по цвету заголовка. (В теме KDE, принятой по умолчанию, у активного окна светло-синий заголовок, а у неактивных - серый.)

Для того чтобы начать работу с окном, его необходимо сделать активным. Есть два способа:

Щёлкните на требуемом окне левой кнопкой мыши. Окно станет активным и будет перемещено поверх остальных окон, если оно перекрывается ими.

Удерживая клавишу **Alt**, нажимайте **Tab**. Появится список доступных окон, одно из которых будет выделено. Нажимая клавишу **Tab**, можно выбрать требуемое окно. Когда вы отпустите клавишу **Alt**, выбранное окно будет сделано активным.

### *Перемещение окон*

Простейший способ упорядочить окна на рабочем столе - переместить их. Окна могут перекрывать друг друга. Есть три способа переместить окно:

- Переместите курсор мыши на заголовок окна, и, удерживая левую кнопку мыши, передвигайте его. Окно будет перемещаться вместе с курсором. Для того чтобы оставить окно на месте, отпустите кнопку мыши.
- Нажмите левой кнопкой мыши на крайнюю левую кнопку в заголовке окна (или правую кнопку мыши в середине заголовка окна). Откроется меню окна. Выберите пункт **Переместить**. Курсор мыши переместится в центр окна, и окно будет двигаться вместе с ним. Щёлкните левой кнопкой мыши, чтобы отпустить окно после перемещения на нужное место.
- Наведите курсор *на любое место окна*, которое требуется переместить, и нажмите клавишу Alt и левую кнопку мыши. Курсор изменит форму, и окно будет двигаться вместе с ним. Отпустите кнопку мыши, чтобы оставить окно на месте. Этот способ может быть особенно полезен, если заголовок окна находится за пределами экрана, что не позволяет применить другие методы.

#### *Изменение размера окна*

Есть несколько способов изменения размеров окна:

- Переместите курсор мыши к границе окна. Курсор примет форму двунаправленной стрелки. Удерживая левую кнопку, передвигайте мышь. Граница окна будет двигаться вместе с курсором. Перемещая верхнюю и нижнюю границы окна, можно изменять его высоту, перемещая левую и правую - его ширину. Для того чтобы изменять оба параметра одновременно, передвигайте угол окна (курсor мыши при наведении на него принимает форму диагонально расположенной двунаправленной стрелки).
- Откройте меню окна, нажав на крайнюю левую кнопку в заголовке. Выберите пункт **Изменить размер**. Курсоры мыши примет форму двунаправленной стрелки. Перемещая его, измените размер окна. Щёлкните левой кнопкой мыши, чтобы отпустить окно.
- Если граница окна или кнопка меню окна недоступны, используйте следующий способ. Удерживая клавишу **Alt**, нажмите правую кнопку мыши и перемещайте мышь. Размер окна будет изменяться. Закончив изменение размера, отпустите кнопку мыши.

Если требуется раскрыть окно на весь экран, нажмите кнопку **Распахнуть** - вторую справа кнопку в заголовке окна. Окно примет максимально возможный размер, если щёлкнуть на ней левой кнопкой мыши; при нажатии средней или правой кнопки мыши окно будет распахнуто по вертикали или горизонтали соответственно.

#### *Как скрыть окно*

Иногда необходимо освободить место на рабочем столе, не закрывая программу. В этом случае можно свернуть окно в значок на панели задач или в заголовок.

- Для того чтобы *свернуть окно в значок*, нажмите кнопку **Свернуть** - третью справа в заголовке окна. Окно будет скрыто, но выполнение программы не прекратится. Для того чтобы вновь развернуть окно, щёлкните мышью на соответствующем значке панели задач. Для восстановления окна также можно использовать комбинацию клавиш **Alt+Tab**;
- Для того чтобы *свернуть окно в заголовок* (т. е. скрыть его рабочую область, оставив только заголовок), необходимо дважды щёлкнуть на заголовке левой кнопкой мыши. Для того чтобы восстановить окно, дважды щёлкните на заголовке ещё раз.

#### *Расположить каскадом*

Бывает, что рабочий стол загромождён множеством окон. Щёлкните средней кнопкой мыши на рабочем столе и в открывшемся меню выберите пункт **Расположить каскадом** - окна будут расположены одно за другим, начиная с левого верхнего угла экрана.

#### *Выстроить окна*

Если вы выберете этот пункт меню, KDE расположит окна так, чтобы сделать доступным как можно большую часть каждого из них. Например, если открыты четыре окна, после выбора этого пункта они будут перемещены в углы экрана. Щёлкните средней кнопкой мыши на рабочем столе и в открывшемся меню выберите пункт **Выстроить окна**.

#### *Как закрыть окно*

После того, как вы завершили работу с приложением, требуется остановить его выполнение и закрыть окно. И в этом случае есть несколько различных способов:

- Щёлкните на крайней правой кнопке в заголовке окна. Если вы изменили какой-либо документ с помощью этого приложения, программа спросит, хотите ли вы **Сохранить изменения, Отклонить** их либо **Отменить выход из приложения**.
- Выберите пункт меню **Файл-Выход**. Откроется то же самое окно с кнопками **Сохранить, Отклонить и Отмена**.
- Щёлкните правой кнопкой мыши на значке, соответствующем окну, на панели KDE (Kicker), и выберите **Закрыть**. Программа предложит вам сохранить изменённые документы.
- Нажмите **Alt+F4**. Если какие-либо документы были изменены, окно подтверждения также будет открыто.

#### *Переключение между виртуальными рабочими столами*

Для того чтобы перейти к следующему виртуальному рабочему столу, используйте комбинацию клавиш **Ctrl+Tab**, аналогично комбинации **Alt+Tab** для переключения

между окнами (см. Переключение между окнами). Появится маленькое всплывающее окно с изображениями рабочих столов, одно из которых будет выделено. Отпустите клавишу **Ctrl**, чтобы перейти к выделенному рабочему столу. Нажимайте клавишу **Tab**, удерживая **Ctrl**, чтобы переключаться между столами.

#### *Окна и виртуальные рабочие столы*

Переместить окно с одного виртуального рабочего стола на другой можно с помощью пункта **На рабочий стол** меню окна: в открывшемся подменю просто выберите требуемый рабочий стол. Для того чтобы сделать окно доступным на всех рабочих столах, установите переключатель **Все рабочие столы**.

### **2.2.5. Справочная система**

В дистрибутив ПСПО входит комплект документации в печатном виде (если Вы приобрели дистрибутив в розницу) и в электронном виде (в формате HTML).

Не пренебрегайте чтением документации: она поможет вам избежать многих сложностей, сэкономить массу времени и усилий при установке, настройке и администрировании системы, поможет найти нужное для работы приложение и быстро разобраться в нём. Даже если вы — опытный пользователь Linux, в документации найдутся полезные для вас сведения об особенностях дистрибутива ПСПО.

Доступ к графической документации можно получить, выбрав в главном меню К пункт **Справка**. Откроется окно Центр Справки KDE. Тут вы сможете прочитать руководства по использованию и настройке KDE, по работе со стандартными приложениями.

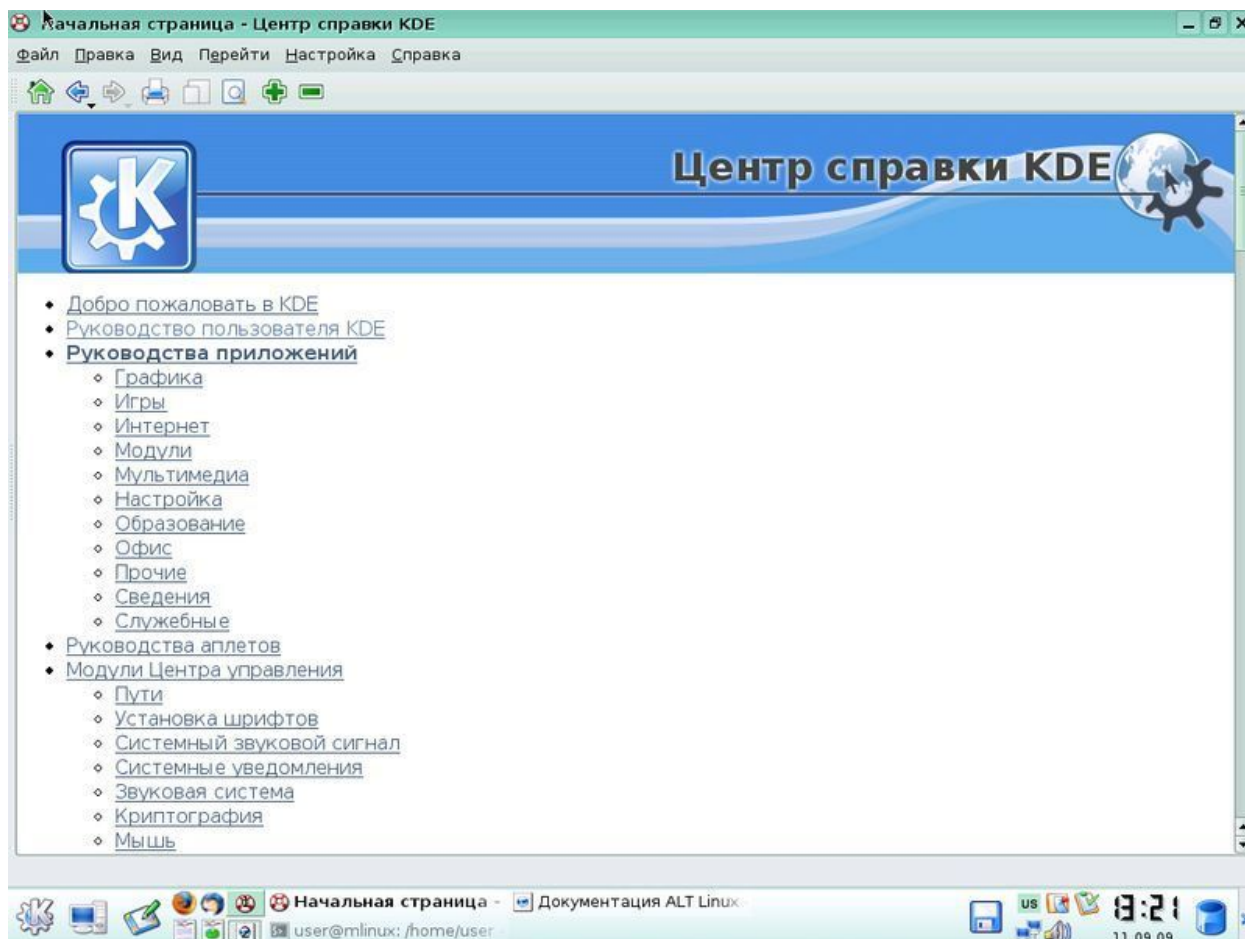


Рис. 9. Центр Справки KDE

Дополнительные программы устанавливаются в систему с помощью так называемых пакетов (подробнее в разделе по «Установка и настройка прикладного программного обеспечения»). Вместе с каждым пакетом устанавливается документация. Основное место для хранения разнообразной документации, в основном на английском языке, — каталог **/usr/share/doc**. Особое внимание обратите на HOWTO (от англ. how to — «как сделать») — собрание практических рекомендаций по самым различным вопросам, связанным с использованием Linux.

Каждый пакет также содержит поставляемую вместе с включённым в него ПО документацию, располагающуюся обычно в каталоге **/usr/share/doc/имя\_пакета**. Например, документация к пакету `foo-1.0-alt1` находится в `/usr/share/doc/foo-1.0-alt1`. Для получения полного списка файлов документации, относящихся к пакету, воспользуйтесь командой **rpm -qld имя\_пакета**.

В документации к каждому пакету вы можете найти такие файлы как README, FAQ, TODO, ChangeLog и другие. В файле README содержится основная информация о программе — имя и контактные данные авторов, назначение, полезные советы и пр. FAQ содержит ответы на часто задаваемые вопросы; этот файл стоит прочитать в первую очередь, если у Вас возникли проблемы или вопросы по использованию программы,



поскольку большинство проблем и сложностей типичны, вполне вероятно, что в FAQ Вы тут же найдёте готовое решение. В файле TODO записаны планы разработчиков на реализацию той или иной функциональности. В файле ChangeLog записана история изменений в программе от версии к версии.

Адреса сайтов в Интернет, посвящённых отдельным программным продуктам, указаны в информационных заголовках соответствующих пакетов, их можно получить с помощью команды **rpm -qi имя\_пакета**.

С помощью Konqueror Вы можете просматривать файлы справки по KDE и руководства по UNIX (Man и Info), не запуская дополнительных приложений.

Чтобы получить справку по какому-либо приложению KDE, введите в строку адреса **help:/название программы** (например, **help:/kmail**).

Konqueror также позволяет просматривать руководства UNIX® (Man); например, чтобы просмотреть руководство по команде **touch**, введите в строку адреса **man:/touch** или **#touch**.

Чтобы просмотреть документацию UNIX (Info), введите в строке адреса **info:/dir**, и Konqueror покажет список всех доступных файлов документации.

Замечание. К сожалению, файлы справки KDE хранятся таким образом, что их невозможно просматривать с помощью других обозревателей. Если использование Konqueror невозможно, Вы всегда можете найти копию файлов справки в Интернете по адресу <http://docs.kde.org>.

Экранная документация. Помимо поставляемой ПСПО документации и дополнительной литературы, всё программное обеспечение, входящее в дистрибутив, снабжается собственной документацией. Стандартный способ получить документацию по той или иной программе, функции или файлу, установленным в системе, унаследованный ПСПО от ОС UNIX, — это команда **man**, отображающая экранную документацию, иногда называемую «страницы руководства» (буквальный перевод англ. manual pages). Для того, чтобы прочесть экранную документацию по программе, достаточно в любой командной строке набрать **man имя\_программы**. Например, команда **man man** выдаёт справку по пользованию самой командой **man**. Если Вы точно не знаете, как называется необходимая вам программа, может помочь поиск по ключевому слову при помощи команд **argros** и **whatis**. Например, если вы введёте команду **argros mail**, Вы увидите список всех программ, в кратком описании которых упоминается слово **mail**. Разница между командами заключается в том, что **whatis** ищет только по названиям руководств, а **argros** ещё и по кратким описаниям.

В технической документации по UNIX и Linux принят стандартный формат ссылки на экранную документацию, выводимую по команде `man`. Например, запись `apt(8)`, отсылает к экранной документации по программе `apt`, вызываемой командой `man apt` (цифра в скобках обозначает раздел, к которому относится данная документация, её требуется вводить только в том случае, если есть несколько руководств с одним именем, но в разных разделах, например `man 8 apt`).

## 2.3. Работа с файловой системой

### 2.3.1. Организация данных

*Файл* — это понятие, привычное любому пользователю компьютера. Для пользователя каждый файл — это отдельный предмет, у которого есть начало и конец и который отличается от всех остальных файлов именем и расположением («как называется» и «где лежит»). Как и любой предмет, файл можно создать, переместить и уничтожить, однако без внешнего вмешательства он будет сохраняться неизменным неопределенно долгое время. Файл предназначен для хранения данных любого типа — текстовых, графических, звуковых, исполняемых программ и многого другого. Аналогия файла с предметом позволяет пользователю быстро освоиться при работе с данными в операционной системе.

*Для операционной системы Linux файл* — не менее важное понятие, чем для её пользователя: все данные, хранящиеся на любых носителях, обязательно находятся внутри какого-нибудь файла, в противном случае они просто недоступны ни для операционной системы, ни для её пользователей. Более того, все устройства, подключённые к компьютеру (начиная клавиатурой и заканчивая любыми внешними устройствами, например, принтерами и сканерами) Linux представляет как файлы (так называемые файлы для работы с устройствами, или сокращенно файлы-устройства). Конечно, файл, содержащий обычные данные, сильно отличается от файла, предназначенного для обращения к устройству, поэтому в Linux определены несколько различных типов файлов. В основном пользователь имеет дело с файлами трёх типов: обычными **файлами**, предназначенными для хранения данных, **каталогами** и **файлами-ссылками**.

*Файловая система с точки зрения пользователя* — это «пространство», в котором размещаются файлы, наличие файловой системы позволяет определить не только «как называется файл», но и «где он находится». Различать файлы только по имени было бы слишком неэффективным: про каждый файл приходилось бы помнить, как он называется и при этом заботиться о том, чтобы имена никогда не повторялись. Более того, необходим механизм, позволяющий работать с группами тематически связанных между собой

файлов (например, компонентов одной и той же программы или разных главы одной диссертации). Иначе говоря, файлы нужно систематизировать.

Linux может работать с различными типами файловых систем, которые различаются списком поддерживаемых возможностей, производительностью в разных ситуациях, надёжностью и другими признаками.

Большинство современных файловых систем используют в качестве основного организационного принципа каталоги. Каталог — это список ссылок на файлы или другие каталоги. Принято говорить, что каталог содержит в себе файлы или другие каталоги, хотя в действительности он только ссылается на них, физическое размещение данных на диске обычно никак не связано с размещением каталога. Каталог, на который есть ссылка в данном каталоге, называется подкаталогом или вложенным каталогом. Каталог в файловой системе более всего напоминает библиотечный каталог, содержащий ссылки на объединённые по каким-то признакам книги и другие разделы каталога (файлы и подкаталоги). Ссылка на один и тот же файл может содержаться в нескольких каталогах одновременно, это может сделать доступ к файлу более удобным. В файловой системе Ext2 каждый каталог — это отдельный файл особого типа («d», от англ. «directory»), отличающийся от обычного файла с данными: в нём могут содержаться только ссылки на другие файлы и каталоги.

Довольно часто вместо термина каталог можно встретить папка (англ. folder). Этот термин хорошо вписывается в представление о файлах как о предметах, которые можно раскладывать по папкам, однако часть возможностей файловой системы, которая противоречит этому представлению, таким образом, затемняется. В частности, с термином «папка» плохо согласуется то, что ссылка на файл может присутствовать одновременно в нескольких каталогах, файл может быть ссылкой на другой файл и т. д. В Linux эти возможности файловой системы весьма важны для эффективной работы, поэтому будем всюду использовать более подходящий термин «каталог».

В файловой системе, организованной при помощи каталогов, на любой файл должна быть ссылка как минимум из одного каталога, в противном случае файл просто не будет доступен внутри этой файловой системы, иначе говоря, не будет существовать.

Главные отличительные признаки файлов и каталогов — их имена. В Linux имена файлов и каталогов могут быть длиной не более 256 символов, и могут содержать любые символы, кроме «/». Причина этого ограничения очевидна: этот символ используется как разделитель имён в составе пути, поэтому не должен встречаться в самих именах. Причём Linux всегда различает прописные и строчные буквы в именах файлов и каталогов, поэтому «methody», «Methody» и «METHODY» будут тремя разными именами.

Есть несколько символов, допустимых в именах файлов и каталогов, которые, при этом, нужно использовать с осторожностью. Это — так называемые спецсимволы «\*», «\», «&», «<», «>», «;», «(», «)», «|», а также пробелы и табуляции. Дело в том, что эти символы имеют особое значение для любой командной оболочки, поэтому нужно будет специально позаботиться о том, чтобы командная оболочка воспринимала эти символы как часть имени файла или каталога

Как можно было заметить, пока во всех встречавшихся именах файлов и каталогов употреблялись только символы латинского алфавита и некоторые знаки препинания. Это не случайно и вызвано желанием обеспечить, чтобы приводимые примеры совершенно одинаково выглядели на любых системах. В Linux в именах файлов и каталогов допустимо использовать любые символы любого языка, однако такая свобода требует жертв. Дело в том, что с давних пор каждый символ (буква) каждого языка традиционно представлялся в виде одного байта. Такое представление накладывает очень жёсткие ограничения на количество букв в алфавите: их может быть не больше 256, а за вычетом управляющих символов, цифр, знаков препинания и прочего — и того меньше. Обширные алфавиты (например, иероглифические японский и китайский) пришлось заменять упрощённым их представлением. Вдобавок, первые 128 символов из этих 256 лучше всегда оставлять неизменными, соответствующими стандарту ASCII, включающему латиницу, цифры, знаки препинания и наиболее популярные символы из тех, что встречаются на клавиатуре печатной машинки. Интерпретация остальных 128 символов зависит от того, какая кодировка установлена в системе. Например, в русской кодировке KOI8-R 228-й символ такой таблицы соответствует букве «Д», а в западноевропейской кодировке ISO-8859-1 этот же символ соответствует букве «а» с двумя точками на ней (как у нашей буквы «ё»).

Имена файлов, записанные на диск в одной кодировке, выглядят нелепо, если при просмотре каталога была установлена другая. Хуже того. Многие кодировки заполняют диапазон символов с номерами от 128 до 255 не полностью, поэтому соответствующего символа может вообще не быть. Это означает, что ввести такое искажённое имя файла с клавиатуры (например, для того, чтобы его переименовать) напрямую не удастся, придётся пускаться на разные ухищрения. Наконец, многие языки, в том числе и русский, исторически имеют несколько кодировок. К сожалению, в настоящее время нет стандартного способа указывать кодировку прямо в имени файла, поэтому в рамках одной файловой системы стоит придерживаться единой кодировки при именовании файлов. Например, вы можете получить проблемы, если с файловой системой одновременно

работает несколько пользователей по сети, используя разные операционные системы, с разными кодировками.

Существует универсальная кодировка, включающая символы всех письменностей мира — UNICODE. Стандарт UNICODE в настоящее время получает всё большее распространение и претендует на статус общего для всех текстов, хранящихся в электронной форме. Однако пока он не достиг желаемой универсальности, особенно в области имён файлов. Один символ в UNICODE может занимать больше одного байта — и в этом главный его недостаток, так как множество полезных прикладных программ, отлично работающих с однобайтными кодировками, необходимо основательно или даже полностью перерабатывать для того, чтобы научить их обращаться с UNICODE. Возможно, причина недостаточной распространённости этой кодировки также и в том, что UNICODE — очень громоздкий стандарт, и он может оказаться неэффективным при работе с файловой системой, где скорость и надёжность обработки — очень существенные качества. *Это не означает, что называя файлы, не следует использовать языки, отличные от английского.* Пока точно известно, в какой кодировке задано имя файла — проблем не возникнет.

Многим пользователям знакомо понятие расширение — часть имени файла после точки, обычно ограничивающаяся несколькими символами и указывающая на тип содержащихся в файле данных. В файловой системе Linux нет никаких предписаний по поводу расширения: в имени файла может быть любое количество точек (в том числе и ни одной), а после последней точки может быть любое количество символов. Хотя расширения не обязательны и не навязываются технологией в Linux, они широко используются: расширение позволяет человеку или программе, не открывая файл, только по его имени определить, какого типа данные в нём содержатся. Однако нужно учитывать, что расширение — это только набор соглашений по наименованию файлов разных типов. Строго говоря, данные в файле могут не соответствовать заявленному расширению по той или иной причине, поэтому, всецело полагаться на расширение просто нельзя.

Определить тип содержимого файла можно и на основании самих данных. Многие форматы предусматривают указание в начале файла, как следует интерпретировать дальнейшую информацию: как программу, исходные данные для текстового редактора, страницу HTML, звуковой файл, изображение или что-то другое. В распоряжении пользователя ПСПО всегда есть утилита **file**, которая предназначена именно для определения типа данных, содержащихся в файле.

### 2.3.2. Иерархическая структура файлов и папок

Понятие каталога позволяет систематизировать все объекты, размещённые на носителе данных (например, на диске). В большинстве современных файловых систем используется иерархическая модель организации данных: существует один каталог, объединяющий все данные в файловой системе — это «корень» всей файловой системы, корневой каталог. Корневой каталог может содержать любые объекты файловой системы, и в частности, подкаталоги (каталоги первого уровня вложенности). Те, в свою очередь, также могут содержать любые объекты файловой системы и подкаталоги (второго уровня вложенности) и т. д.. Таким образом, всё, что записано на диске — файлы, каталоги и специальные файлы — обязательно «принадлежит» корневому каталогу: либо непосредственно (содержится в нём), либо на некотором уровне вложенности.

Иерархию вложенных друг в друга каталогов можно соотнести с иерархией данных в системе: объединить тематически связанные файлы в каталог, тематически связанные каталоги — в один общий каталог и т. д. Если строго следовать иерархическому принципу, то чем глубже будет уровень вложенности каталога, тем более частным признаком должны быть объединены содержащиеся в нём данные. Если этому принципу не следовать, то вскоре окажется гораздо проще складывать все файлы в один каталог и искать нужный среди них, чем проделывать такой поиск по всем подкаталогам системы. Однако в этом случае, о какой бы то ни было систематизации файлов, говорить не приходится.

Структуру файловой системы можно представить наглядно в виде дерева, «корнем» которого является корневой каталог, а в вершинах расположены все остальные каталоги. На Рис.10 изображено дерево каталогов, курсивом обозначены имена файлов, прямым начертанием — имена каталогов.

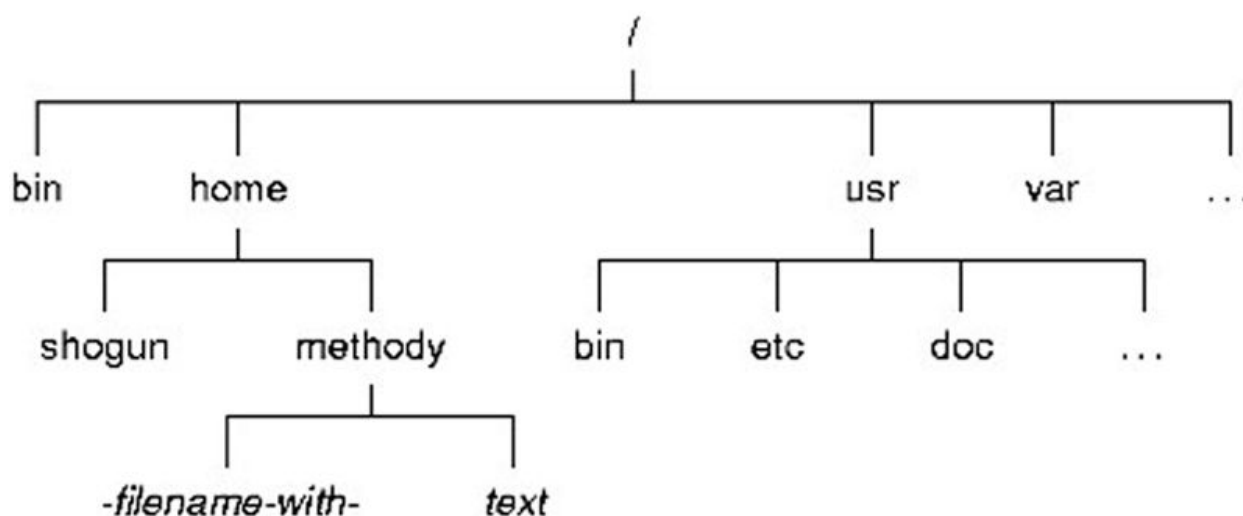


Рис.10. Дерево каталогов в Linux

В любой файловой системе Linux всегда есть только один корневой каталог, который называется «/». Пользователь Linux всегда работает с единым деревом каталогов, даже если разные данные расположены на разных носителях: нескольких жёстких или сетевых дисках, съёмных дисках, CD-ROM и т. п. Для того, чтобы подключать и отключать файловые системы на разных устройствах в одно общее дерево, используются процедуры монтирования и размонтирования. После того, как файловые системы на разных носителях подключены к общему дереву, содержащиеся на них данные доступны так, как если бы все они составляли единую файловую систему: пользователь может даже не знать, на каком устройстве какие файлы хранятся.

Положение любого каталога в дереве каталогов точно и однозначно описывается при помощи полного пути. Полный путь всегда начинается от корневого каталога и состоит из перечисления всех вершин, встретившихся при движении по рёбрам дерева до искомого каталога включительно. Названия соседних вершин разделяются символом «/» («слэш»). В Linux полный путь, например, до каталога «academyit» в файловой системе, записывается следующим образом: сначала символ «/», обозначающий корневой каталог, затем к нему добавляется «home», затем разделитель «/», за которым следует название искомого каталога «academyit», в результате получается полный путь «/home/ academyit».

Расположение файла в файловой системе аналогичным образом определяется при помощи полного пути, только последним элементом в данном случае будет не название каталога, а название файла. Например, полный путь до созданного academyit файла start» будет выглядеть так: «/home/ academyit /start».

Организация каталогов файловой системы в виде дерева не допускает появления циклов: т. е. каталог не может содержать в себе каталог, в котором содержится сам. Благодаря этому ограничению полный путь до любого каталога или файла в файловой системе всегда будет конечным.

Опишем кратко, что находится в каждом из подкаталогов корневого каталога. Мы не будем приводить полные списки файлов для каждого описываемого каталога.

**/bin** - Название этого каталога происходит от слова «binaries» («двоичные», «исполняемые»). В этом каталоге находятся исполняемые файлы самых необходимых утилит. Сюда попадают такие программы, которые могут понадобиться системному администратору или другим пользователям для устранения неполадок в системе или при восстановлении после сбоя.

**/boot** - «Boot» — загрузка системы. В этом каталоге находятся файлы, необходимые для самого первого этапа загрузки: загрузки ядра и, обычно, само ядро.

Пользователю практически никогда не требуется непосредственно работать с этими файлами.

**/dev** - В этом каталоге находятся все имеющиеся в системе файлы для работы с устройствами. Например, файлы `/dev/ttyN` соответствуют виртуальным консолям, где N — номер виртуальной консоли. Данные, введённые пользователем на первой виртуальной консоли, система считывает из файла `/dev/tty1`, в этот же файл записываются данные, которые нужно вывести пользователю на эту консоль. В таких файлах в действительности не хранятся никакие данные, при их помощи данные передаются. Подробнее о принципе работы с файлами для обращения к устройствам речь пойдёт в разделе Работа с внешними устройствами.

**/etc** - Каталог для системных конфигурационных файлов. Здесь хранится информация о специфических настройках данной системы: информация о зарегистрированных пользователях, доступных ресурсах, настройках различных программ. Подробно системные конфигурационные файлы будут рассмотрены в лекции Конфигурационные файлы.

**/home** - Здесь расположены каталоги, принадлежащие пользователям системы — домашние каталоги, отсюда и название «home». Отделение всех файлов, создаваемых пользователями, от прочих системных файлов даёт очевидное преимущество: серьёзное повреждение системы или необходимость обновления не затронет наиболее ценной информации — пользовательских файлов.

**/lib** - Название этого каталога — сокращение от «libraries» (англ. «библиотеки»). Библиотеки — это собрания наиболее стандартных функций, необходимых многим программам: операций ввода/вывода, рисования элементов графического интерфейса и проч. Чтобы не включать эти функции в текст каждой программы, используются стандартные функции библиотек — это значительно экономит место на диске и упрощает написание программ. В этом каталоге содержатся библиотеки, необходимые для работы наиболее важных системных утилит (размещённых в `/bin` и `/sbin`).

**/mnt** - Каталог для монтирования (от англ. «mount») — временного подключения файловых систем, например, на съёмных носителях (CD-ROM и др.). Подробно о монтировании файловых систем речь пойдёт в лекции Работа с внешними устройствами.

**/proc** - В этом каталоге все файлы «виртуальные» — они располагаются не на диске, а в оперативной памяти. В этих файлах содержится информация о всех программах (процессах), выполняемых в данный момент в системе.

**/root** - Домашний каталог администратора системы — пользователя root. Смысл размещать его отдельно от домашних каталогов остальных пользователей состоит в том,



что /home может располагаться на отдельном устройстве, которое не всегда доступно (например, на сетевом диске), а домашний каталог root должен присутствовать в любой ситуации.

**/sbin** - Каталог для важнейших системных утилит (название каталога — сокращение от «system binaries»): в дополнение к утилитам /bin здесь находятся программы, необходимые для загрузки, резервного копирования, восстановления системы. Полномочия на исполнение этих программ есть только у системного администратора.

**/tmp** - Этот каталог предназначен для временных файлов: в таких файлах программы хранят промежуточные данные, необходимые для работы. После завершения работы программы временные файлы теряют смысл и должны быть удалены. Обычно каталог /tmp очищается при каждой загрузке системы.

**/usr** - Каталог /usr — это «государство в государстве». Здесь можно найти такие же подкаталоги bin, etc, lib, sbin, как и в корневом каталоге. Однако в корневой каталог попадают только утилиты, необходимые для загрузки и восстановления системы в аварийной ситуации, все остальные программы и данные располагаются в подкаталогах /usr. Прикладных программ в современных системах обычно установлено очень много, поэтому этот раздел файловой системы может быть очень большим.

**/var** - Название этого каталога — сокращение от «variable» («переменные» данные). Здесь размещаются те данные, которые создаются в процессе работы разными программами и предназначены для передачи другим программам и системам (очереди печати и электронной почты и др.) или для сведения системного администратора (системные журналы, содержащие протоколы работы системы). В отличие от каталога /tmp сюда попадают те данные, которые могут понадобиться после того, как создавшая их программа завершила работу.

Рекомендации стандарта по размещению файлов и каталогов основываются на принципе разносить в разные подкаталоги файлы, которые по-разному используются в системе. По типу использования файлов их можно разделить на следующие группы:

*Пользовательские файлы* — это все файлы, созданные пользователем и не принадлежащие ни одному из компонентов системы. О пользе разграничения пользовательских и системных файлов речь уже шла выше.

*К неизменным файлам* относятся все статические компоненты программного обеспечения: библиотеки, исполняемые файлы и др. — всё, что не изменяется само без вмешательства системного администратора. *Изменяющиеся* — это те, которые изменяются без вмешательства человека в процессе работы системы: системные журналы,

очереди печати и пр. Выделение неизменных файлов в отдельную структуру (например, /usr) позволяет использовать соответствующую часть файловой системы в режиме «только чтение», что уменьшает вероятность случайного повреждения данных и позволяет использовать для хранения этой части файловой системы CD-ROM и другие носители, доступные только для чтения.

### **2.3.3. Работа с файлами и папками: создание файлов, папок, копирование, перемещение, переименование, удаление, восстановление, поиск.**

Для просмотра файлов в KDE используется стандартная программа, называемая Konqueror.

Программа Konqueror устроена очень просто. В левой части ее окна содержится дерево папок (каталогов), а в правой — отображается содержимое выделенной папки. Щелчком мыши на какой-либо папке можно открыть ее содержимое. Если щелкнуть мышью на имени какого-либо файла, можно просмотреть этот файл.

Если необходимо, можно настроить систему так, чтобы файлы открывались от двойного щелчка мыши, а не от одинарного. При этом программа Konqueror отображает текстовые файлы, изображения, веб-страницы и еще некоторые типы файлов. Для отображения остальных файлов автоматически открывается та программа, которая «понимает» формат выбранного файла.

Так как Konqueror выполняет функцию как менеджера файлов, так и обозревателя Интернета, он автоматически переключается в необходимый режим в процессе работы. Однако иногда бывает удобно указать режим при запуске.

1. Если у вас на рабочем столе или на панели есть значок в виде компьютера, щёлкните по нему левой кнопкой, при необходимости выберите уровень просмотра, например **Документы** и Konqueror запустится в режиме менеджера файлов.
2. Также можно щёлкнуть по значку в виде глобуса (на рабочем столе или панели), и Konqueror будет запущен в режиме обозревателя Интернета.
3. В главном меню (**К**) выберите пункт **Интернет-Веб-браузер Konqueror**, и Konqueror будет запущен в режиме обозревателя; при выборе пункта **Домой** он запустится в режиме менеджера файлов.
4. При нажатии **Alt+F2** появится окно **Выполнить программу**; введите в нём konqueror (маленькими буквами) и нажмите Enter или кнопку Выполнить; Konqueror будет запущен в режиме обозревателя.

Если Konqueror запустился в режиме просмотра файлов, перейти в режим обозревателя можно, введя любой адрес Интернета, например <http://www.konqueror.org>.

Konqueror автоматически запускается в режиме менеджера файлов, когда вы щёлкаете на рабочем столе по значку, который представляет собой каталог, диск или Корзину.

Рассмотрим основные части окна Konqueror. В самом верху находится заголовок окна Konqueror; он работает так же, как и заголовки других программ. Щёлкните по нему правой кнопкой, и вы сможете воспользоваться контекстным меню заголовка окна.

Под заголовком находится главное меню. Щёлкнув мышью по одному из его пунктов, вы вызовете ещё одно, вертикальное, меню; его можно вызвать также с помощью комбинации клавиш **Alt+подчёркнутая** буква (в названии нужного пункта). Например, нажав **Alt+П**, вы вызовете меню Правка.

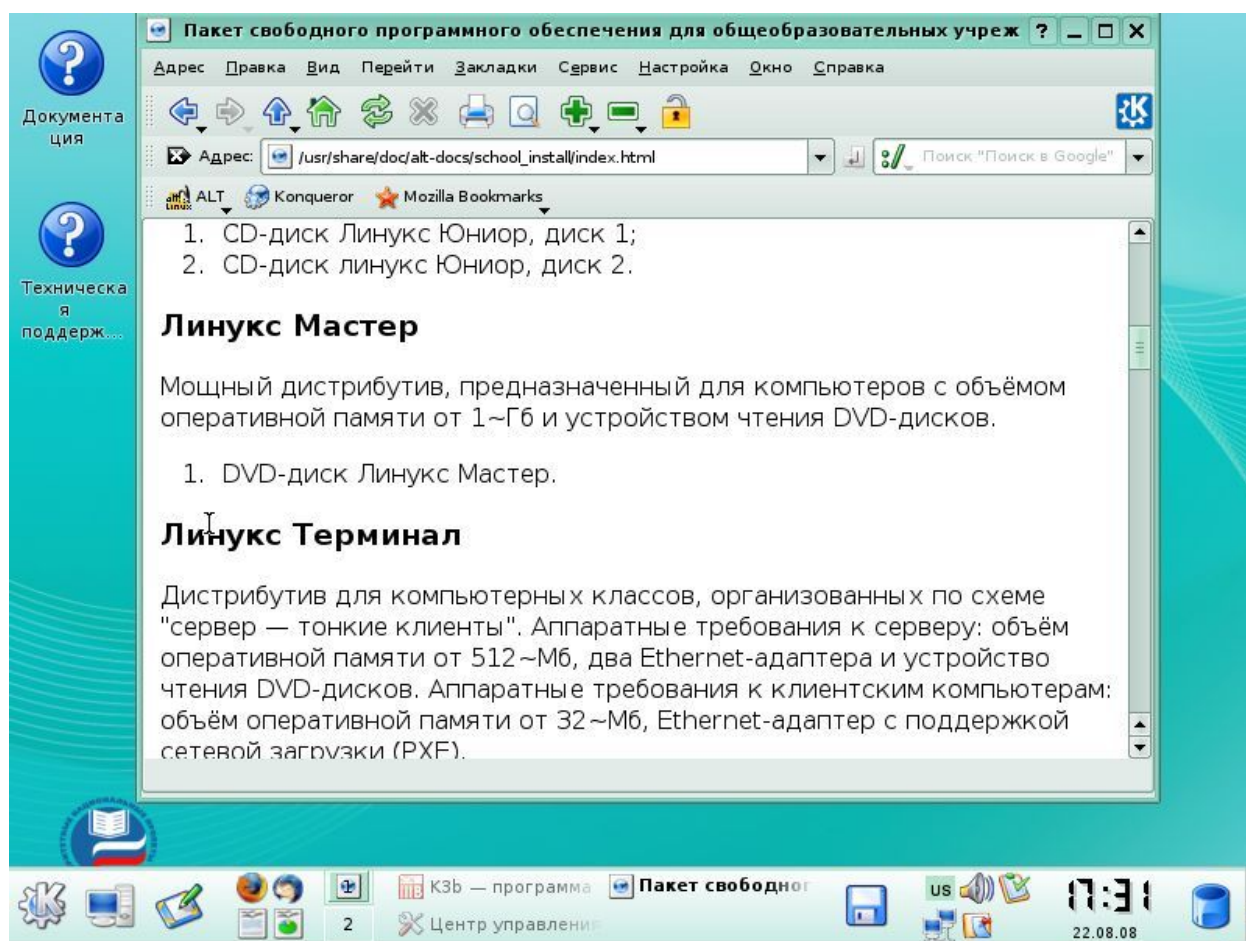


Рис. 11. Konqueror

На панели инструментов находятся кнопки для наиболее распространённых действий. У некоторых кнопок, как, например, **Вверх** и **Назад** (Вы можете увидеть их на рисунке выше), в нижем левом углу есть небольшой чёрный треугольник. Если вы нажмёте такую кнопку и будете удерживать её нажатой, появится выпадающее меню.

Щёлкнув правой кнопкой мыши по панели инструментов, Вы вызовете контекстное меню панели инструментов, в котором сможете настроить её вид и расположение.

В строке адреса показывается адрес текущего каталога, страницы в Интернете или файла. Чтобы перейти к другому адресу, введите его в это поле и нажмите Enter на клавиатуре или кнопку **Перейти справа**. С помощью чёрной кнопки с крестиком слева, вы можете очистить поле ввода.

Под строкой адреса находится панель закладок (см. рисунок выше). Сюда можно помещать адреса часто посещаемых страниц. Подробнее об этом можно прочитать в разделе **Работа с закладками**.

Снизу от меню, панели инструментов и панели закладок находится главная часть окна Konqueror, в которой отображается каталог, веб-страница, документ или рисунок. С помощью меню **Окно** можно разделить главное окно на два или более (что часто бывает удобно для перетаскивания элементов из одного окна в другое) или указать, что все окна должны отображаться во вкладках в одном окне Konqueror.

В нижней части окна находится строка состояния; в ней отображается общая информация об объекте, на который вы указываете мышью. Если Вы разделили окно на несколько панелей, у каждой из них будет своя строка состояния; на ней также будет показан индикатор, который становится зелёным, если панель активна. Щелчок по строке состояния правой кнопкой мыши позволяет вызвать контекстное меню.

#### *Замечание*

Ничего страшного, если у вас Konqueror выглядит не совсем так, как показано на рисунке: его внешний вид можно изменить на любой вкус. Например:

1. С помощью меню **Настройка** Вы можете включить или отключить панель меню, панель инструментов и закладок, строку адреса и даже добавить ещё одну панель.
2. Вы можете временно отключить панель, щёлкнув мышью по области с вертикальными полосками в левой крайней части панели. Также можно перемещать панели, нажав и удерживая на этих полосках левую кнопку мыши.

На рисунке не показана дополнительная боковая панель.

Чтобы получить справку о каком-либо элементе окна, выберите пункт **меню Помощь-Что это?**, нажмите комбинацию клавиш **Shift+F1** или кнопку со знаком вопроса в верхнем правом углу окна. Указатель мыши примет форму стрелки со знаком вопроса.

После этого щёлкните по тому элементу, по которому вы хотели бы получить справку, и вам будет представлено достаточно подробное описание его функций. (Эта возможность присутствует практически во всех окнах Konqueror.)

Расположение любого файла или каталога в этой древовидной структуре можно описать с помощью «пути»: это список каталогов, которые необходимо пройти, чтобы найти нужный файл. Например, `/home/ram` означает подкаталог `ram` каталога `/home`,

находящегося в корневом каталоге; а `/home/pam/words.txt` — файл `words.txt` в данном каталоге. Начальная косая черта (`«/»`) в приведённых путях означает корневой каталог.

Все имеющиеся у вас файлы, включая те, которые находятся на других разделах жёсткого диска, на гибком диске или на компакт-диске, доступны через корневой каталог, а их точное местоположение зависит от того, как настроена ваша система.

В системе ПСПО у каждого пользователя есть свой «домашний» каталог, в котором хранятся его личные файлы и настройки. Примеры таких каталогов — `/home/vasya` или `/home/lena`. Домашний каталог часто условно обозначают значком `~`. Так, например, путь `~/letter.txt` указывает на файл `letter.txt` в домашнем каталоге.

Заметьте, что часто вместо термина «каталог» используют слово «папка». Также в старой документации может встретиться термин «директорий».

«Путь» к текущему каталогу показан в заголовке окна и в строке адреса; заметьте, что в строке адреса путь приведён с префиксом `«file:»`, чтобы указать, что просматриваемый вами каталог — это часть обычной файловой системы вашего компьютера (например `file:/home/pam`), а не адрес Интернета.

Вы можете выбрать режим отображения файлов и папок с помощью меню Вид-Режим. Доступны следующие варианты:

1. *Значки.* Для каждого файла Konqueror показывает его имя и соответствующий значок.
2. *Список.* То же, что и режим Значки, однако все значки (и их имена) выстроены в столбцы.
3. *Таблица.* Konqueror показывает имя каждого файла (и небольшой значок) в отдельной строке. Также отображаются дополнительные сведения о файле (объём сведений можно изменить с помощью пункта меню **Вид-Подробности**).
4. *Только текст.* Этот режим похож на режим Таблица, но отличается тем, что Konqueror не отображает значков, а все имена каталогов начинаются с косой черты (`«/»`).
5. *В виде дерева.* Этот режим удобен, когда вам необходимо просмотреть структуру каталогов, начиная с текущего. Файлы отображаются так же, как и в режиме Таблица, но слева от каждого имени каталога находится небольшой плюс `+` в квадратике. Щёлкнув по нему, вы развернёте соответствующий каталог и сможете увидеть его содержимое.
6. *Таблица с подробностями.* Этот режим похож на режим Таблица, однако отличается тем, что Konqueror по возможности показывает количество строк, слов и символов в каждом файле, а также формат файла.

Вместо стандартных значков (в режиме **Значки** и **Список**) Konqueror может показывать содержимое файла в уменьшенном виде.

Если Вы включите режим **Показывать подсказки к файлам** в разделе Поведение окна настройки Konqueror, то каждый раз, когда Вы наводите мышь на имя файла (или на значок), Konqueror будет показывать дополнительные сведения о файле в виде подсказки. Объём отображаемых сведений можно настроить с помощью поля **Количество строк** в подсказках в окне настройки Konqueror. Если включён режим **Показывать миниатюры в подсказках**, в окошке с подсказкой также будет отображено содержимое файла в уменьшенном виде.

Выбрав пункт меню **Вид-Миниатюры**, Вы вызовете подменю, которое позволит вам разрешить показ миниатюр для файлов определённого типа. Это означает, что вместо значка будет показано содержимое файла в уменьшенном виде. Если просмотр миниатюр включён для звуковых файлов, то когда вы наведёте мышь на значок такого файла, файл будет воспроизведён. Заметьте, что просмотр миниатюр доступен только в режимах **Значки** и **Список**.

### **Перемещение по каталогам**

Чтобы найти нужный файл, необходимо открыть каталог, в котором он находится. Переходя от одного каталога к другому, Вы перемещаетесь по древовидной структуре каталогов. Чтобы перейти в подкаталог, щёлкните по нему левой кнопкой мыши, а если он уже выделен, просто нажмите клавишу **Ввод**. Чтобы перейти в родительский каталог, нажмите кнопку **Вверх** на панели инструментов. Также можно использовать комбинацию клавиш **Alt+Стрелка вверх** или пункт меню **Переход-Вверх**.

Чтобы выделить файл или папку, не открывая её, воспользуйтесь клавишами со стрелками. Сведения о выделенном объекте будут показаны в строке состояния.

Чтобы облегчить обзор каталогов, можно воспользоваться режимом **В виде дерева**; в этом режиме слева от каждого каталога Вы увидите небольшой квадратик. Если в нём показан знак +, щелчок по нему позволит развернуть данный каталог, то есть отобразить все его файлы и подкаталоги в виде ветвей. Знак в квадратике сменится на -; щелчок по нему позволяет свернуть каталог. Когда Вы найдёте необходимый вам каталог, щёлкните по его значку (или имени), и он будет открыт. Боковая панель навигации также может помочь в нахождении необходимого каталога.

Вы можете перейти к известному вам каталогу, введя путь в строке адреса, выбрав пункт меню **Адрес-Открыть** страницу или нажав **Ctrl+ O**. При этом Вам может пригодиться возможность автодополнения текста. Следует также помнить, что в Linux в названиях файлов строчные и прописные буквы различаются.

После того, как Вы перешли в какой-либо каталог, Вы можете выйти из него, нажав кнопку Назад на панели инструментов, выбрав пункт меню **Переход-Назад** или нажав **Alt+Стрелка влево**.

Если Вы вернулись назад, Вы также можете перейти вперёд. Для этого нажмите кнопку Вперёд на панели инструментов, выберите пункт меню **Переход-Вперёд** или нажмите **Alt+Стрелка вправо**.

*Подсказка.* Если вы нажмёте левую кнопку мыши на значке **Вверх, Назад** или **Вперёд** и будете удерживать кнопку нажатой, появится меню, из которого вы сможете выбрать последние посещённые вами адреса.

### **Удаление файлов и каталогов**

Konqueror предоставляет Вам два способа удалить файл или каталог:

1. Вы можете перенести его в корзину, что является самым безопасным способом, так как если файл был удалён по ошибке, его можно восстановить.
2. Вы можете просто удалить его; при этом файл будет удалён из каталога, а пространство, которое он занимал на диске, станет доступно для размещения других файлов. Это также можно сделать с помощью команды **rm**

Самый простой способ удалить файл или каталог — щёлкнуть по нему правой кнопкой мыши и в появившемся меню выбрать пункт: **Выбросить в корзину**. Если нажимать правую кнопку мыши, одновременно удерживая кнопку **Shift**, то можно в меню пункт **Выбросить в корзину** заменится пунктом **Удалить**, что приведет в полному удалению файла, без возможности восстановления.

Если нужный файл или каталог уже выделен, вы также можете воспользоваться одним из пунктов меню Правка: **Выбросить в корзину** или **Удалить**.

Нажатием клавиши **Del** вы переместите выделенные файлы в корзину. Нажатием **Shift+Del** вы удалите их.

*Замечание.* Для того, чтобы удалить файл или каталог, у Вас должны быть соответствующие права. Подробнее об этом можно прочитать в разделе Работа в режиме администратора.

### **Восстановление файлов**

Если вы по ошибке удалили файл в корзину, его можно оттуда восстановить. Для работы с корзиной щелкните по иконке с изображением корзины на нижней панели.

В свойствах каждого файла или каталога вы можете посмотреть его исходный путь ( в закладке Свойства → Метаданные), именно туда будет по умолчанию происходить восстановление.

Находясь в корзине, вы можете посмотреть содержимое всех фалов и принять решение какие из них вы хотите восстановить.

Щелкнув правой кнопкой мыши по файлу или каталогу, вы имеете возможность:

- Восстановить – восстановить файл в то место, из которого он был удален
- Вырезать – скопировать файл в буфер обмена и после копирования в другое место удалить из корзины
- Копировать - скопировать файл в буфер обмена и после копирования в другое место оставить в корзине
- Переименовать – переименовать файл в корзине
- Удалить – удалить файл из корзины. **Важно:** при этом восстановление этого файла в дальнейшем будет невозможно.

### Перемещение и копирование

Переместить файл или каталог можно несколькими способами:

1. В контекстном меню выберите пункт **Копировать**. Если нужный файл или каталог уже выделен, Вы также можете воспользоваться кнопкой Копировать на панели инструментов, пунктом меню **Правка-Копировать** или сочетанием клавиш **Ctrl+C**.
2. После этого перейдите в каталог, в который Вы хотели бы поместить нужный вам файл, и нажмите кнопку Вставить на панели инструментов, выберите пункт меню **Правка-Вставить** или нажмите **Ctrl+V**. Вы также можете поместить указатель мыши в свободную область окна, щёлкнуть правой кнопкой мыши и в появившемся меню выбрать пункт **Вставить**.

Перемещение файла осуществляется так же, как и копирование, за исключением того, что вместо пункта или кнопки **Копировать** следует воспользоваться пунктом (или кнопкой) **Вырезать** или комбинацией клавиш **Ctrl+X**. При вставке вырезанный файл будет перемещён в данный каталог.

Вы также можете скопировать или переместить выделенные файлы, выбрав пункт меню **Правка-Копировать файлы (F7)**, **Правка-Переместить файлы (F8)** или щёлкнув по файлу правой кнопкой мыши и в появившемся меню выбрав пункт **Копировать в**, **Переместить в**.

*Замечание.* Для того, чтобы скопировать или переместить файл (или каталог), у вас должны быть соответствующие права. Подробнее об этом можно прочитать в разделе Работа в режиме администратора.

*Перетаскивание мышью.* Концептог также поддерживает копирование и перемещение файлов с помощью мыши.



Для этого откройте два окна (или вкладки) Konqueror: одно — с каталогом, где находится нужный вам файл, а другое — с каталогом, куда этот файл необходимо скопировать. Щёлкните по файлу левой кнопкой мыши и удерживайте её нажатой; перенесите мышь во второе окно, на место, не занятое другими значками, и отпустите кнопку мыши. После этого в появившемся меню выберите пункт **Копировать** или **Переместить**. Обратите внимание: если вы перенесёте значок не на свободное место, а на значок каталога, файл появится не в текущем каталоге, а в том, на значок которого Вы его перенесли.

Также можно настроить Konqueror таким образом, что в одном окне будет отображаться сразу несколько каталогов, что сделает перетаскивание мышью гораздо удобнее.

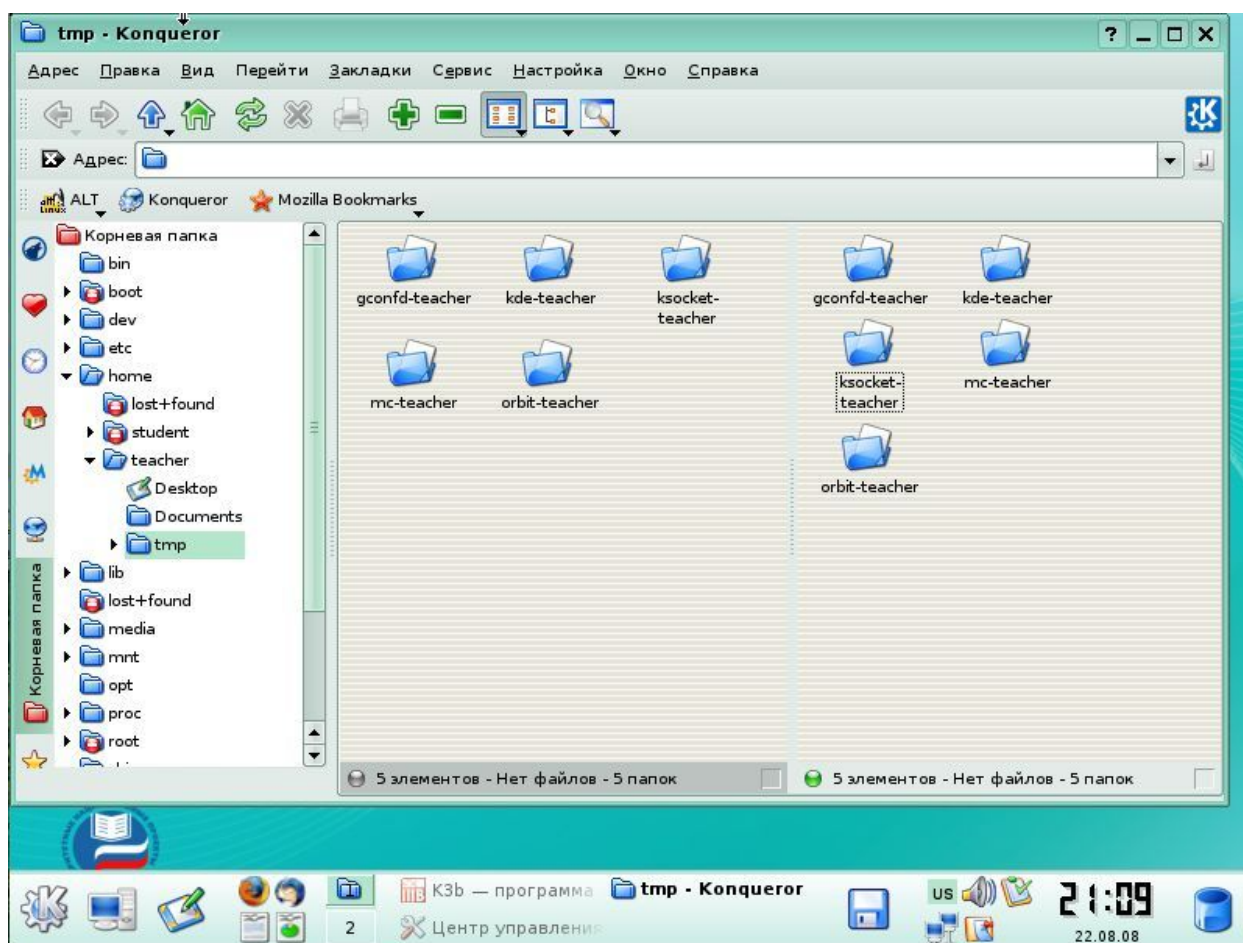


Рис. 12. Два окна (вкладки) Konqueror

На этом рисунке показано, как можно разделить окно с помощью пункта **меню Окно-Разделить** панель по вертикали или комбинации клавиш **Ctrl+Shift+L**; при этом в каждом окне будет показан свой каталог.

Чтобы Konqueror мог отображать в обеих панелях разные каталоги, необходимо убедиться, что панели не связаны между собой: квадратик в нижнем правом углу каждой из них должен быть пуст.

Чтобы узнать, какая панель в данный момент активна, посмотрите на строки состояния панелей: у активной слева на строке состояния будет гореть зелёный индикатор. Чтобы перейти к другой панели (сделать её активной), щёлкните по любому свободному месту в самой панели или в её строке состояния.

Чтобы убрать текущую панель, воспользуйтесь комбинацией клавиш **Ctrl+Shift+R** или пунктом меню **Окно-Удалить** активную панель. Вы также можете щёлкнуть правой кнопкой мыши по строке состояния этой панели и в появившемся меню выбрать пункт **Удалить активную панель**.

Работая с вкладками в Konqueror, Вы можете перетащить файл на заголовок вкладки, не отпуская кнопку, тогда она сама откроется, и вы сможете перенести файл туда, куда хотите.

Вы можете открыть одновременно сколько угодно панелей, разделив экран как по горизонтали, так и по вертикали. Для этого в активной панели щёлкните по строке состояния и выберите пункты меню **Разделить панель по горизонтали (Ctrl+Shift+T)** или **Разделить панель вертикали (Ctrl+Shift+L)**.

### **Файлы с одинаковыми именами**

Если вы попытаетесь вставить файл в каталог, в котором файл с таким именем уже имеется, Konqueror предложит вам следующие действия:

1. **Заменить:** новый файл будет записан поверх старого. Если Вы копируете несколько файлов, вы также можете выбрать **Заменить все**.
2. **Пропустить:** файл скопирован не будет. При копировании нескольких файлов вы также можете выбрать **Пропустить все**.
3. **Дать копируемому файлу новое имя:** Вы можете ввести имя сами или позволить это сделать Konqueror, нажав на кнопку **Предложить**. После этого нажмите кнопку **Переименовать**.

### **Выделение нескольких файлов**

Иногда бывает необходимо скопировать, переместить или удалить несколько файлов, которые имеют между собой что-то общее. Например, вам может понадобиться перенести все графические файлы с расширением .PNG из одного каталога в другой. Konqueror облегчает эту задачу, позволяя автоматически выделить группу файлов, имеющих похожие имена.

Выберите пункт меню **Правка-Выделить...** или нажмите **Ctrl++**. В появившемся окне введите необходимый шаблон файла, используя спецсимволы.

Спецсимвол **\*** - означает любое количество символов,.

Спецсимвол **?** - означает один любой символ, символ не может быть пропущен.

Спецсимвол [ ] – задает перечисление возможных символов в этой позиции строки.

Нажмите кнопку ОК, и Konqueror автоматически выделит все файлы, имена которых совпадают с данным шаблоном. Например:

5. При шаблоне `flag*.png` будут выделены все имена файлов, начинающиеся со слова «flag» и заканчивающиеся на «.png».
6. При шаблоне `memo?.txt` будут выделены, к примеру, файлы `memo1.txt` и `memo9.txt`, но не `memo99.txt` и не `memo.txt`.

Не забывайте о том, что в Linux различаются регистры. Решить эту проблему можно используя спецсимвол [ ]. Например шаблон `[dDt]*` выделит все файлы и директории, имена которых начинаются с букв D,d,t.

Можно уменьшить количество выделенных файлов, указав, с каких файлов следует снять выделение. Для этого выберите пункт меню **Правка-Отменить выделение...** или нажмите **Ctrl+-** (прим. Ctrl + дефис).

Чтобы снять выделение со всех файлов, нажмите **Ctrl+U**, выберите пункт меню **Правка-Отменить** всё выделение или просто щёлкните по любому месту окна, не занятому значками.

Также имеется возможность обратить выделение, то есть снять его со всех выделенных файлов и выделить остальные. Для этого выберите пункт меню **Правка-Инвертировать выделение** или нажмите **Ctrl+\***.

Ниже приведены примеры клавиш, которые часто используются для выделения файлов в режимах **Таблица**, **В виде дерева** и **Только текст**:

1. **Пробел**. Выделить или снять выделение.
2. **Ins**. Обратить выделение и перейти к следующему файлу.
3. **Ctrl+Стрелка вверх**, **Ctrl+Стрелка вниз**, **Ctrl+Home**, **Ctrl+End**, **Ctrl+Page Up**, **Ctrl+Page Down**. Перейти в соответствующем направлении, по пути обращая выделение всех файлов.
4. **Shift+Стрелка вверх**, **Shift+Стрелка вниз**, **Shift+Home**, **Shift+End**, **Shift+Page Up**, **Shift+Page Down**. Снять всё выделение, затем перейти в соответствующем направлении, по пути выделяя все файлы.

Когда необходимые вам файлы выбраны, вы можете использовать такие команды, как копирование, перемещение или удаление: их действие будет распространяться на все файлы сразу.

*Замечание.* В зависимости от установленной клавиатуры и текущей локали для сочетаний **Ctrl++**, **Ctrl+-** и **Ctrl+\***, возможно, необходимо будет использовать клавиши **+**, **-** и **\*** на цифровой части клавиатуры.

## Создание новых файлов и каталогов

Если Konqueror находится в режиме менеджера файлов, выберите пункт **Создать новый...** из меню **Правка** или щёлкните правой кнопкой мыши по любому месту окна, которое не занято файлами. Появится меню, с помощью которого Вы сможете создать в текущем каталоге такие объекты:

1. **Ссылка на приложение...** С помощью этого пункта можно создать значок, щёлкнув по которому, вы запустите определённое приложение. При выборе этого пункта появится окно с тремя вкладками: **Общие** — здесь вы можете выбрать значок приложения и указать его название; **Права доступа** — здесь можно указать, кто имеет право использовать или изменять создаваемую ссылку; **Запуск** — здесь вы можете указать путь к запускаемой программе, например `kedit`, чтобы запустить редактор KEdit, или `kedit /home/pam/todo.txt`, чтобы запустить KEdit и открыть в нём файл `/home/pam/todo.txt`; вкладку **Приложение** обычно можно не заполнять. Если Вы хотите поместить значок на рабочий стол, создавайте ссылку в каталоге `~/Desktop` (в зависимости от вашей системы этот файл может называться и по-другому). Также можно щёлкнуть по рабочему столу правой кнопкой мыши, и воспользоваться таким же меню **Создать**. Если требуется создать множество ссылок, и Вы не хотите, чтобы они загромождали рабочий стол, Вы можете поместить их в специальный каталог для приложений. Чтобы перейти в этот каталог, выберите пункт меню **Переход-Приложения**. После этого останется лишь поместить ссылку на этот каталог (обычно это `~/kde/share/applink`) на рабочий стол. Если Вы хотите, чтобы ссылка на приложение находилась на панели быстрого запуска, создайте ссылку в каталоге приложений и перетащите её мышью на любое свободное место на панели.
2. **Адрес Интернет...** С помощью этого пункта можно создать значок, щёлкнув по которому, Вы перейдёте к определённому адресу Интернета или к каталогу. Как и в случае с пунктом **Ссылка на приложение...**, Вы можете поместить значок на рабочий стол, если создадите ссылку в каталоге `~/Desktop` или воспользуетесь меню **Создать...** прямо на рабочем столе (щёлкнув по нему правой кнопкой мыши). После создания значка в качестве его имени будет отображаться адрес, на который он ссылается; чтобы изменить имя, щёлкните по значку правой кнопкой мыши и в появившемся меню выберите пункт **Свойства...**. В окне свойств выберите вкладку **Общие** и введите новое название значка.
3. **Ссылка на устройство...** Этот пункт позволяет создать значок, щёлкнув по которому, вы сможете просмотреть содержимое различных устройств: CD-ROM, DVD-ROM, дисковод, жесткий диск, а также другие устройства, в зависимости от комплектации вашего компьютера.

4. **Папку...** Этот пункт позволяет создать (под)каталог.
5. **Текстовый файл...** Этот пункт позволяет создать обычный текстовый файл с заданным именем.
6. **Файл HTML...** Этот пункт позволяет создать файл HTML. Когда вы будете давать ему имя, желательно в качестве расширения указать `.html`, чтобы избежать возможных недоразумений.

### **Переименование и изменение прав доступа к файлу**

Самый простой способ переименовать файл или каталог — это щёлкнуть по его значку правой кнопкой мыши и в появившемся меню выбрать пункт **Переименовать**.

Чтобы изменить права доступа к файлу или каталогу, щёлкните по его значку правой кнопкой мыши и в появившемся меню выберите пункт **Свойства...** или, если файл уже выделен, воспользуйтесь пунктом меню **Правка-Свойства**.

Появится окно свойств файла, содержащее несколько вкладок:

1. Вкладка **Общие** предоставляет общие сведения об объекте и позволяет изменить его имя и, в случае с каталогом, его значок.
2. Вкладка **Права** позволяет изменить права доступа к объекту.

### **Копирование и переименование**

Если Вы хотите создать копию определённого файла и поместить её в тот же каталог, что и оригинал, «скопируйте» нужный файл (с помощью команды **Копировать**), а затем выполните вставку. При этом появится сообщение о том, что файл с таким именем уже существует. Введите новое имя и нажмите кнопку **Переименовать** (или просто нажмите кнопку **Предложить**, и Konqueror переименует файл автоматически).

### **Работа в режиме администратора**

Если Вы зашли в систему как обычный пользователь и пытаетесь получить доступ к файлам вне вашего домашнего каталога, Вы нередко будете получать такое сообщение об ошибке: **Отказано в доступе**. Чтобы получить доступ ко всем файлам, Вам необходимо зайти в систему под именем администратора, или `root`. Вместо того, чтобы выходить из системы, а затем вновь заходить, можно запустить Konqueror сразу в режиме администратора. Для этого в главном меню **К** выберите пункт **Система-Менеджер файлов (в режиме администратора)**.

*Внимание.* Будьте осторожны. В режиме администратора (`root`) вы получаете полный контроль над системой, и одна неправильная команда может нанести непоправимый ущерб. Кроме того, в этом режиме крайне не рекомендуется подключаться к Интернету, так как риск подвергнуться атаке существенно повышается.

## Работа с командной строкой

Konqueror является чрезвычайно мощным и гибким менеджером файлов, однако опытные пользователи часто предпочитают иметь под рукой командную строку и работать с системой на уровне командной строки. Для этого можно, например, запустить Konsole, воспользовавшись пунктом меню **Сервис-Открыть терминал** или нажав **Ctrl+T**.

Если необходимо лишь запустить программу или открыть документ, легче воспользоваться пунктом меню **Сервис-Выполнить команду... (Alt+F2)**.

Если выбрать пункт меню **Сервис-Выполнить команду оболочки... (Ctrl+E)**, появится окно, в котором вы сможете ввести нужную вам команду, например **ps -ax | grep kdeinit**. Следует отметить, что в этом случае Вам будут недоступны некоторые возможности (например, не будут корректно работать программы `top` и `less`), однако преимущество данного способа в том, что вам не придётся тратить время на запуск Konsole.

Для более сложных действий в Konqueror имеется ещё одна возможность: выбрав пункт меню **Окно-Показать эмулятор терминала**, Вы откроете терминал прямо в Konqueror, в виде ещё одной панели. Кроме того, если в правом нижнем углу одной из обычных панелей установлен специальный флажок, то в какой бы каталог вы ни перешли, в терминале он также станет текущим.

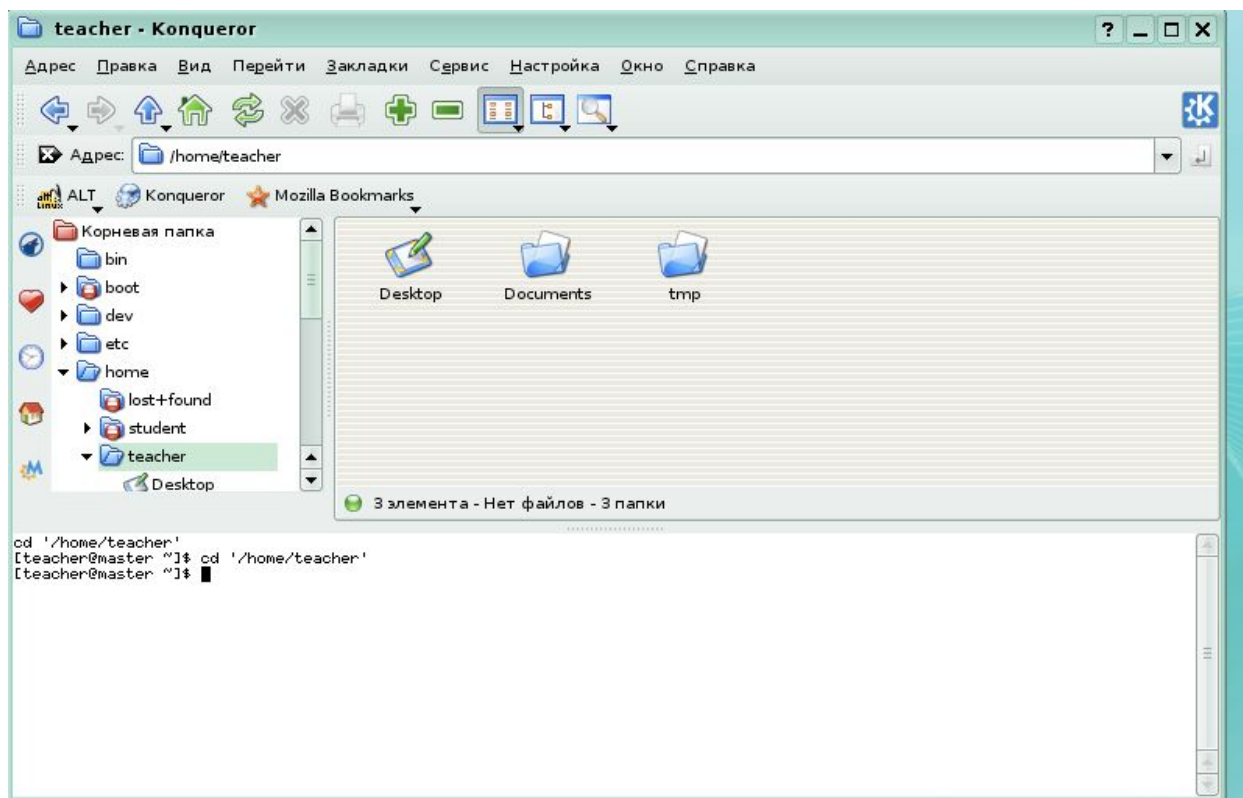


Рис. 13. Konqueror. Режим эмулятора терминала

## Поиск файлов и каталогов

Если Вы забыли, где находится нужный Вам файл, нажмите кнопку **Поиск файла** на панели инструментов или выберите пункт меню **Сервис-Поиск файла...** В окне Konqueror появится окошко программы KFind. *Замечание:* если название файла или каталога начинается с точки, он называется «скрытым», и Konqueror его обычно не показывает. Чтобы изменить это, установите флажок напротив пункта меню **Вид-Показывать скрытые файлы.**

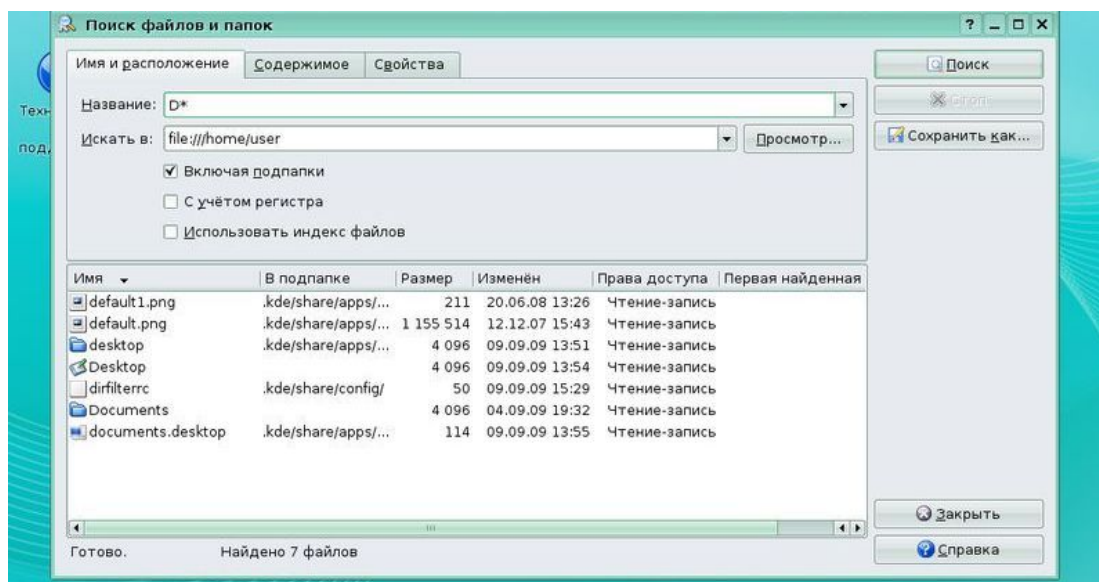


Рис. 14. Kfind. Окно поиска файлов и папок

При запуске KFind появится довольно простое окно, закладка **Имя и расположение**. Введите имя файла, который вы ищете, в поле ввода **Название:**, а каталог для поиска в поле ввода **Искать в:** или укажите его с помощью кнопки **Просмотр...** и нажмите **Enter** или кнопку **Поиск**. Если установлен флажок **Включая подкаталоги**, поиск будет вестись во всех подкаталогах выбранного каталога. Результаты поиска будут отображены в поле внизу окна.

Можно использовать следующие шаблоны:

1. *Звёздочка «\*»*. Звёздочка обозначает любое количество (в том числе и ноль) пропущенных символов. Это означает, например, что в результате поиска `marc*` вы получите `marc`, `marc.png` и `marc_must_not_read_this.kwd`. Поиск `marc*.kwd`, в свою очередь, приведёт к результатам `marketplace.kwd` и `marc_must_not_read_this.kwd`.
2. *Вопросительный знак «?»*. В отличие от звёздочки, вопросительный знак означает, что пропущен ровно один символ, так что в результате поиска `mar?` вы получите `marc`, а по `marc?` не найдете ничего, так как наши файлы называются `marc` и `marc.png`. В условие поиска можно поместить сколько угодно вопросительных знаков, и вы найдете ровно столько символов.

Разумеется, оба шаблона в условии можно комбинировать.

Закладка **Содержимое**.

Уточнить тип нужного файла можно с помощью **Содержимое-Тип файла**.

**Содержит текст**. Введите слово или фразу, содержащиеся в файле, который вы ищете. Помните, что если поиск идёт в большом каталоге или с пометкой **Включая подкаталоги**, он может занять много времени.

*Замечание.* Эта опция не работает для всех файлов, перечисленных в списке **Тип файла**. Поддерживаются только следующие типы:

1. Текстовые файлы, например исходные коды, и файлы README
2. KWord
3. KPresenter
4. KSpread
5. OpenOffice.org Writer
6. OpenOffice.org Impress
7. OpenOffice.org Calc

**Поиск с учётом регистра**. Если выбрать эту опцию, KFind найдёт только те файлы, у которых совпадает регистр. Например, MARC найдёт только «MARC» но не «Marc».

Закладка **Свойства**. Дополнительные условия для поиска:

**Найти все файлы созданные или изменённые**. Здесь вы можете определить две даты, между которыми были изменены или созданы файлы, также можно задать определённый период времени.

**Размер файла**. В этом поле можно указать, больше или меньше какого значения должен быть найденный файл.

**Принадлежащие группе, Принадлежащие пользователю**. Здесь можно уточнить пользователя и группу.

### 2.3.4. Популярные файл-менеджеры

Для многих пользователей наиболее удобный способ работы с файловой системой — «классический» двухпанельный диспетчер файлов, работающий в текстовом режиме (в терминале) — **Midnight Commander** (название утилиты — **mc**). Его функциональность шире стандартных операций с файлами — он позволяет открывать файлы для просмотра и редактирования, вызывать вспомогательные программы для работы с архивами (и даже «заходить» в архивы, как в каталоги), передавать данные по сети и т. п. Midnight Commander имеет также неплохой встроенный текстовый редактор, опять-таки «классического» стиля.



Вызвать **mc** можно из пункта меню **Службные - Midnight Commander**. Или через пункт меню **Выполнить программу**, введя в строке **Команда** **mc** и отметив в **Параметрах** пункт **Выполнить в терминале**.

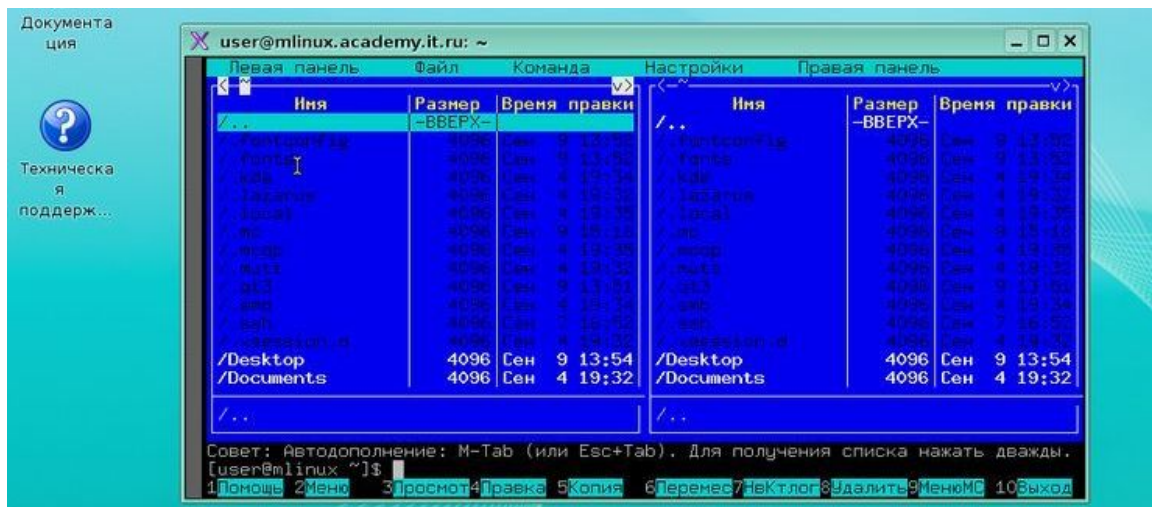


Рис. 15. Midnight Commander

Работа в **mc** осуществляется с помощью стрелок, переключение между панелями кнопкой **Tab**.

*Основные возможности менеджера доступны через функциональные кнопки:*

- **F1** – справка
- **F2** – доступ в настраиваемое меню пользователя для выбора дополнительных команд
- **F3** – просмотр содержимого текстового файла
- **F4** – редактирование текстового файла
- **F5** – копирование
- **F6** – перемещение
- **F7** – создание нового каталога
- **F8** – удаление
- **F9** – доступ в верхнее контекстное меню
- **F10** – выход из программы

Далеко не всё, что нужно делать в Linux, в среде **mc** так же удобно, как и в полноценной командной строке. Кроме того, при работе с графическими файлами сильно не хватает представления этих файлов в виде миниатюр (thumbnails), чтобы выбирать среди них по содержимому, а не только по имени. Такими возможностями обладают многочисленные графические диспетчеры файлов; помимо тех, что включены в среды KDE и GNOME, есть множество независимых: **dfm** (похожий на диспетчер файлов OS/2), **emelfM2**, **EZFM** и **X Northern Captain**, **gentoo** и **worker**, **FSV** и **XCruiser** (трёхмерные).

Среди них встречаются и ориентированные специально на просмотр изображений, такие как GQView, endeavour, gview, qiv, xzgv и некоторые другие, — с возможностями слайд-шоу, автоматического изменения размера, показа картинки на полный экран и т. п. Эти редакторы можно при желании дополнительно установить в систему.

### 2.3.5. Форматирование дискет

**Kfloppy** - утилита, позволяющая легко форматировать гибкие диски размером 3.5»и 5.25» дюймов. Запустить программу можно следующим образом: **К - Настройка - Форматирование дискет**.

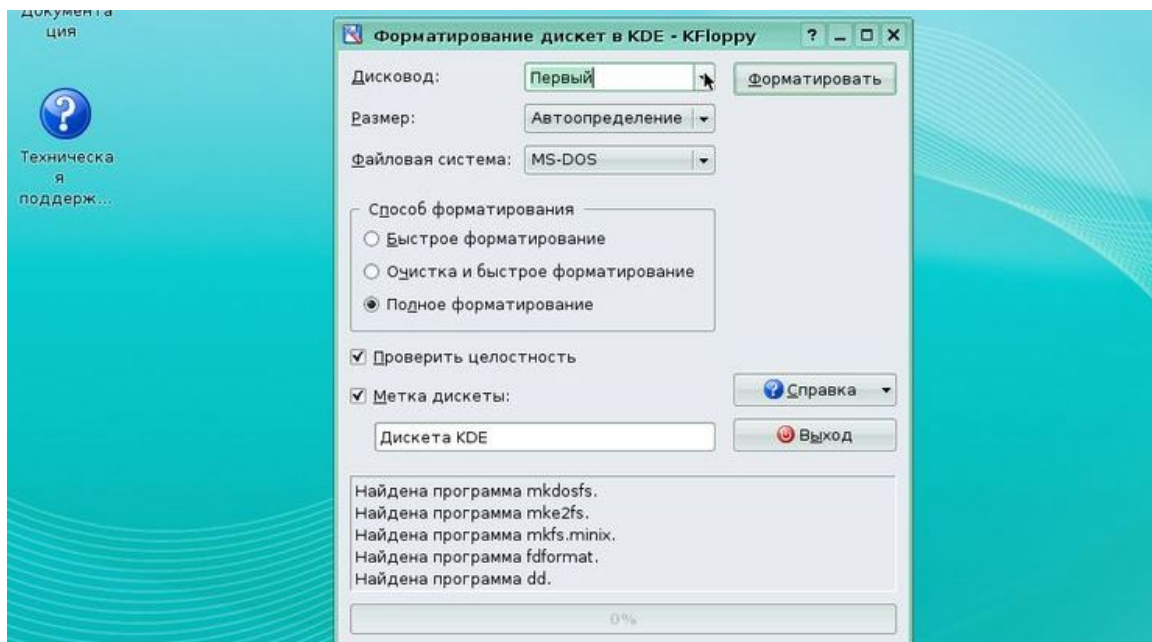


Рис. 16. KFloppy

*Важно.* Убедитесь в том, что ваша дискета не смонтирована. Kfloppy не может форматировать смонтированные дискеты.

При запуске Kfloppy открывается окно, в котором вы можете выбрать наиболее подходящие настройки и контролировать процесс форматирования дискет. Детальное описание этих настроек вы можете прочитать ниже.

*Основные настройки.* В данных настройках вы можете выбрать дисковод, тип дискет и файловую систему для формируемой дискеты.

**Дисковод.** Нажав на стрелку в выпадающем меню с меткой **Дисковод:**, вы можете выбрать номер дисковода, если у вас в системе их несколько.

В выпадающем меню с меткой **Размер** можно выбрать следующие типы дисковода:

1. Автоопределение
2. 3.5» 1.44MB
3. 3.5» 720KB
4. 5.25» 1.2MB

## 5. 5.25» 360KB

*Замечание.* Дискеты формата 800k, используемые дисководами Mac®, не пригодны для использования на PC.

**Файловые системы.** При нажатии на стрелку в выпадающем меню с меткой Файловая система вы можете выбрать следующие типы файловой системы:

**MS-Dos.** При выборе этой файловой системы Kflorry отформатирует вашу дискету под файловую систему, используемую в MSDOS/Windows®. Выберите эту файловую систему, если вы собираетесь использовать дискету в этих операционных системах.

**Ext2.** При выборе этой файловой системы Kflorry отформатирует вашу дискету под файловую систему, используемую в Linux®. Выберите эту файловую систему, если вы собираетесь использовать дискету только в Linux®.

**Minix.** При выборе этой файловой системы Kflorry отформатирует вашу дискету под файловую систему, используемую в Linux.

В центре окна Kflorry вы можете настроить некоторые параметры форматирования. При этом два верхних параметра являются взаимоисключающими.

*Замечание.* Под термином «Взаимоисключающие параметры» понимается ряд параметров, из которых можно выбрать только один.

Два нижних параметра предназначены для маркировки дискет. Детальное описание параметров представлено ниже:

*Способ форматирования:*

Kflorry может форматировать дискету тремя способами:

**Быстрое форматирование.** Форматирование этим способом просто пересоздаст файловую систему на дискете. Все данные на дискете будут удалены. Однако если вы выберете плотность, отличную от плотности вашей дискеты, то Kflorry перераспределит дорожки и сектора перед созданием файловой системы.

**Полное форматирование.** Форматирование этим способом перераспределяет дорожки и сектора, создаёт выбранную вами файловую систему и проверяет дискету на плохие сектора.

**Проверить целостность.** При включении данного параметра Kflorry проверит правильность форматирования и пометит найденные плохие сектора.

**Метка дискеты.** В двух нижних параметрах можно назначить метку для формируемой дискеты. Метку для дискеты можно ввести в поле ввода, расположенное ниже флажка **Метка:**. После окончания форматирования метка будет записана на дискету.

В правой части окна Kflorry расположены следующие три кнопки:

**Форматирование.** Нажатие этой кнопки запускает процесс форматирования. Если вы уверены в правильности настроек, можете начать форматирование дискеты, нажав на эту кнопку. При этом вместо кнопки Форматирование на кнопке появится кнопка Отмена. Процесс форматирования будет отображаться в двух прямоугольных полях, расположенных в самом низу окна программы.

Верхнее поле является индикатором состояния, в нем будет отображаться информация о ходе процесса, а также специфические детали форматирования и процесс проверки дискеты.

В нижнем поле расположен индикатор выполнения, в котором отображается процесс форматирования, а затем и проверки поверхности в виде цветной полосы и числа. До того, как вы запустите процесс форматирования, это поле будет пустым, и по его центру будет видна только цифра 0%. Когда же вы начнёте форматировать дискету, цветная полоса будет постепенно, по мере выполнения, заполнять поле слева направо, а число по центру будет отображать ход выполнения задачи в процентах.

**Важно.** Убедитесь в том, что ваша дискета не смонтирована. Kfloppy не может форматировать смонтированные дискеты.

*Помощь.* Нажатие на эту кнопку запускает центр помощи KDE с выводом справки по Kfloppy.

*Выход.* Нажатие на эту кнопку закрывает Kfloppy.

### **2.3.6. Запись CD-R и CD-RW**

**K3b** – мощная, универсальная программа для записи CD и DVD. Существует довольно распространенное, небезосновательное мнение, что K3b входит в состав KDE. Действительно, это единственный достойный программный продукт для записи оптических дисков, использующий библиотеку QT и модуль kdelibs. Несмотря на это, K3b является независимым программным продуктом, хотя на официальном сайте проекта дается четкий намек на то, что программа оптимизирована для работы в KDE.

Официальный сайт: [www.k3b.org](http://www.k3b.org)

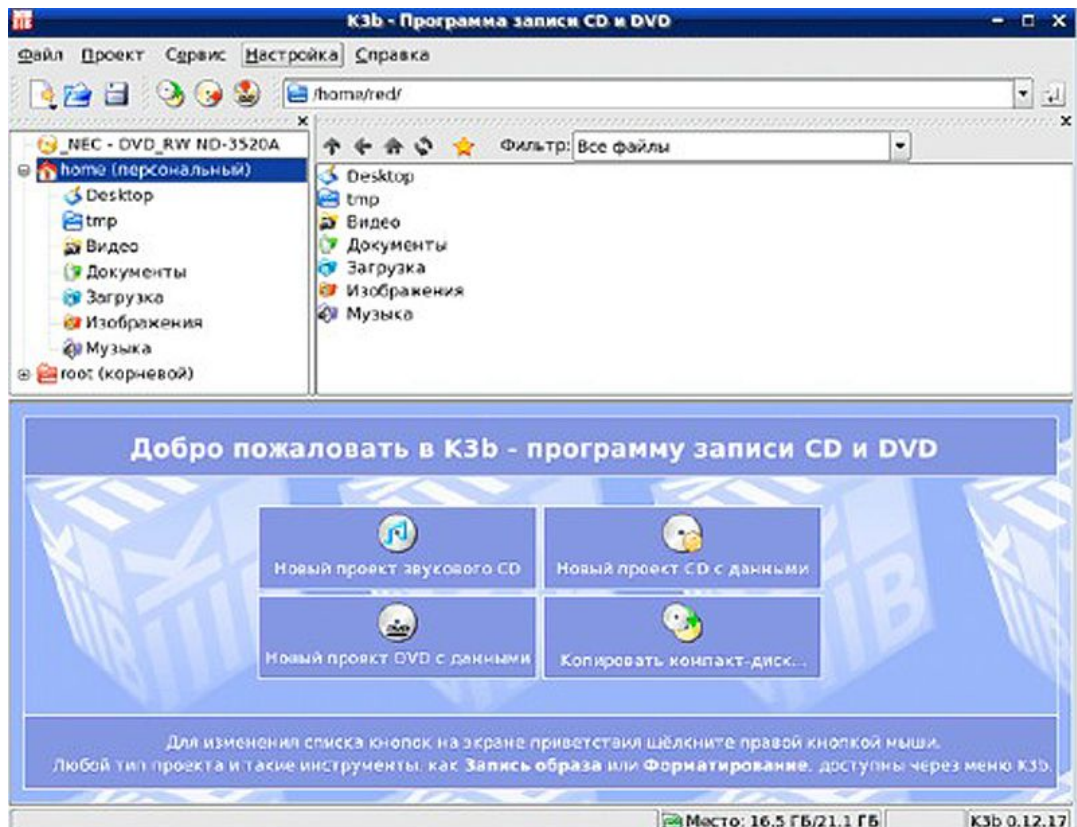


Рис.17. Рабочее окно K3b

После своего старта программа предлагает на выбор четыре варианта действий. Вы можете создать проект CD или DVD диска с данными, начать работу над Audio CD, а также заняться клонированием оптических носителей.

В главном меню программы находится список остальных поддерживаемых программой возможностей. Вы можете, например, создавать проекты DVD Video. При этом внутри его панели автоматически создаются две папки, присущие данному формату.

После выбора типа проекта, например, создание проекта DVD с данными, открывается новое окно, в котором можно добавить в проект нужные файлы. После проверки корректности созданной структуры можно начинать запись.

Многие программы, являющиеся оболочками к пакету cdrecord, обладают одним неприятным свойством. Перед тем, как начать запись оптического диска, программа обязана сначала создать образ данных проекта. На это тратится немало времени. Подобное поведение легко объяснимо. Как и в случае работы с консольными утилитами, вся процедура записи делится на три этапа. Сначала данные собираются в одном временном каталоге, затем создается образ данных и лишь затем включается механизм записи носителя. В результате, время, затрачиваемое за запись отдельного диска, в среднем удваивается. И здесь нечего не поделаешь, ведь такова специфика работы cdrecord в Linux.

K3b позволяет начинать запись оптического диска сразу же после запуска данного процесса, минуя долгое, утомительное ожидание окончания копирования информации и

создания образа. По этой причине K3b можно назвать самым быстрым, оперативным программным продуктом для записи CD и DVD, основанным на утилитах пакета cdrecord.

Программа поддерживает работу над несколькими проектами внутри единого рабочего окна приложения. Переключение между документами происходит с помощью вкладок.

Во время создания музыкального диска, вы можете сразу же просматривать и редактировать ID3-теги музыкальных файлов.

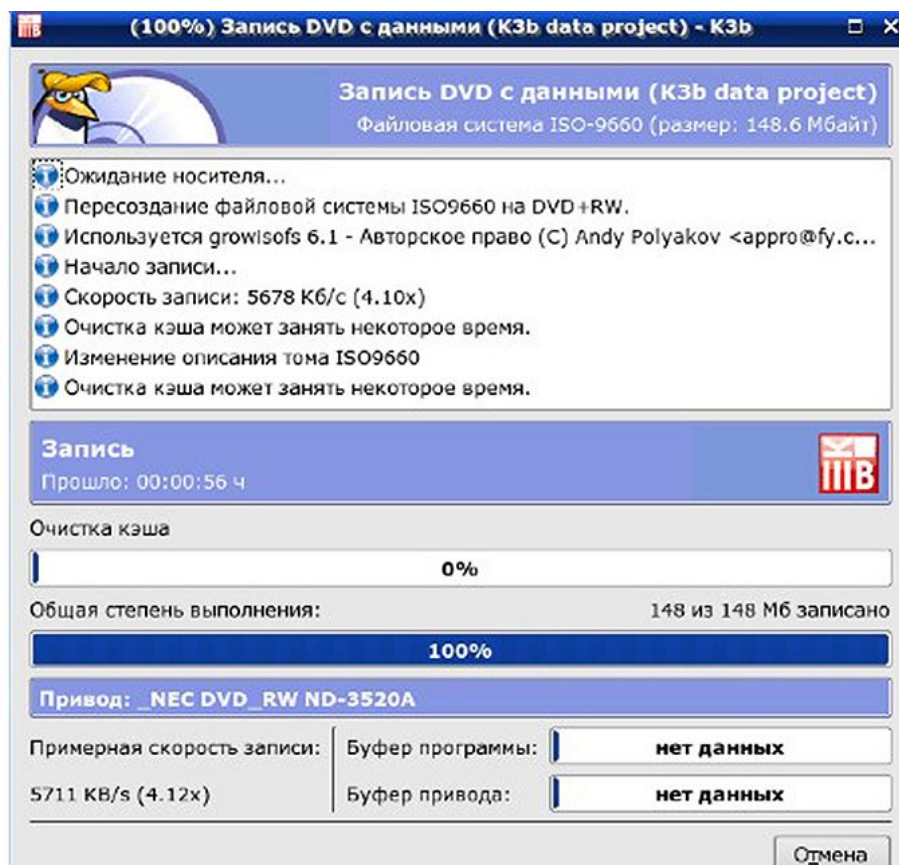


Рис.18. Отображение статуса записи диска в K3b

В состав K3b входит модуль преобразования фильмов, записанных в формате DVD Video, в единый файл с расширением AVI. При этом можно использовать модули кодирования DivX или xVID. Редактор проекта позволяет обрезать исходную картинку (можно менять пропорции кадра), нормализовывать звуковую дорожку, автоматически разбивать материал на фрагменты (например, по 700МБ). Также можно вручную задавать скорость потока данных отдельно для видео и аудио дорожек. Звук может кодироваться с переменным битрейтом. Помимо извлечения и преобразования материала дисков DVD Video, можно работать с носителями Audio CD. Музыкальные дорожки преобразовываются в разнообразные звуковые форматы, в том числе использующие сжатие данных.

Программа обладает очень удобным интерфейсом, использующим две файловые панели для локального браузера и текущего проекта. Навигация по файловой системе

выполнена в классических традициях приложений KDE. Программа имеет качественную, грамотную русификацию всех пунктов меню, диалоговых окон. В именах папок и файлов внутри проектов можно использовать символы кириллицы. K3b поддерживает смену тем оформления. Допускается создание и прожиг образов дисков в форматах ISO и CUE.

Программа тесно интегрируется в рабочую среду KDE. Например, вы можете выделить в Konqueror несколько файлов или папок, после чего отправить их через контекстное меню на запись в проект K3b. Все визуальные, стилевые настройки рабочей среды применяются и к программе прожига дисков. Вы можете гибко настраивать горячие клавиши на любое действие K3b, менять положение и конфигурацию всех панелей инструментов. По большому счету, в этом нет ничего уникального – любая программы, входящая в состав KDE, обладает идентичными свойствами.

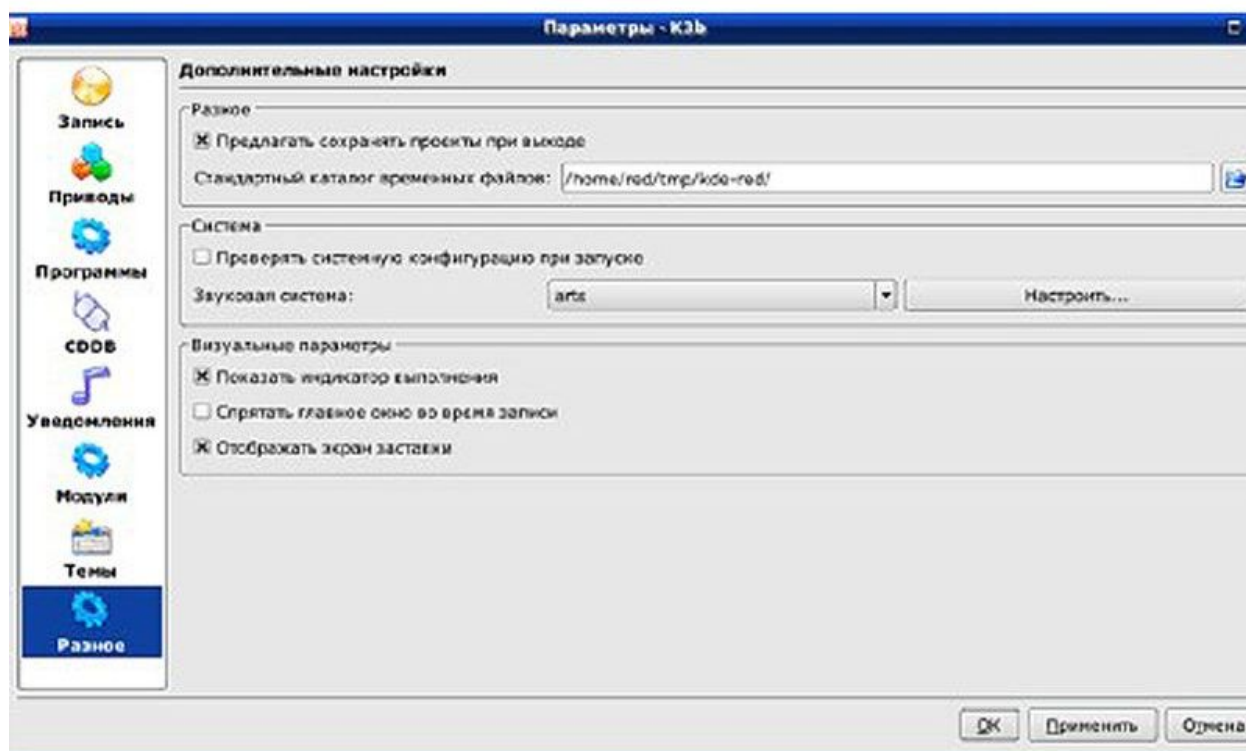


Рис.19. Настройки K3b

Обратной стороной тесной интеграции с рабочей средой являются системные требования программы. Разумеется, если вы работаете в KDE, то K3b будет очень быстро запускаться, и никаких проблем вы не ощутите. Но KDE – не единственная рабочая среда в Linux. Кто-то любит GNOME, а кто-то предпочитает компактные оболочки, такие как WindowMaker и другие. В этом случае при вызове K3b будет дополнительно загружаться огромная библиотека kdelibs, что, помимо резкого возрастания расхода памяти, значительно увеличивает время запуска программы.

Но, несмотря на это, K3b обладает огромной популярностью среди пользователей Linux. Программа нравится не только тем, кто использует KDE, но и любителям других рабочих сред. K3b – как уже было сказано ранее, позволяет прожигать диски, минуя утомительные процедуры создания образов данных. K3b поддерживает все популярные стандарты дисков. Простые проекты с данными, музыкальные диски, фильмы в формате DVD Video – все это без труда можно организовать в K3b. Программа имеет очень простой интерфейс. Если пользователь недавно вырвался из тесных объятий Windows, то, можно предположить, что он привык к интерфейсу Nero или к чему-либо подобному. Создание проекта, две файловых панели и перенос данных между ними – все предельно просто. При этом NeroLinux многие недолюбливают за его лицензию. Программа не бесплатна. K3b же, как и подавляющее количество программного обеспечения для Linux, распространяется под лицензией GPL, что позволяет беспрепятственно загружать и использовать программу. В некоторых дистрибутивах вы можете даже самостоятельно собрать ее из исходных текстов, получив небольшой прирост в производительности, а также некоторые нестандартные свойства, присущие вашей уникальной рабочей станции.

K3b – мощный, универсальный инструмент для создания проектов большинства популярных типов данных. Программа входит в состав всех популярных дистрибутивов Linux, где есть в наличии рабочая среда KDE.

## **2.4. Стандартные программы (обзор)**

Основная особенность программного обеспечения Linux — многообразие продуктов, решающих сходные задачи, особенно если дело касается области, в которой существует несколько подходов к их решению. Открытая модель разработки программ, позволяет любому выбрать самый подходящий для него инструмент и развивать именно его. Поэтому список проектов, так или иначе связанных с Linux, насчитывает десятки (или даже сотни) тысяч наименований.

Все усилия по изучению операционной системы Linux и основных утилит нужны для того, чтобы впоследствии наилучшим образом решать в этой операционной системе любые из своих прикладных задач, разрешимых при помощи компьютера. Для очень многих задач достаточно стандартных инструментов Linux и текстового редактора, однако есть случаи, в которых всё-таки необходима специальная прикладная программа, именно для этого предназначенная, или в которых специальная программа удобнее комбинации стандартных утилит.

Этот раздел посвящён краткому обзору прикладных программ для ПСПО, специально предназначенных для решения самых разных пользовательских задач. Вошедший сюда материал нужно воспринимать только как пример, демонстрацию того,



что и как можно делать в ПСПО, но вовсе не исчерпывающий список. В отличие от основных принципов устройства системы или стандартных утилит, которые не изменяются в течение десятилетий, прикладное программное обеспечение — это область, где всё меняется очень быстро. Технологии, сегодня считающиеся самыми передовыми, уже через несколько месяцев могут устареть.

Прикладные программы для Linux не являются частью самой Linux, поэтому любой из названных ниже программ может не оказаться в каком-то из конкретных дистрибутивов Linux. Но почти наверняка в любом дистрибутиве найдётся не меньше одной или нескольких программ для решения каждой из перечисленных ниже прикладных задач. Остановимся лишь на самых распространённых программных продуктах, входящих во многие дистрибутивы Linux.

### **Текстовые редакторы**

При работе в любой системе вам не обойтись без редактирования текстовых файлов. В Unix-системах для этой цели существуют два вида текстовых редакторов — графические редакторы и текстовые (позволяют работать в режиме командной строки).

В этом разделе мы рассмотрим графические редакторы, однако в системе существуют текстовые редакторы, например *vi*, которые позволят вам работать с текстом удаленно, в режиме командной строки.

### **Текстовый редактор Kate**

В состав рабочей среды KDE, входит несколько текстовых редакторов. Они отличаются по мощности, функциональности и сфере применения. *Kate* можно назвать самым мощным приложением KDE для работы с простым текстом. Следующие, более продвинутые программы, уже рассчитаны на работу со сложным форматированием RTF.

*Kate* позволяет работать с несколькими документами одновременно. Особенностью данного текстового редактора является возможность создания сессий. Каждая сессия состоит из произвольного набора документов. Также все они могут иметь собственные настройки приложения. Возле левой границы рабочего окна приложения находится панель с вкладками. Одна из них вызывает список документов текущей сессии.

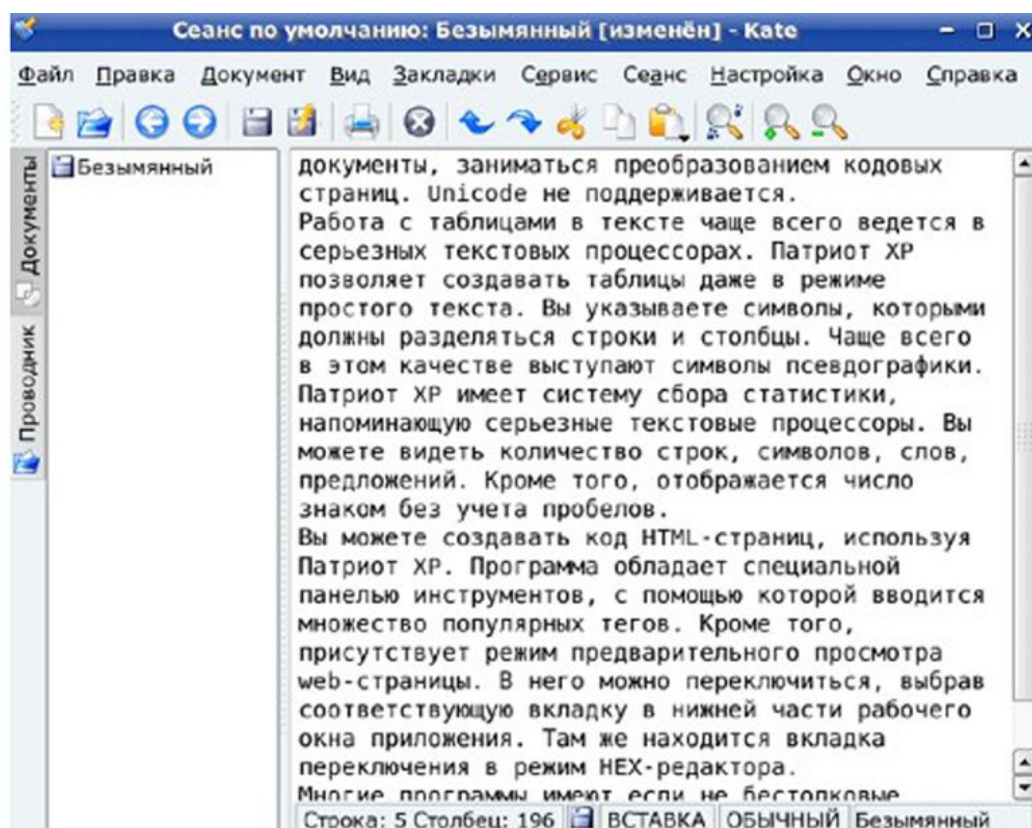


Рис.20. Рабочее окно Kate

При запуске Kate предлагается воспользоваться уже имеющейся сессией, т е получить доступ к документам, с которыми вы работали в прошлый раз, или создать новую сессию, по умолчанию состоящую из одного нового пустого документа.

Большинство текстовых редакторов, имеющих возможность одновременной работы с несколькими файлами, реализуют переключение между документами с помощью панели вкладок. Kate не имеет подобной панели, да она ему и не нужна. Переключение между текстами осуществляется из той же самой панели со списком документов текущей сессии. Программа поддерживает также разделение окна на несколько областей редактирования, внутри каждой из которых открыт собственный документ. Окно можно разделить по горизонтали или по вертикали, это может быть удобно, например, для перевода текста.

Благодаря тесной интеграции с рабочей средой KDE, Kate способен проверять орфографию на русском языке. Проверка возможна во всем документе, от текущей позиции курсора, внутри выделенного фрагмента текста.

Текстовый редактор поддерживает большое число кодировок, включая все кодовые страницы, в которых используется кириллица. Работает их автоматическое определение, переключение режима просмотра, а также преобразование кодировок.

Большое количество вариантов подсветки синтаксиса позволяет использовать Kate для разработки исходных текстов различных приложений, для применений в области Web. Опции раскрашивания текстов сохраняются вместе с сессиями.

Kate позволяет работать со всеми вариантами завершения строк. Имеется возможность переключения режимов. Текстовый редактор позволяет менять отступы строк, а также осуществлять их выравнивание.

Левая боковая панель содержит вкладку файлового навигатора **Проводник**. Он позволяет открывать документы не только с локального диска, но также через сеть, используя протоколы HTTP, FTP, SSH и другие. Для этого адрес документа указывается в окне проводника.

Вы можете устанавливать закладки, а также быстро переходить в папку текущего документа. Закладки имеют место и внутри самих документов. Вы можете добавлять якоря в текст, а потом быстро перемещаться к ним в любое время.

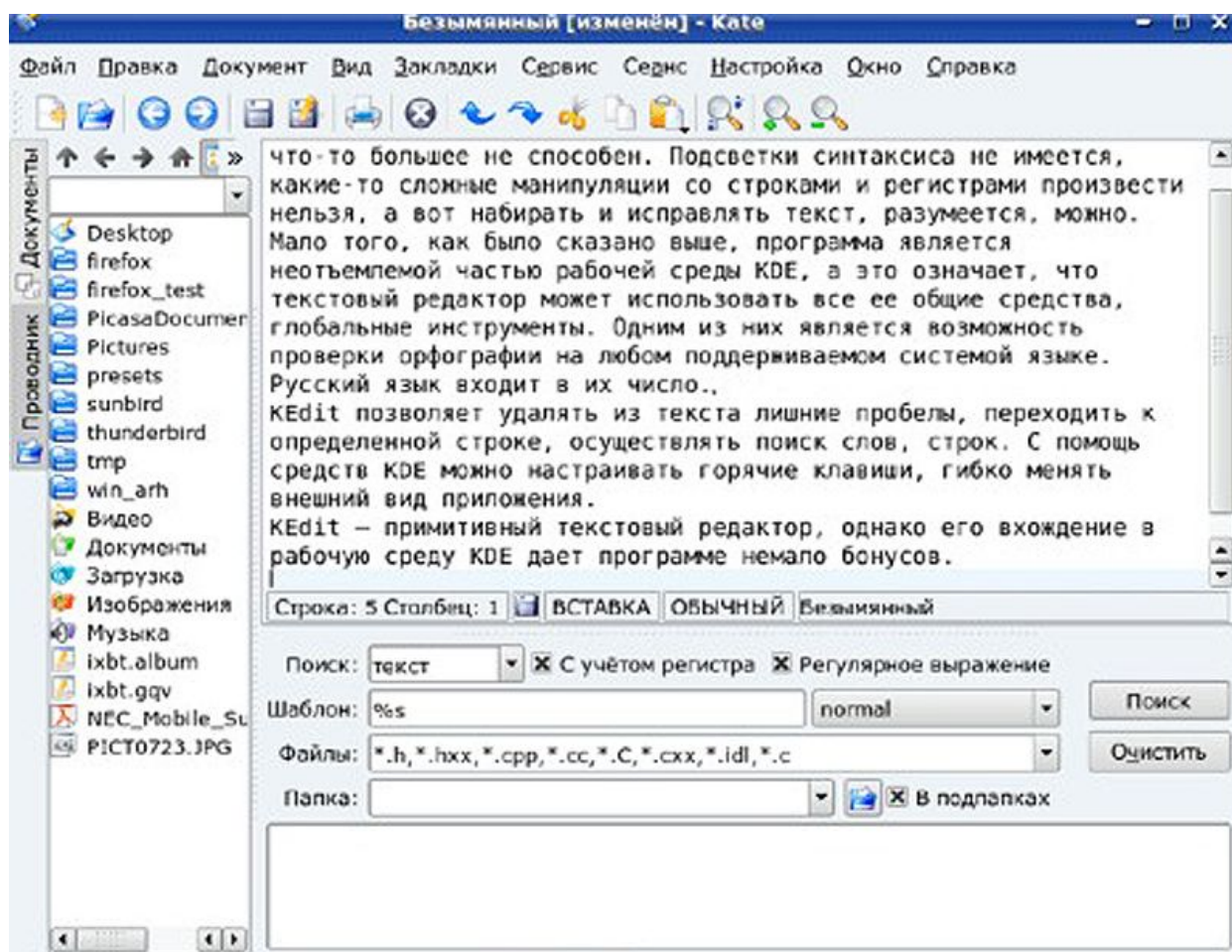


Рис.21. Поиск текста внутри файлов с помощью Kate

Kate поддерживает поиск текста не только внутри текущего документа, но и в любых файлах. Поддерживается большое число регулярных выражений.

Благодаря интеграции в KDE, вы можете настраивать горячие клавиши, присваивая их практически всем функциям текстового редактора. Поддерживается быстрая отправка документов по электронной почте, а также их распечатка на принтере с предварительным просмотром. Любая панель инструментов может быть скрыта или, напротив, восстановлена на экране. Вы можете настраивать кнопки на панелях, менять их размер, дизайн.

Внизу экрана находится меню **Терминал**, позволяющее запустить терминал в отдельной области прямо в текстовом редакторе и выполнить в нем необходимые команды.

Kate - мощный текстовый редактор, входящий в состав KDE. Он обладает многими достоинствами, наследуемыми от рабочей среды.

### Текстовый редактор KWrite

Еще один редактор *KWrite* — не просто текстовый редактор для среды KDE. Это также и редактор для программистов, и он может считаться, по крайней мере, частичной заменой более мощным приложениям. Его лучше всего использовать вместе с Konqueror для просмотра файлов с исходными кодами на разных языках. KWrite хорошо выполняет и обычные функции редактора текстов. Одна из его основных особенностей — подсветка синтаксиса, которая настраивается для разных языков программирования.

Работа с KWrite очень проста. Если вы хоть раз пользовались текстовым редактором, у вас не должно возникнуть проблем.

В KWrite используется собственный протокол KDE для манипулирования графическими объектами. Файлы в KWrite можно перетаскивать с рабочего стола, из Konqueror или, например, FTP-сайта, открытого в одном из окон Konqueror.

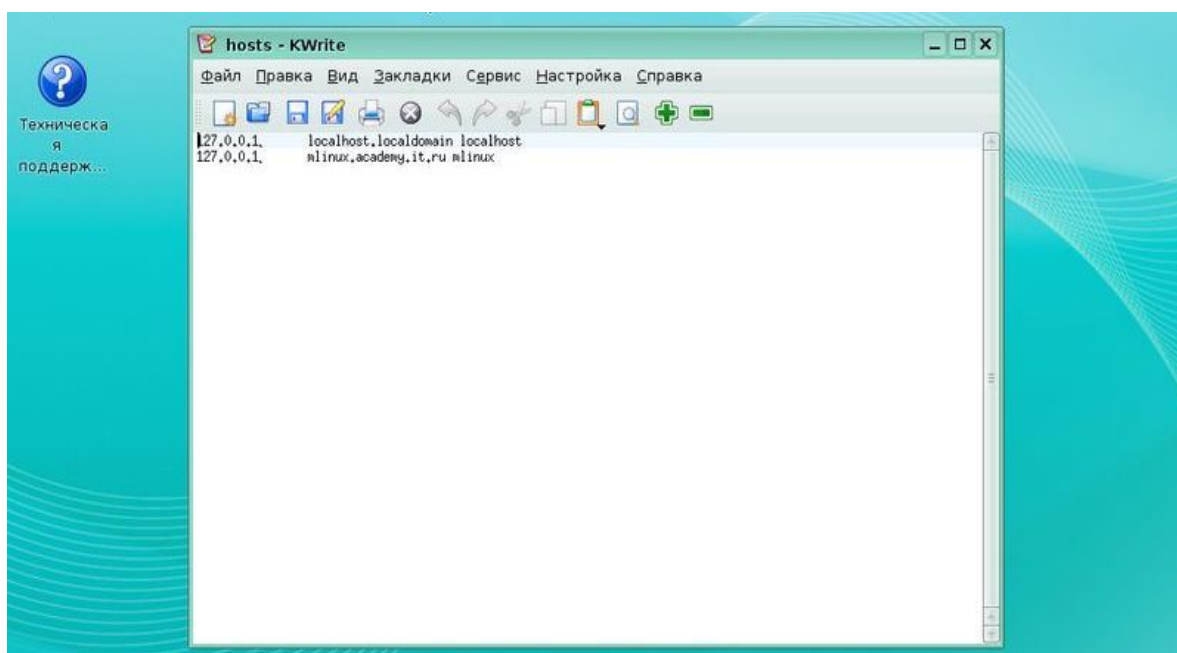


Рис.22. Текстовый редактор KWrite

## Продукт OpenOffice.org

*OpenOffice.org* — это свободный офисный пакет. OpenOffice.org успешно заменит вам любой другой офисный пакет, а открытая лицензия освободит пользователя от всех лицензионных отчислений, связанных с использованием пакета, в настоящем и будущем.

OpenOffice.org состоит из шести компонентов-программ:

- Текстовый редактор и редактор web-страниц Writer;
- Редактор электронных таблиц Calc;
- Средство создания и демонстрации презентаций Impress;
- Векторный редактор Draw;
- Система управления базами данных Base;
- Редактор для создания и редактирования формул Math.

Все компоненты интегрированы в общее ядро и, таким образом, обладают хорошей совместимостью друг с другом.

OpenOffice.org обладает рядом интересных особенностей, частично или полностью отсутствующих в других офисных пакетах, обеспечивает совместимость с форматами MS Office 97-2003. Также поддерживается импорт файлов в формате Office OpenXML (MS Office 2007).

Пользователи, ранее работавшие с MS Office, смогут работать в этом пакете без дополнительного обучения.

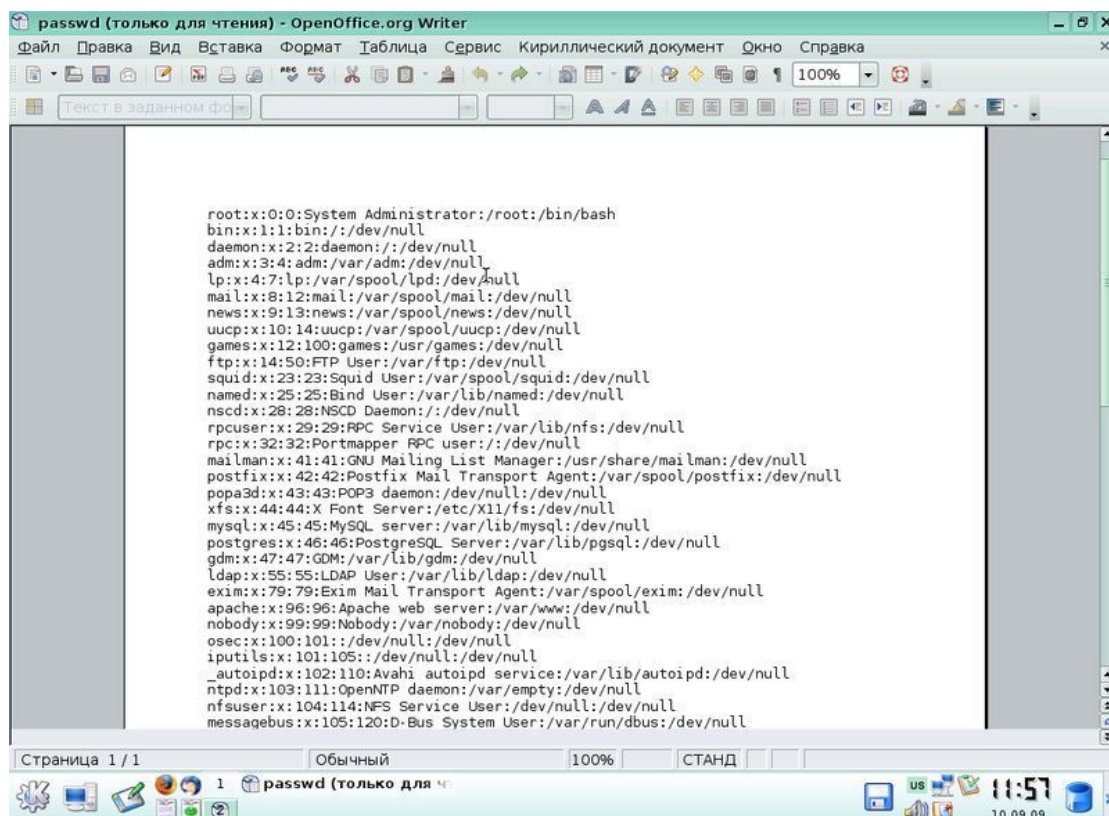


Рис.23. Текстовый редактор Writer из пакета OpenOffice.org

## **Рабочий стол**

Первое, что стоит сделать, начав постоянно использовать Linux — организовать для себя удобное «рабочее место»: подобрать и настроить программы, с которыми приходится работать каждый день. Рабочее место в может выглядеть очень по-разному. Можно вовсе обойтись без графического интерфейса, используя только текстовый терминал для управления системой. Такой выбор будет правильным, если рабочее место находится на сервере, подключённом к сети Internet, доступ к которому осуществляется только при помощи ssh или аналогичных клиентов удалённого доступа.

Если графический интерфейс используется, то и в этом случае есть огромный выбор, как его организовать. Прежде всего, решить: нужно ли устраивать «рабочий стол» (для этого подходят GNOME, KDE) или можно обойтись возможностями одного из развитых диспетчеров окон (Enlightenment, WindowMaker и многие другие). Помимо функциональности, в выборе графической среды решающее значение могут сыграть и эстетические критерии. Дальше всех в этом направлении продвинулась среда Enlightenment, работа с которой в некоторых вариантах настройки количеством украшений и эффектов напоминает участие в компьютерной игре (скорее всего, сетевой).

### ***Диспетчеры файлов (file manager)***

Поскольку представление файлов и каталогов как папок и документов нужно в первую очередь в рамках метафоры рабочего стола, то и диспетчеры файлов для Linux разрабатываются, прежде всего, как приложения той или иной среды рабочего стола. В частности, и в KDE, и в GNOME есть свои диспетчеры — konqueror и nautilus соответственно, которые по совместительству служат www-браузерами.

### **Эмулятор терминала**

Даже для такой на первый взгляд тривиальной функции, как эмуляция терминала для X Window System, существует целый круг программ. Самая стандартная из них поставляется вместе с XOrg — xterm. Вариант xterm, поддерживающий отображение шрифтов в кодировке UNICODE, вызывается командой `uxterm`. Однако каждое приложение, организующее среду рабочего стола, включает собственный эмулятор терминала, внешний вид и поведение которого настраивается централизованно вместе со всеми остальными приложениями рабочего стола.

### **Сеть. WWW-браузеры**

*WWW-браузер* — программа для просмотра гипертекста, доступного через Internet — на сегодня чуть ли не самое важное приложение для персонального компьютера. Сегодняшний www-браузер должен «уметь» гораздо больше, чем просто отображать страницы HTML и переходить по гиперссылкам. Фактически, на него ложится задача

работы данными Internet во всём их многообразии, сюда входит и поддержка постоянно развивающихся стандартов, и обеспечение безопасности, и многое другое.

В Linux есть довольно большой выбор www-браузеров, однако первым действительно современным свободным приложением для работы с Internet стала Mozilla, а затем её потомки, которые сегодня вполне успешно конкурируют с аналогичными коммерческими программами. Mozilla — это целый пакет приложений для работы с Интернетом: мощный, насыщенный функциями коммуникационный центр для персонального компьютера. В состав пакета входит браузер, программа для работы с электронной почтой, календарём и редактор www-страниц. История Mozilla началась в 1998 году, когда фирма Netscape опубликовала исходные тексты своего браузера Netscape Navigator. Одно из важных свойств пакета Mozilla — его принципиальная расширяемость. В Mozilla реализован язык XUL на основе XML, при помощи которого очень легко разрабатывать дополнительные компоненты Mozilla, ориентированные на выполнение специальных функций.

Более современен FireFox, разрабатываемый командой Mozilla на основе исходных кодов, соответствующих только WWW-браузеру. Остальная часть FireFox написана полностью на XUL, поэтому разработка этой молодой программы идёт существенно быстрее и проще, её настройка считается самой гибкой среди www-браузеров, а главное, любой желающий может написать на высокоуровневых языках программирования XUL/JavaScript и опубликовать свой модуль расширения (т. н. plugin; на сегодня таких модулей известно более полутора).

Помимо Mozilla, существуют версия под Linux такого популярного браузера, как Opera.

Важная разновидность www-браузеров — текстовые браузеры, т. е. те, которые могут быть запущены в любом текстовом терминале Linux. Самый старый и известный из них, один из прототипов современных www-браузеров — Lynx. Он не имеет возможности отображать графическую информацию, но отлично поддерживает HTML, формы и таблицы. Современные версии поддерживают также соединения, защищённые при помощи SSL. Links — это текстовый браузер, на первый взгляд очень похожий на Lynx, но все же несколько отличающийся от него:

- умением работать с таблицами и фреймами;
- отображением цветов, указанных в HTML-странице;
- использованием выпадающих меню (как в Midnight Commander);
- возможностью загрузки файлов в фоновом режиме.

## Почтовые программы

Thunderbird повторяет и расширяет почтовую составляющую Mozilla. Большинство сказанного о Thunderbird на сегодня справедливо и для MozillaMail. Так же, как и в Firefox, в Thunderbird используется часть исходного кода Mozilla, которая работает с сетью (на этот раз — с отсылкой почты и доступом к почтовым ящикам), а интерфейс и архитектура приложения в целом — переделаны для того, чтобы избавиться от стародавних частей Netscape и облегчить дальнейшую разработку. Thunderbird (как и MozillaMail) обладает самым мощным на сегодняшний день встроенным антиспам-фильтром. Как и Firefox, Thunderbird легко расширять собственными модулями, написанными на высокоуровневых языках, и можно очень гибко настраивать.

Ещё один почтовый клиент, несколько уступающий Thunderbird по возможностям, но превосходящий его по быстродействию, называется Sylpheed.

Поскольку управление электронной перепиской — одна из задач рабочего стола, в каждой среде рабочего стола есть свой собственный почтовый клиент. Почтовый клиент для KDE называется KMail, он поддерживает как локальную доставку почты, так и множество почтовых протоколов (POP3, IMAP, SMTP). Почтовый клиент для GNOME называется Evolution, он интегрирован с календарём, адресной книгой и претендует на функции индивидуальной «записной книжки».

## Обмен сообщениями

Если компьютер подключён к Internet постоянно, бывает удобно пользоваться службами, передающими сообщения в реальном времени (instant messaging service). Таких служб довольно много, самая популярная из них — ICQ. Множественность объясняется тем, что в большинстве случаев этот сервис предоставляется централизованно, какой-нибудь крупной корпорацией. Во многих случаях серверы этих служб не доступны под свободной лицензией. Исключение в ряду «собственников» — служба Jabber, основанная на полностью открытом протоколе XMPP. Jabber позволяет любому сообществу создавать собственные сервера, управляемые собственными администраторами. Сам Jabber-сервер имеет возможность соединять своих клиентов не только с другими Jabber-серверами, но и со службами ICQ, MSN, Yahoo и AIM. В Linux есть несколько клиентских программ для обмена мгновенными сообщениями. Особняком стоят клиенты IRC (Internet Relay Chat), службы с более долгой историей и сложным протоколом (имеется в виду и сетевой протокол, и протокол работы пользователя в IRC).

*Psi* — удобный графический клиент сети быстрого обмена сообщениями Jabber (а значит, по всем протоколам, которые поддерживает выбранный Jabber-сервер). Psi поддерживает такие возможности Jabber, как одновременная работа с несколькими



серверами, конференции, криптозащиту передаваемой информации (через SSL и GnuPG), работу через HTTP (S) прокси-сервер и т. д. SIM — многопротокольный клиент обмена мгновенными сообщениями. Поддерживаются протоколы ICQ, Jabber, MSN, AIM, YIM, а также LiveJournal. Кроме того, имеется множество модулей, реализующих дополнительные возможности. Есть вариант SIM, ориентированный на среду KDE. На среду KDE ориентирован и другой мощный клиент, имеющий поддержку также и IRC, — Kopete. На среду Gnome ориентирован Gaim — наиболее мощный и наиболее гибко настраиваемый клиент. Имеет модули доступа почти ко всем мыслимым протоколам, позволяет писать сценарии на Perl и TCL. Для IRC есть и специальные клиенты: ChatZilla (как можно догадаться из названия, он «встроен» в Mozilla, но доступен и как дополнение к Firefox) или X-Chat — весьма мощная программа, ориентированная на «хитрости» IRC.

### **Офисные программы**

Важной частью современной рабочей станции являются так называемые офисные средства обработки информации. Под офисными приложениями обычно понимают стандартный набор из текстового процессора, средства работы с электронными таблицами, средства создания презентаций, средства для работы с базами данных. Все перечисленные офисные приложения входят в пакет OpenOffice.org — это свободный набор офисных программ, не уступающий по возможностям несвободному Microsoft Office, а кое в чём даже превосходящий его. Например, частности, которая может иметь очень важное значение: компонент OpenOffice.org OpenWriter позволяет экспортировать документы непосредственно в формат PDF. Интерфейс OpenOffice.org устроен принципиально так же, как и у аналогичных продуктов Microsoft, так что пользователю, привыкшему к Microsoft Office, не составит большого труда перейти к работе в OpenOffice.org. Кроме того, OpenOffice.org позволяет работать со всеми форматами файлов Microsoft Office.

История OpenOffice.org напоминает историю Mozilla: поначалу проект (под именем StarOffice) развивался закрыто, без доступа мирового программистского сообщества к исходным текстам. Однако в 2000-м году компания Sun Microsystems открыла исходные тексты программного продукта, образовав OpenOffice.org. Так же, как и в случае Netscape/Mozilla, пара StarOffice/OpenOffice.org использует двойное лицензирование, дающее право как свободного доступа к исходным текстам, так и использования их в закрытых коммерческих продуктах.

По возможностям OpenOffice.org остаётся самым развитым и полным офисным пакетом для Linux, однако есть и другие офисные средства. В частности, офисный пакет Koffice, ориентированный на среду KDE, в котором есть примерно тот же набор офисных

приложений, что и в OpenOffice.org. Кроме того, есть отдельные офисные приложения, не составляющие пакетов — словарный процессор Abiword и электронные таблицы GNUmeric.

## **Графика**

Чем проще пользовательская задача, тем больше программ под Linux её решают. В частности, манипуляция геометрическими фигурами с возможностью изменения их параметров (цвета, размера и т. п.), хранением набора фигур в файле и преобразованием получившегося изображения в растровый формат — довольно простая задача, требующая аккуратной реализации основных функций какой-нибудь высокоуровневой библиотеки (или двух — интерфейсной и графической). Неудивительно, что редакторы с подобными возможностями есть и для каждого рабочего стола, и независимо от них. Это утверждение относится и к ещё более простым программам работы с растровой графикой. Ниже описаны только существенно более сложные программы.

### **Векторная графика**

Векторной графикой называется способ работы с изображениями, при котором оно представлено в виде фигур, каждая из которых имеет собственное описание (тип, размеры, кривизну или иные параметры составных частей, их цвета, способ представления и т. п.). Некоторые графические устройства (например, распознающие формат PostScript) умеют сами интерпретировать описания фигур, для других необходимо заранее просчитать и сформировать картинку программным путём.

### **Работа с PostScript и PDF**

Для современной полиграфии de facto стандартом является формат PostScript. *PostScript* — это язык описания страницы, позволяющий представить любые полиграфические материалы в векторном формате (однако он допускает и включение растровых фрагментов). Файл в формате PostScript фактически представляет собой программу, описывающую, какие действия нужно произвести, чтобы получить требуемый вывод. Профессиональные печатающие устройства умеют самостоятельно интерпретировать документы на языке PostScript.

*PDF* (Portable Document Format, переносимый формат документов) создан на основе языка PostScript. Его основная задача — обеспечить одинаковый внешний вид документа в любой операционной системе. В PDF есть специальные возможности для публикации документов в Сети, в частности, поддержка гиперссылок, а некоторые возможности языка PostScript оттуда, наоборот, исключены.

*Ghostscript* — интерпретатор языка описания страниц PostScript и файлов в формате PDF (формат переносимых документов). Ghostscript преобразует PostScript во

многие растровые форматы, подходящие для вывода данных на экран или на принтер, не поддерживающий PostScript. Обычно Ghostscript используется для просмотра файлов PostScript и для печати на принтерах, не поддерживающих язык PostScript, GhostScript используется множеством приложений для вывода данных на печать. Графический интерфейс для GhostScript предоставляет программа GhostView (команда `gv`), она позволяет отображать документы в форматах PostScript и PDF в графической среде X Window System. Для различных манипуляций с файлами в формате PostScript предназначен пакет утилит командной строки `psutils`, с их помощью можно выбрать, переупорядочивать, масштабировать страницы в PostScript-файлах, изменять параметры текста и делать многое другое.

Специально для просмотра PDF-файлов предназначена программа `xpdf`, она позволяет переходить по гиперссылкам в документе, просматривать структуру документа, производить поиск и поддерживает сглаживание шрифтов. На основе исходных текстов `xpdf` создана библиотека `poppler`, предназначенная для отрисовки PDF, которая стала основой других популярных приложений для просмотра PDF: `kpdf` (компонент графической среды KDE, отличается более богатыми интерфейсными возможностями) и `Evince`. Многие дистрибутивы Linux включают `Acroread` — версию известного приложения Adobe Acrobat для Linux, однако, в отличие от названных выше, оно является несвободным программным продуктом.

## Диаграммы

Отдельно стоит упомянуть редакторы диаграмм и блок-схем, которые часто смешивают с обычными редакторами векторной (плакатной) графики. Между тем, задачи у них разные: если для плакатной графики главное — построение «картинки», соответствующей задумке автора по внешнему виду, то в диаграмме автора более беспокоит логическое соответствие изображения проекту и его наглядность. Поэтому при построении диаграммы много внимания уделяется «стрелочкам» и прочим соединительным линиям, оптимальному размещению объектов на странице, типизации объектов и т. п.

Самая старая из подобных утилит, `xfig`, и по сей день активно используется, формат её диаграмм распознают многие средства работы с векторной графикой. Более мощной является утилита `Dia`, возможности которой продолжают расти (среди проектов: перевод диаграмм, представленных в нотации языка моделирования программных продуктов UML, непосредственно в текст программ на C++ и других языках). Аналогом `Dia` для KDE является встроенная в пакет KOffice утилита `Kivio`.

## Плакатная графика

Что же касается собственно векторной (плакатной) графики, то и здесь есть из чего выбирать. Например, Inkscape — программа векторного рисования общего назначения. Она использует в качестве формата собственных файлов W3C SVG и обладает не только полным набором базовых функций работы с векторными объектами и слоями, но и рядом функций, аналогов которым нет и в крупных закрытых продуктах, таких как Adobe Illustrator или Corel DRAW. Кроме того, в ней реализован механизм расширений, благодаря которому к программе можно дописать новые функции на языках Perl, Python и Ruby. Inkscape — программа Sodipodi, обладающая сходными возможностями и более продолжительной историей. Многообещающе выглядит проект Skencil, позволяющий редактировать некоторые виды PostScript-файлов. Среда KDE также имеет «свой» редактор векторной графики, Karbon14, входящий в состав семейства программа KOffice.

## Растровая графика

Растровая графика означает работу с изображением, представленным в виде матрицы точек («пикселей»). Это значит, что при сильном увеличении границы любого объекта будут выглядеть «лесенкой» из точек (в отличие от векторного представления, где увеличение повышает качество изображения).

В GNU/Linux есть развитые средства для редактирования растровой графики. Самым мощным из них является GIMP (GNU Image Manipulation Program). С её помощью пользователь сможет редактировать изображения, создавать логотипы и другие графические элементы, особенно полезные при создании Web-страниц. GIMP включает много инструментов и фильтров, аналогичных тем, которые можно найти в коммерческих графических редакторах, а также несколько возможностей, эксклюзивных для этой программы. GIMP предоставляет возможность работать с цветовыми каналами, уровнями изображения, накладывать эффекты, сглаживать шрифты и конвертировать изображения в разные форматы. В GIMP имеется собственный язык программирования сценариев (на основе Scheme), на котором можно создавать довольно замысловатые дополнения к основной программе. Такие дополнения можно писать также на Tiny-Fu (облегчённая версия Script-Fu), Python, Perl и C#. Недостаток GIMP — слабая поддержка цветовой модели CMYK, используемой в полиграфии, поэтому в электронной документации, редактировании изображений для www-страниц и прочих областях, не имеющих дела с бумагой, его применяют чаще.

Очень полезен набор утилит для обработки графики из командной строки — ImageMagick. В этот набор входят утилиты для отображения (display), преобразования (convert) изображений, захвата изображений с экрана (import) и даже собственный

интерпретируемый язык программирования, Magick Scripting Language. Для полуавтоматического перевода из растрового представления в векторное существует несколько специальных утилит, например, autotrace/autofig или potrace.

### **Трёхмерная графика**

Для Linux создано несколько программных пакетов, работающих с пространственными представлением объектов.

Исходные тексты одного из самых мощных пакетов трёхмерного моделирования, пересчёта (рендеринга) и анимации — Blender — в 2002 году были открыты и весь проект полностью переведён под свободную лицензию.

Для выполнения задач, совмещаемых Blender, есть и отдельные программные средства. Например, популярный пакет трассировки лучей (трёхмерного проектирования и сценографии) POV-Ray, с помощью которого создаются проекты удивительной сложности и красоты. Многие графические редакторы имеют встроенные средства анимации, а иные (как, например, CinePaint, называвшийся ранее FilmGimp) специально разрабатываются для кадровой обработки видео.

Не стоит забывать, что популярный нынче стандарт OpenGL — открытый; он разрабатывался для UNIX-подобных систем, и используется большим числом программ для Linux (в том числе и Blender). К сожалению, производители аппаратного обеспечения (видеокарт), как правило, скрывают не только устройство своих карт, но даже и способ их низкоуровневого использования. Поэтому в открытом доступе оказываются лишь готовые драйверы (без исходных текстов) к некоторым версиям ядра Linux и определённым сборкам XOrg. Отображение трёхмерных объектов с пересчётом на программном уровне пока работает существенно медленнее, хотя ничуть не хуже, поэтому используя OpenGL для игр и прочих программ, требующих действительно быстрой работы графической подсистемы, нужно всегда помнить о необходимости получить — возможно, несвободный — драйвер.

### **Мультимедиа**

#### **Музыкальные шкатулки**

Очевидный лидер по популярности среди этого класса программ — XMMS (X Multi Media System). Помимо основной функции — играть музыку (поддерживается множество форматов) — в нём реализовано немало звуковых и визуальных эффектов благодаря большому количеству расширений. Интерфейс XMMS аналогичен интерфейсу не менее популярного в системах Windows приложения WinAMP (кстати, XMMS умеет использовать «шкурки» WinAMP2). Почти не уступает XMMS программа BEEP, использующая графическую библиотеку GTK2, а не GTK. Есть и другие программы,

которые ничуть не хуже этих играют музыку. Обычно каждая среда рабочего стола реализует собственный проигрыватель звуковых файлов, хотя бы для того, чтобы воспроизводить собственные звуковые эффекты, связанные с различными системными событиями, однако с их помощью прослушивать файлы может и пользователь.

Очевидно, что для прослушивания звука совсем не обязательно использовать графический интерфейс, поэтому в Linux есть большое количество терминальных утилит для воспроизведения звука. Некоторые из них, например, `mpg123`, `mpg321`, `ogg123` или `splay`, предназначены для проигрывания оцифрованного звука. Другие, такие как `lazy` или `cd-console`, управляют музыкальными *лазерными* дисками. Есть утилиты, играющие музыку в нотном (`midi`) и других форматах — `timidity`, `mikmod`, `sidplay` и прочие. Чтобы пользователь не запутался, специальные оболочки, например `mpfc` или `splay`, предоставляют общий интерфейс ко всем консольным проигрывателям.

### **Музыкальные редакторы**

Часть профессиональных музыкантов предпочитает использовать для записи и сведения многоканального звука дорогие *специализированные* цифровые станции: в этом повинна и реклама, и низкое, с точки зрения профессионала, качество звука большинства звуковых карт в компьютерах общего назначения. Несмотря на это и для таких компьютеров существует немало программ, работающих со звуком на профессиональном уровне. Такие программы можно разделить на две категории: нотные редакторы, задача которых — создание, редактирование, запись и нотное представление музыкальных композиций, и звуковые редакторы для собственно звука, а также преобразования его, наложения эффектов и т. п.

### **Нотные редакторы**

В операционных системах, основанных на GNU/Linux, также присутствуют мощные программы для редактирования музыки и звука. Пожалуй, самым известным из них является Rosegarden. Программа изначально разрабатывалась для профессиональных мультимедиа-станций от Silicon Graphics и работала на операционной системе IRIX, потом она была перенесена на Linux, а исходные тексты программы были *открыты*. Сегодня Rosegarden представляет из себя развитый MIDI- и аудиосеквенцер, нотный редактор, а также редактор общего назначения для сочинения и редактирования музыки. Он прост в изучении и идеально подходит для композиторов, музыкантов или студентов музыкальных специальностей, работающих в маленькой студии или записывающихся дома.

*Noteedit* — нотный редактор (редактор партитур), основанный на MIDI-библиотеке TSE3. Он может писать и читать MIDI-файлы и сигналы от внешней MIDI-клавиатуры.

Системные MIDI-устройства используются для воспроизведения нотной записи. Имеется возможность сохранить партитуры в формате MusiXTeX или Lilypond для последующего вывода на печать.

### **Редакторы и фильтры оцифрованного звука**

Популярный свободный редактор звука — Audacity. Он умеет записывать звук сразу в форматы WAV, AIFF, AU, IRCAM или MP3. В нём есть всевозможные инструменты для редактирования записанного звука, в том числе встроенный редактор амплитуды, настраиваемый режим отображения спектрограммы и средства частотного анализа звуковых дорожек. Встроенные простейшие аудиоэффекты включают усиление баса, WahWah, удаление шума и т. д. Audacity поддерживает модульные дополнения, в которых обычно поставляются более сложные аудиоэффекты. В список поддерживаемых форматов модулей входят VST, LADSPA и Nyquist.

*Sweep* — это многоканальный звуковой редактор, в котором реализованы все основные операции, такие как удаление, копирование, вставка и применение эффектов, оформленных в виде плагинов, к любой части звукового файла. Примерно теми же возможностями обладают и другие редакторы звука — ReZound, WaveSurfer и GNUSound.

Как и в случае с другими мультимедиа-форматами, в Linux существуют терминальные утилиты для обработки звука, не требующие графического интерфейса. Основной пакет терминальных утилит для работы со звуком называется SOX, в него входят утилиты для преобразования, записи и проигрывания звуковых файлов, поддерживается множество форматов.

### **Видеопроекторы**

Наиболее полнофункциональным и удобным «домашним кинотеатром» для Linux является программа xine. Xine поддерживает mpeg-2 и mpeg-1 (включая DVD) потоки, MPEG-4 и другие форматы. Альтернативный ему универсальный проигрыватель — MPlayer. Существует приложение для проигрывания *видеопотока*, получаемого по Сети — VideoLAN (vlc), которое работает с форматами MPEG1, MPEG2, MPEG4 (также известный как DivX) и DVD.

*Hawtv* — программа для просмотра и записи видеопотоков Video4Linux, то есть программа для просмотра ТВ. Hawtv использует набор графических элементов Athena. Может использоваться совместно с VDR для *просмотра* цифрового спутникового, кабельного и эфирного ТВ формата DVB.

### **Видеоредакторы и конвертеры**

В Linux есть выбор средств для преобразования и обработки видео. LiVES (the Linux Video Editing System) претендует на звание простого, но мощного средства

редактирования и эффект-обработки видео. Базируясь на GTK+, оно использует для работы такие широко распространённые средства, как MPlayer/mencoder и ImageMagick (в будущем, возможно, GStreamer и Xine). В настоящий момент рекомендуется использовать LiVES для работы с небольшими файлами.

*GStreamer* представляет собой библиотеку для обработки медиапоток, основанное на идее объединённых в графы фильтров, обрабатывающих медиаданные. Приложения, использующие эту библиотеку, смогут производить любую обработку медиаданных от обработки звука до проигрывания видео. Модульная архитектура позволяет реализовать поддержку любого нового формата данных, просто установив соответствующее расширение.

*Kino* — это нелинейный редактор цифрового видео (DV) для GNU/Linux. Он хорошо интегрирован с IEEE 1394 и позволяет захватывать изображение, управлять VTR, и записывать на камеру. Этот редактор записывает видео на диск в формате AVI в кодировках type-1 DV и type-2 DV. Существуют терминальные утилиты для обработки видеопотока, например, пакет transcode. Кодирование и декодирование видеопотока осуществляется с помощью загружаемых модулей. Также поддерживается загрузка внешних фильтров.

*Ffmpeg* — это «сверхзвуковой» кодировщик/декодировщик видео и звука, работающий в режиме реального времени, а также потоковый сервер и преобразователь различных звуковых и видеоформатов. Ffmpeg умеет захватывать видеосигнал из источника Video4Linux и преобразовывать его в файлы различных форматов на основе компенсирующего кодирования DCT/motion. Звук при этом сжимается по алгоритму MPEG-2 или алгоритму, совместимому с AC3.

### **Запись CD и DVD**

Для записи дисков и сопровождающих запись задач в Linux есть как минимум два приложения с графическим интерфейсом: входящее в комплект приложений для KDE k3b и написанное на GTK xcdroast. Фактически, оба этих приложения — это графические оболочки над терминальными утилитами для записи CD и DVD, в первую очередь cdrecord и cdrdao, которыми можно пользоваться и непосредственно из командной оболочки. *cdrecord* — утилита для записи дисков с цифровыми данными, в нём реализована полная поддержка аудио-, смешанных, мультисессионных и CD+ дисков. *cdrdao* — программа записи аудиодисков в одну сессию позволяет управлять областями в начале дорожек данных (длиной до 0, ненулевые аудиоданные) и, например, международными стандартными кодами записи. Все данные, которые будут записаны на



диск, должны быть описаны в текстовом файле. Аудиоданные могут быть в форматах WAVE или raw.

### **Издательские системы**

Подготовка печатных документов и оригинал-макетов изданий — хоть и не очень распространённое, но важное приложение компьютера. В Linux самой известной и системой подготовки качественных документов, пригодных к печати в типографии, является TeX. TeX — это фактически специализированный язык программирования, специально разработанный для описания типографского набора. Документ в TeX представляет собой текст, сопровождаемый командами, указывающими, какое форматирование следует произвести. Возможности TeX очень широки, однако для того, чтобы их использовать в полной мере, требуются довольно серьёзные познания в нём. Чем шире познания — тем легче, быстрее и удобнее готовить документы в TeXе и тем лучше их качество.

Обычно TeX используется совместно с пакетами форматирования более высокого уровня, например, LaTeX. LaTeX — это комплекс написанных на языке TeX макропакетов, предоставляющих удобные средства для решения типичных задач оформления печатных изданий. В LaTeX определено оформление для нескольких стандартных классов документов.

LyX — это современный подход к написанию документов, разрывающий с устаревшей парадигмой использования компьютеров как пишущих машинок, применяемой в большинстве других систем подготовки документов. Он разработан для тех, кто хочет получить профессиональное качество документа при печати, не тратя при этом много времени и усилий, и не становясь специалистом по полиграфическому оформлению. Основное новшество в LyX — это WYSIWYM (What You See Is What You Mean — вы видите то, что вы имели в виду), которое означает, что автор сосредотачивается над своей работой, а не над деталями оформления документа. Это позволяет продуктивно работать, оставляя заключительное оформление специальному движку (такому как LaTeX), который специально разработан для подобных задач. С LyX автор может сконцентрироваться на содержании своей работы и позволить компьютеру взять большинство забот об оформлении на себя.

В Linux есть, по крайней мере, одна программа для визуальной подготовки оригинал-макетов, аналогичная издательским системам Adobe PageMaker, QuarkXPress и подобным — scribe. Возможности его могут быть более ограничены, чем у перечисленных коммерческих аналогов, однако он распространяется свободно и в настоящее время активно разрабатывается.

## 2.5. Настройка настольной системы с помощью графических утилит

### 2.5.1. Настройка меню графической среды KDE

Главное меню в ОС Linux также называется **К-меню**. Из него вы сможете запускать различные программы.

1. Левой кнопкой мыши щелкните кнопку с буквой К, ее вы можете найти на панели внизу слева. При этом откроется **К-меню**.
2. В **К-меню** поочередно подводите указатель мыши к пунктам **Приложений** и ознакомьтесь с перечнем программ, которые установлены на вашем компьютере.

Вы также можете настроить **К-меню** самостоятельно. Для этого:

- Щелкните правой кнопкой мыши на панели задач
- Выберите пункт контекстного меню **Настроить панель...**

Если такого пункта меню не появляется, а есть пункт **Разблокировать панели**, то выберите его, после чего настройки панели можно будет редактировать ранее описанным способом.

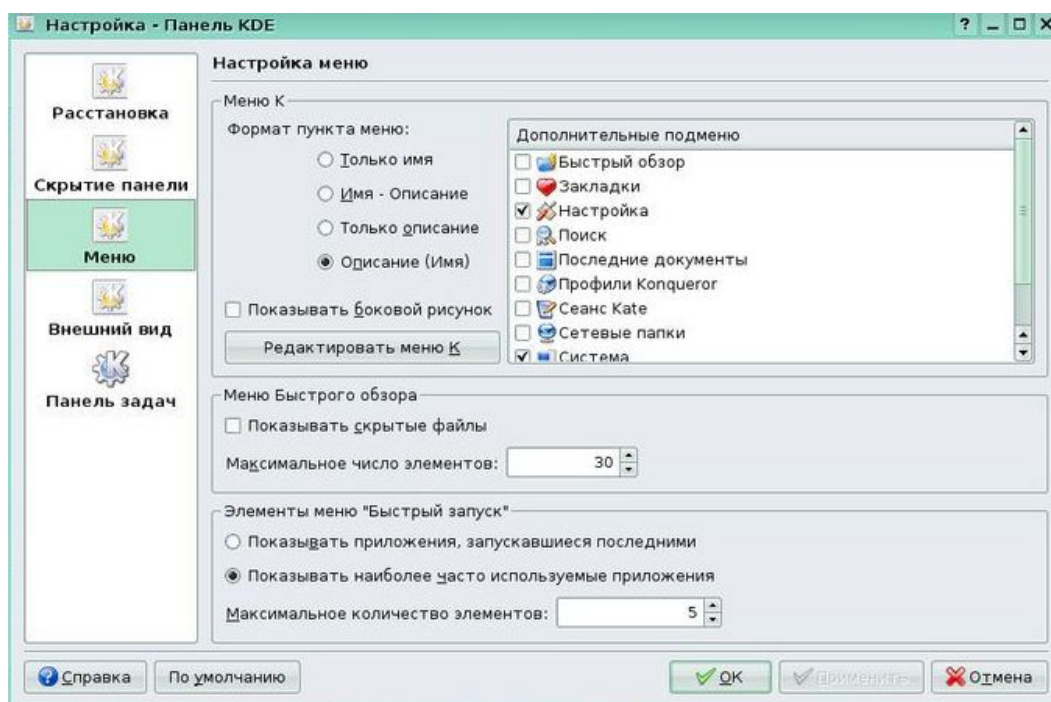


Рис.24. Настройка главного меню KDE

Щелкните раздел **Меню**.

В разделе **К- меню** можно настроить его функциональность.

В разделе **Формат пункта меню** можно установить режим отображения – в виде имени, описания, показывать ли боковой рисунок.

Параметры **Дополнительные подменю** определяют, какие дополнительные пункты будет содержать **К-меню**. Ознакомьтесь с представленной ниже информацией и выберите нужные разделы для вашего главного меню.

- Подменю **Закладки** позволяет перейти по любой закладке Konqueror.
- Подменю **Поиск** предоставляет средства поиска среди локальных файлов и в Интернете.
- Подменю Профили **Konqueror** содержит профили программы **Konqueror**.
- Подменю **Сетевые папки** открывает окно программы **Konqueror**, показывающее настроенные сетевые папки.
- Подменю **Настройка** позволяет загрузить любой из модулей Центра управления KDE непосредственно из меню К.
- Подменю **Система печати** содержит несколько пунктов для управления принтерами, доступными в вашей системе.
- Подменю **Быстрый обзор** позволяет быстро перемещаться по папкам файловой системы.
- Подменю **Последние документы** содержит список последних открывавшихся документов и автоматически запускает программу для просмотра или редактирования выбранного документа.
- Подменю **Система** используется для быстрого доступа к некоторым часто посещаемым местам, включая домашнюю папку и корзину. Оно может оказаться удобнее отдельных пунктов в **К-меню**.
- Подменю **Терминальные сеансы** содержит пункты для загрузки разного типа терминалов (интерфейсов командной строки).

Кнопка **Редактировать меню** позволяет вручную настроить связь между пунктами К-меню и запускаемыми приложениями.

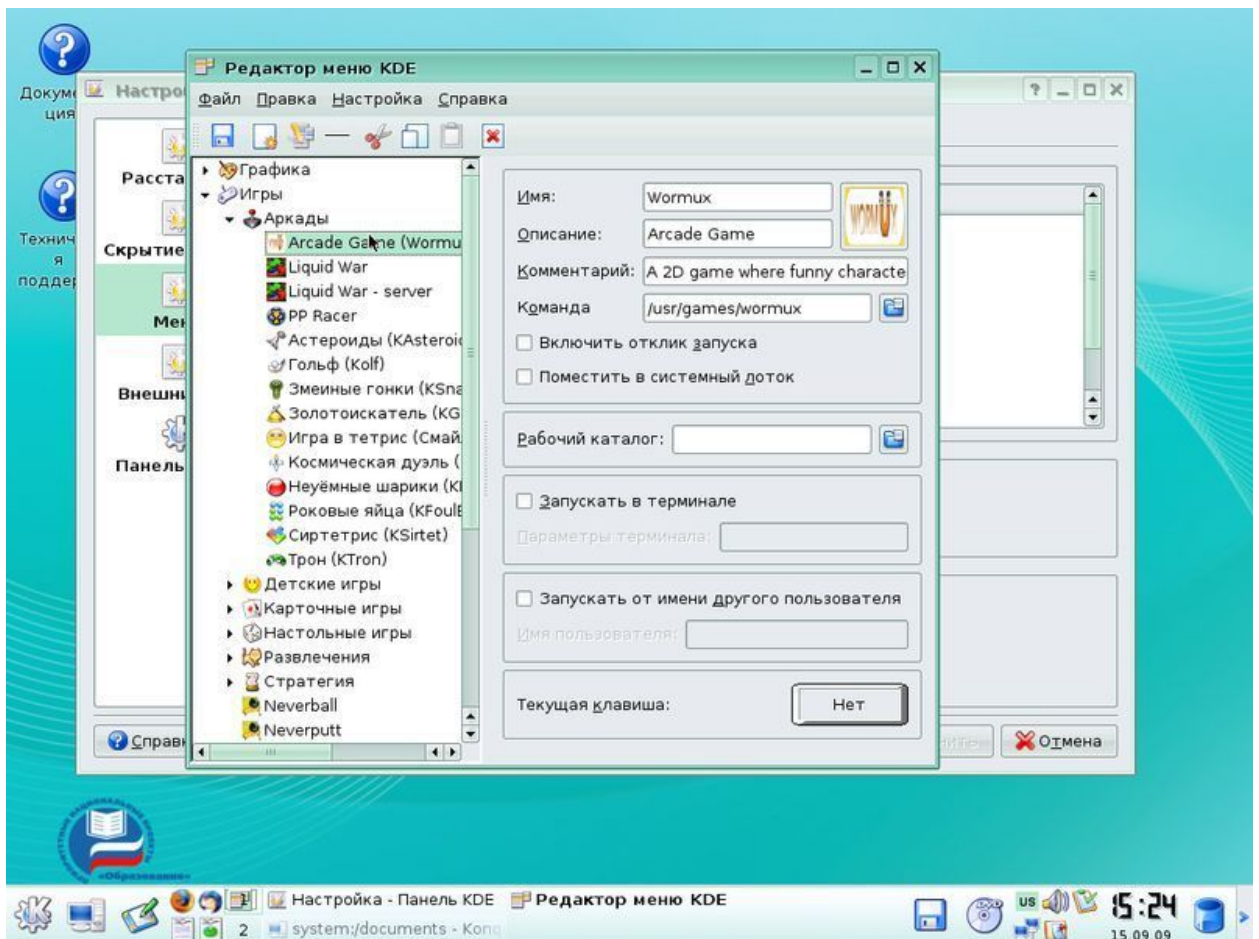


Рис.25. Настройки программ в меню KDE

Здесь вы можете указать, какая программа вызывается при активации пункта меню, иконку, отображаемую рядом с названием программы, установить параметры запуска программы, например учетную запись, из-под которой она будет запускаться, связать вызов программы с функциональной клавишей на клавиатуре.

В остальных разделах настройки панели KDE можно настроить следующие параметры:

**Расстановка** – панель может быть не только внизу экрана, а справа, слева или сверху. Тут так же можно установить длину панели и ее ширину.

**Скрытие панели** – устанавливаются параметры автоматического скрывания панели. Дополнительно в KDE имеется удобная возможность, панель может скрываться не вниз, а вправо или влево, при этом на экране остается маленькая стрелка, щелкнув по которой можно восстановить панель.

**Внешний вид** – позволяет настроить фон, прозрачность.

**Панель задач** – настройки отображения запущенных программ в панели: группировка схожих задачи, показ во всех рабочих столах или только в текущем, настроить цвета.

## 2.5.2. Запуск программ

Запускать программы в KDE можно различными способами.

- Через **К-меню - Из соответствующего раздела**
- Через **К-меню - Выполнить программу**. В окне **Команда** следует указать имя программы для запуска, если программа находится не в стандартном каталоге, то ее имя следует указать полностью. Например /home/user/app1.

Окно **Параметры** позволяет запустить программу с другим приоритетом (быстрее или медленнее), запустить программу в текстовой консоли или с привилегиями другого пользователя. Для этого достаточно щелкнуть кнопку **Параметры** и указать логин и пароль пользователя, от имени которого данная команда будет запущена.

- Комбинацией клавиш **Alt + F2**.

### Управление запущенными программами.

Все запущенные программы отображаются на панели. Щелкнув правой кнопкой мыши на программе, вы можете управлять окном: свернуть, развернуть, переместить на другой виртуальный рабочий стол, закрыть программу.

## 2.5.3. Панель управления KDE

Настройка KDE - процесс многогранный. Почти все параметры внешнего вида и поведения KDE можно настроить через **КСС - Центр управления KDE**. И это - штатный способ конфигурирования, теоретически рассуждая, у пользователя не должно возникать потребности обращаться к прямому редактированию файлов конфигурации (за редкими исключениями, о которых речь пойдет далее). КСС позволяет в графическом интерфейсе управлять наиболее востребованными настройками системы: пользователями, сетевыми подключениями, периферийным оборудованием, правами доступа и т. п. Центр управления системой состоит из нескольких независимых диалогов — модулей. Каждый модуль отвечает за настройку определённой функции или свойства системы и может быть запущен автономно. Так что с общего обзора КСС мы и начнем.

Вызов КСС осуществляется из стартового **К-меню** - через пункт Центр управления.

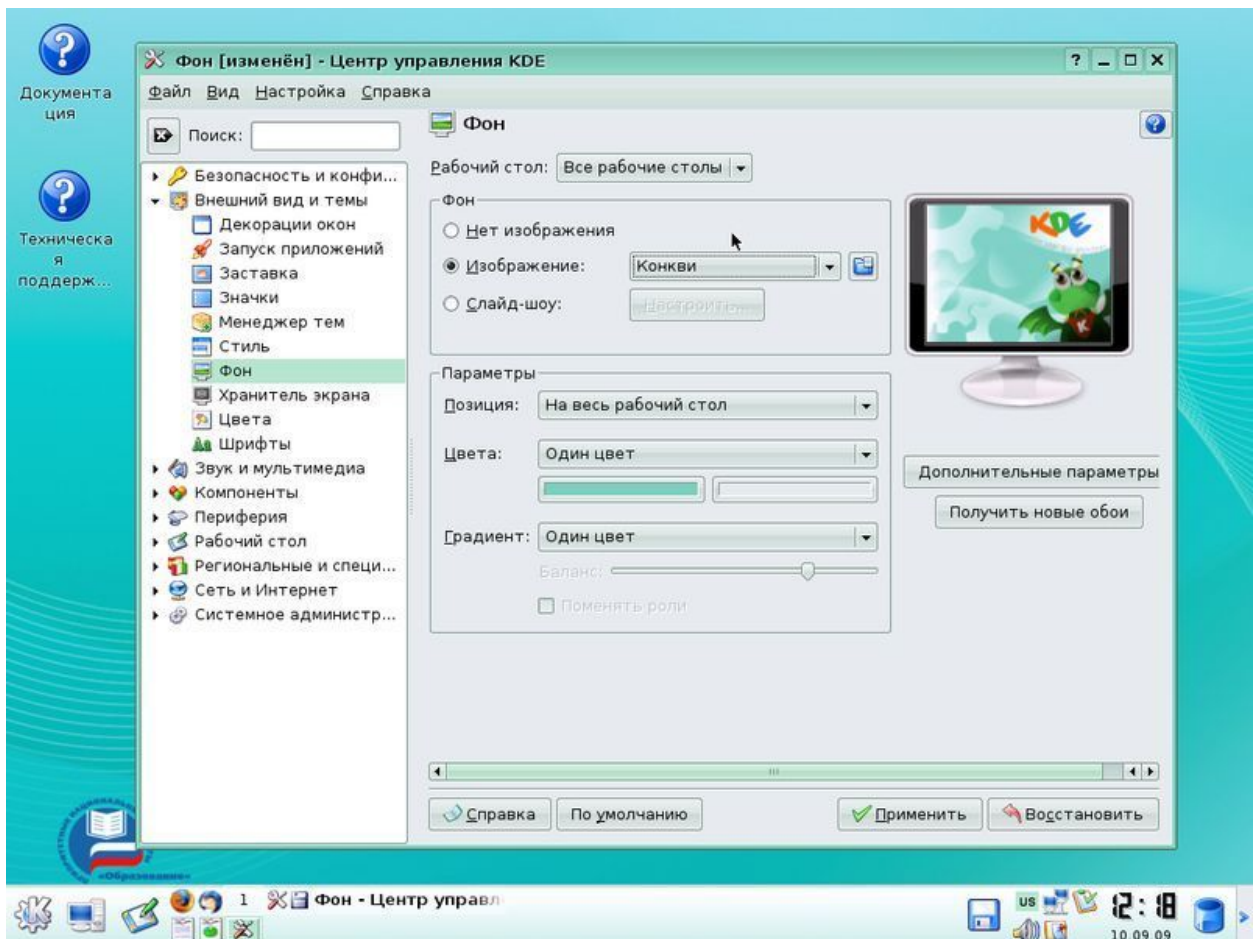


Рис.26. Центр управления KDE

Можно видеть, что окно Центра управления состоит из двух фреймов, левый включает в себе меню, правый же - расшифровку его основных пунктов и настроечные панели к отдельным из них.

Пункты меню отсортированы по алфавиту и в русскоязычном варианте имеют следующий порядок:

1. Безопасность и конфиденциальность
2. Внешний вид и темы
3. Звук и мультимедиа
4. Компоненты KDE
5. Периферия
6. Рабочий стол
7. Региональные и специальные возможности
8. Сеть и Интернет
9. Системное администрирование

Рассмотрим основные пункты.

## Безопасность и конфиденциальность

Смысл компонентов этого пункта вполне ясен из вводных комментариев - это очистка кэша, шифрование, изменение некоторых параметров пользователя. Программа **kwallet** представляет собой базу данных для хранения пользовательских паролей - как локальных, так и удаленных (например, для доступа к ftp-серверам), избавляя от необходимости запоминать их.

## Внешний вид и темы

Можно настроить практически все визуальные элементы интерфейса KDE, как:

1. Декорации окон
2. Запуск приложений
3. Значки
4. Темы (подпункт Менеджер тем)
5. Стиль
6. Фон
7. Хранитель экрана
8. Цвета
9. Шрифты
10. Экран-заставка

## Звук и мультимедиа

В этом пункте можно настроить параметры воспроизведения аудио-CD и доступа к базе данных музыки в Интернет, изменить характер системных звуковых сообщений в ответ на разного рода события и, конечно же, сконфигурировать собственно систему воспроизведения звука. Конечно, здесь можно просто включить/выключить воспроизведение звука (по умолчанию - включено) и установить приоритет звуковоспроизведения. Но главное - это выбор звукового устройства.

*Настройка компонентов KDE* - весьма разнообразна по своему смыслу.

Во-первых, это **Быстродействие KDE**, под которым подразумевается режим кэширования (в соответствующей панели именуется минимизацией использования памяти - при достаточном ее объеме это можно отключить).

Во-вторых, это **Выбор компонентов**, что подразумевает смену компонентов по умолчанию: браузера, текстового редактора, клиента электронной почты, терминала на другие программы, хотя и не любые.

В-третьих, это **Менеджер сеансов**. Здесь устанавливается, нужно ли выводить предупреждение при выходе из сеанса KDE (по пунктам Завершить сеанс контекстного или стартового К-меню), восстанавливать ли при следующем запуске текущее состояние,

включая окна открытых приложений и загруженные в них файлы, а также - что собственно должно происходить по выходе из KDE - только ли завершение сеанса, перезагрузка системы или даже выключение машины.

Пункт **Настройка ресурсов KDE** относится к настройке возможности работы с календарями, находящимися как локально, так и удаленно.

В пункте **Привязки файлов** устанавливается связь определенных их типов (точнее, масок имен) с тем или иным приложением, например, файлов вида \*.txt - с текстовым редактором, а \*.avi - с медиаплеером. Большинство таких привязок уже выставлены по умолчанию - необходимость в их изменении возникает, скорее всего, при установке новых приложений (например, mplayer в дополнение к штатным poatun и kaboodle). Одна и та же маска файла может быть связана с несколькими приложениями - например, для html-файлов это могут быть браузер konqueror и web-редактор Quanta; в этом случае приоритет связи определяется порядком перечисления программ и может быть легко изменен.

Пункт **Управление службами** выводит панель со списком KDE-специфичных демонов (таких, как упоминавшийся ранее kwallet) и их статусом (Выполняется/Не запущен). Однако как раз управлять-то большинством из них не получится...

Наконец, пункт **Файловый менеджер** позволяет определить некоторые свойства программы konqueror в этом качестве. Некоторые из них (например, гарнитура, размер и цвет шрифта) дублируются собственными настройками konqueror, который подробно будет описан в отдельном разделе. Иные же (как кэширование операций копирования и перемещения файлов) присутствуют только здесь.

## **Периферия**

В этом разделе можно посмотреть и изменить свойства таких устройств, как экран, мышь, клавиатура и так далее. Правда, изменению поддается не так уж много параметров. Для экрана это его разрешение, баланс цветов и включение/выключение энергосбережения (то есть гашения при простое - не путать со скринсейверами).

Для клавиатуры здесь мы не найдем возможности смены раскладок (таковые имеют место быть совсем в другом месте), можно лишь включить задержки ее автоповтора и статус NumLock при запуске KDE. Для мыши можно переопределить левую/правую кнопки (для правой/левой, соответственно), сменить одинарный щелчок для открытия файлов и каталогов на двойной (как это принято в Windows), изменить вид курсора и установить его акселерацию. В это же разделе настраиваются принтеры, вы можете добавить новый принтер или изменить параметры службы печати, посмотреть



очередь печати и снять свое задание из очереди на печать или временно приостановить его.

### **Рабочий стол**

Этот раздел посвящен оформлению рабочего пространства в среде KDE как целостности: количества рабочих столов, вид и местоположение главной управляющей панели, представление стартового К-меню (отредактировать его состав можно также через этот пункт), привязку контекстных меню рабочего стола к кнопкам мыши и, наконец, правила поведения окон.

Однако о чем обязательно нужно сказать чуть подробнее - это о правилах «поднятия» и фокусировки окон. В Windows придание окну фокуса, или его активизации, и «поднятия» (то есть вывода окна на первый план) - понятия в большинстве случаев идентичные. В Иксах же окно может быть активным, даже будучи погребено под геологическими напластованиями других окон. Причем и «поднятие», и фокусировка окна не обязательно требуют щелчка на нем мышью - при соответствующих настройках оконного менеджера достаточно привести на окно курсор мыши.

Пункт **Виртуальные рабочие столы** позволяет изменить их количество с двух стандартных до максимального количества – 20. Это позволяет пользователям группировать приложения на разных виртуальных рабочих столах и потом легко переключаться между ними. Переключение осуществляется либо мышью на панели внизу экрана, либо с клавиатуры кнопками Ctrl+FN, где N – номер виртуального стола.

Пункт **Панели** позволяет создать более одной стандартной панели (внизу экрана). Есть возможность создать дополнительные панели сбоку или сверху экрана, вы можете настроить их размер, параметры скрытия и другие параметры. В этот же разделе можно настроить **К-меню**, указав формат подменю, какие из имеющихся подменю должны отображаться, настроить элементы быстрого запуска.

### **Региональные и специальные возможности**

Этот раздел также очень важен, и содержит такие пункты:

- **KHotKeys** - модуль для настройки клавиатурных комбинаций («горячих клавиш»), служащих для выполнения сложных команды и вызова приложений;
- **Привязка клавиш** - служит для определения «горячих клавиш» для выполнения простых повседневных действий, таких, как переключение рабочих столов, навигация по открытым окнам и т.д.;
- **Раскладка клавиатуры** - позволяет менять раскладку клавиатуры и подключать еще до двух дополнительных;

- **Специальные возможности** - предназначен для настройки звуковых сигналов, а также использования т.н. «залипающих» и «замедленных» клавиш;
- **Страна/область и язык** - установка локально-зависимых параметров.

В комментариях тут нуждается три пункта. Во-первых, *Привязка клавиш*. Здесь для начала можно выбрать привычную схему «горячих клавиш» - в стиле Windows, например, MacOS, Unix, и так далее. Затем в рамках выбранной схемы можно скорректировать привычные клавишные комбинации для обыденных действий. Для этого достаточно перейти на закладку **Последовательности привязок**, выбрать нужное действие из списка.

Здесь следует учесть, что большинство удобных клавишных комбинаций уже задействованы по умолчанию в любой из предлагаемых на выбор схем.

*Во-вторых, Раскладка клавиатуры*. Модуль управления клавиатурой KDE - kxkb полностью аннулирует все настройки клавиатуры X-сервера из конфигурационных файлов. И позволяет в принципе вообще обходиться без таких настроек, указав необходимые параметры в удобном графическом интерфейсе.

*В-третьих, Страна/область и язык*. Кроме собственно страны (например, России) и языка (скажем, русского, используемого для меню, вывода сообщений и прочего), здесь же определяются денежная единица, десятичные разделители, формат даты, дополнительные параметры - стандартный формат бумаги (A4 или Letter), система единиц измерения (метрическая или английская), число десятичных знаков после запятой.

Все локальные параметры настраиваются независимо друг от друга. То есть можно определить страну как Россия, однако, при стойком отвращении к русскоязычным меню, языком по умолчанию назначить английский (вводу/выводу русских букв это не воспрепятствует), в качестве десятичного разделителя указать точку вместо запятой (это требуется некоторым счетным программам), и так далее.

### **Сеть и Интернет**

Этот раздел содержит настройки браузера и параметры сетевой работы, например имя и пароль пользователя по умолчанию при подключении к сетевым ресурсам, устанавливает параметры прокси-сервера.

### **Системное администрирование**

Тут можно настроить пути для рабочего стола, папки автозапуска и документов. В этом же разделе можно подключить дополнительные шрифты, если вам почему-то не хватило стандартных.

### **2.5.4. Получение информации о системе**

Для расширенной настройки Linux, как правило, необходимо обладать детальной информацией. Например, чтобы настроить сеть Вы должны поинтересоваться у своего

системного администратора о параметрах настройки сети. Если у Вас система уже установлена, и Вы ее собираетесь переустановить, то необходимо сконфигурировать систему так, как это сделано сейчас. В этом случае Вы можете и другими способами собрать интересующую вас информацию. Например: **К - Настройка - Центр информации**. Выбирая в левом окне интересующий нас параметр, справа Вы увидите информацию по нему.

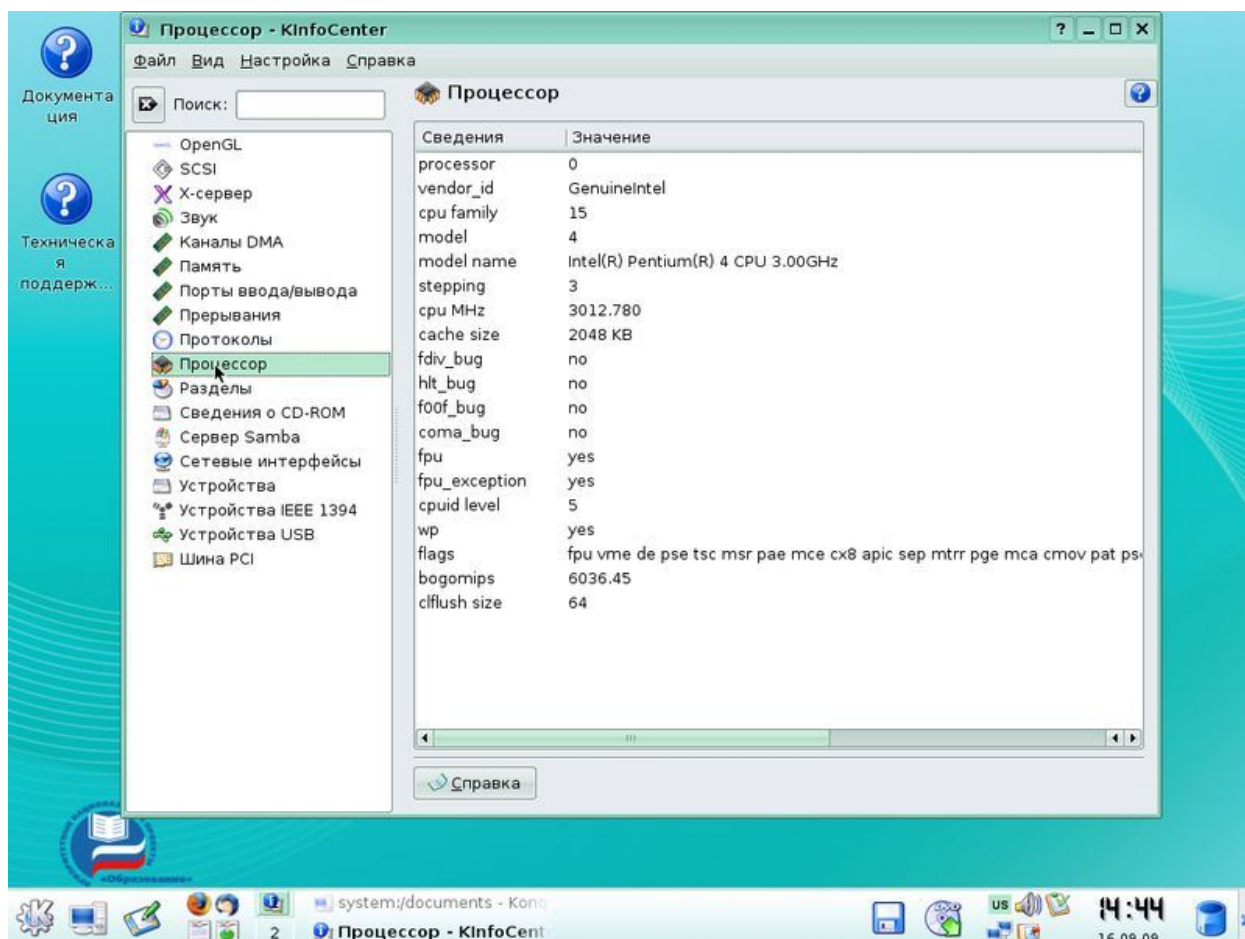


Рис.27. KinfoCenter – получение информации о системе

Тут можно получить полезную информацию о количестве и типе процессоров, общему количеству памяти в системе и степени ее текущего использования. Увидеть список устройств, использующихся в системе.

Еще один, классический способ получения информации о системе, состоит в использовании системной утилиты **uname**, которую можно вызвать из командной строки.

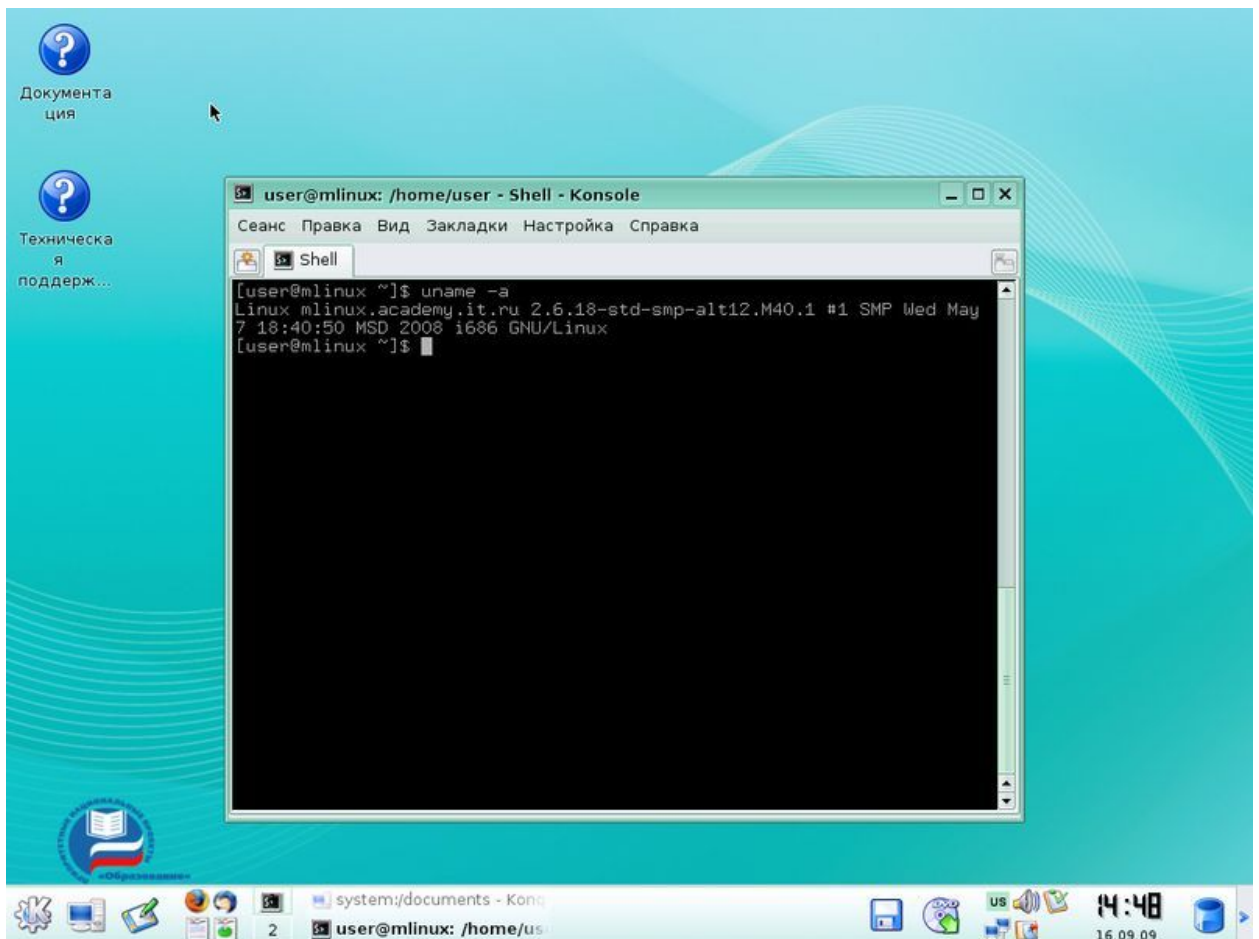


Рис.28. Команда `uname -a`

Полную информацию о системе можно получить с ключом **a**, т.е. набрать команду **uname -a**.

Вы увидите имя ядра, имя машины в сети, номер версии операционной системы, тип процессора, тип аппаратной платформы, имя операционной системы.

## 3. Инсталляция Linux

### 3.1. Введение

В этой главе мы рассмотрим процедуру установки ОС Линукс из пакета ПСПО на персональный компьютер – достаточно простой процесс, если отнестись к нему внимательно и аккуратно все спланировать. Вам необходимо правильно выбрать дистрибутив из комплекта поставки ПСПО в соответствии с аппаратными возможностями компьютера и поставленными задачами. В поставке ПСПО могут быть различные варианты дистрибутивных дисков, каждый из которых имеет особенности и предназначен для установки на различные по мощности компьютеры. Итак, первый шаг в процедуре установки – исходя из поставленных задач определиться с параметрами компьютера (тактовая частота процессора, количество оперативной памяти, объем дисковой памяти и т.д.), и выбрать для этого набора аппаратных характеристик подходящий дистрибутив

В этой главе мы рассмотрим процедуру установки ОС Linux из пакета ПСПО на персональный компьютер – достаточно простой процесс, если отнестись к нему внимательно и аккуратно все спланировать. Вам необходимо правильно выбрать дистрибутив из комплекта поставки ПСПО в соответствии с аппаратными возможностями компьютера и поставленными задачами. В поставке ПСПО могут быть различные варианты дистрибутивных дисков, каждый из которых имеет особенности и предназначен для установки на различные по мощности компьютеры. Итак, первый шаг в процедуре установки – исходя из поставленных задач определиться с параметрами компьютера (тактовая частота процессора, количество оперативной памяти, объем дисковой памяти и т.д.), и выбрать для этого набора аппаратных характеристик подходящий дистрибутив.

## **3.2. Выбор программного обеспечения для начальной установки**

### **Легкий Линукс**

Наиболее оптимален для использования на компьютерах, имеющих от 128 до 256 МБ оперативной памяти и процессор от P1 233 МГц. Располагается на двух CD-дисках. Этот вариант дистрибутива имеет самые низкие аппаратные требования, обеспечивает приемлемую производительность даже на системах с объёмом памяти 128 МБ.

Включает в себя графическую среду XFCE, в состав этого дистрибутива входят лёгкий и быстрый текстовый редактор Abiword и программа редактирования электронных таблиц Gnumeric.

### **Линукс Юниор**

Предназначен для использования на компьютерах с объемом оперативной памяти от 256 МБ до 1 ГБ. Располагается на двух CD-дисках. Этот вариант дистрибутива способен работать с достаточно высокой производительностью, в то же время не требует процессоров последнего поколения и большой оперативной памяти. Основан на графической среде KDE, помимо офисных приложений и обучающих программ включает в себя средства разработки программ Lazarus и KDevelop.

### **Линукс Мастер**

Дистрибутив на располагается на DVD, требует для эффективной работы объем оперативной памяти от 1ГБ. Этот дистрибутив можно использовать как для обучения компьютерной грамотности, так и для разработки полноценных программ на Java, C/C++ и PHP, благодаря среде разработки Eclipse, входящей в его состав. Кроме Eclipse, в Линукс Мастер входят все те программы, которые есть в варианте Линукс Юниор.

### **Линукс Терминал**

Предназначен для организации процесса обучения в компьютерном классе, возможно не располагающем мощными и современными компьютерами. В таком классе

один компьютер должен быть достаточно мощным для организации одновременной работы на нем нескольких учеников в терминальном режиме (сервер класса), все остальные компьютеры могут иметь низкие параметры по быстродействию, объему оперативной и дисковой памяти, т.е. например вполне могут быть использованы устаревшие компьютеры. Данная схема называется "сервер — тонкие клиенты". Иными словами, вся вычислительная мощность сосредоточена в одном компьютере класса – сервере, все остальные компьютеры просто обеспечивают доступ по сети с рабочего места ученика к сессии на стороне сервера.

Аппаратные требования к серверу: объём оперативной памяти от 512Мб, два Ethernet-адаптера и устройство чтения DVD-дисков. Аппаратные требования к клиентским компьютерам: объём оперативной памяти от 32Мб, Ethernet-адаптер с поддержкой сетевой загрузки (PXE). Дистрибутив расположен на DVD.

Наш курс ориентирован в первую очередь на Линукс Мастер, поэтому некоторые рекомендации, приведенные в данном учебном пособии, для других дистрибутивов Linux могут незначительно отличаться.

### **3.3. Установка дистрибутива**

Прежде всего, следует настроить компьютер на загрузку с устройства CD-ROM (или DVD-ROM). Перезагрузите компьютер, зайдите в режим настройки BIOS (на большинстве компьютеров для этого следует нажать либо <Del>, либо <F2>), затем выберите в BIOS в качестве первого загрузочного устройства привод лазерных дисков.

Подготовьте необходимое пространство на жестком диске. Требования к этому параметру следующие:

- Лёгкий Линукс — не менее 7 Гб;
- Линукс Юниор — не менее 7 Гб;
- Линукс Мастер — не менее 7 Гб;

Если устанавливаемый вариант Linux будет единственной операционной системой на этом компьютере, можно ничего не предпринимать – вся подготовка пройдет автоматически в процессе установки. Другое дело, если помимо Linux на данном компьютере должна присутствовать другая ОС, скажем Microsoft Windows какой-либо версии. В этом случае каждой ОС следует выделить свой собственный, отдельный раздел на диске. Желательно установить сначала Windows, заняв при этом не все место на диске, а оставив часть для последующей установки Linux. Дело в том, что процесс установки Linux позволяет впоследствии корректно осуществить выбор (через меню при старте компьютера), какую именно ОС загрузит, а процесс установки Windows игнорирует наличие загрузчиков других ОС и для исправления этого придется заново установить

загрузчик Linux. Таким образом, правильным выбором будет установить Linux в последнюю очередь, после установки других ОС.

Установите в привод CD-ROM дистрибутивный диск и перезагрузите компьютер. Вы увидите на экране сообщения о загрузке Линукс, после чего на экране появится основное меню процесса установки.

В загрузочном меню выберите пункт **Установка** и нажмите **Enter**. Через некоторое время загрузится графический интерфейс программы установки. О значении каждого пункта меню и о возможных параметрах установки можно узнать из Справки, доступной по нажатию клавиши **F1**.

Выполните последовательно шаги установки. В большинстве случаев процесс установки очень прост и сводится к проверке правильности предлагаемых параметров и нажатию кнопки **Далее** для перехода к следующему шагу. Для получения информации о каждом шаге воспользуйтесь справкой, нажав **F1**. Более подробно о процессе установки можно прочитать в **Руководстве по установке**, помещенном на установочном диске в каталоге **docs**.

По завершении установки перезагрузите компьютер и загрузите установленную операционную систему. Не забудьте извлечь установочный диск из привода и настроить BIOS на загрузку компьютера с жесткого диска.

### **3.4. Точки монтирования и их использование**

Если до этого вы не использовали Linux или Linux-подобные операционные системы, понятие точки монтирования на первый взгляд может показаться странным. Однако, это один из самых мощных и гибких методов управления файловыми системами. Во многих других операционных системах полный путь к файлу включает имя файла, некоторое указание каталога, в котором располагается файл, и идентификатор физического устройства, на котором находится файл.

В Linux как и в других операционных системах, полный путь к файлу включает название файла и указание каталога, в котором он находится. Однако явное указание устройства отсутствует.

И причиной этому кажущемуся недостатку является точка монтирования. В других операционных системах в каждом разделе есть своя структура каталогов. Однако в Linux-подобных системах, во всей системе есть только одна древовидная структура каталогов, и эта структура может занимать несколько разделов. Ключом здесь является точка монтирования. Когда файловая система монтируется, она становится доступной в виде набора подкаталогов указанной точки монтирования.

Этот кажущийся недостаток на самом деле является преимуществом. Это значит, что возможно гибкое расширение файловой системы Linux, так как каждый каталог может стать точкой подключения дополнительного дискового пространства.

Предположим, что в корневом каталоге системы Linux содержится каталог **foo**, полный путь к которому будет **/foo/**. Затем предположим, что есть также раздел, который будет монтироваться, а точкой монтирования раздела выбран каталог **/foo/**. Если в корневом каталоге этого раздела находится файл с именем **bar.txt**, после того, как раздел будет смонтирован, вы сможете обратиться к этому файлу, указав следующий путь:

**/foo/bar.txt**

Другими словами, после того как раздел был смонтирован, при обращении к файлам в каталоге **/foo/** будет происходить обращение к файлам в этом разделе.

Во многих системах Linux часто используется точка монтирования **/home/**, потому что в **/home/** обычно размещаются домашние каталоги всех пользователей. Если **/home/** используется как точка монтирования, все файлы пользователей хранятся в выделенном разделе, и никогда не займут раздел операционной системы.

В UNIX существует только один корневой каталог, а все остальные файлы и каталоги вложены в него. Чтобы получить доступ к файлам и каталогам на каком-нибудь диске, необходимо монтировать этот диск командой **mount**. Например, чтобы открыть файлы на CD, нужно, говоря простым языком, сказать операционной системе: возьми файловую систему на этом компакт-диске и покажи её в каталоге **/mnt/cdrom**. Все файлы и каталоги, находящиеся на CD, появятся в этом каталоге **/mnt/cdrom**, который называется точкой монтирования (англ. mount point). В большинстве UNIX-подобных систем съёмные диски (дискеты и CD), флеш-накопители и другие внешние устройства хранения данных монтируют в каталог **/mnt**, **/mount** или **/media**. Unix и UNIX-подобные операционные системы также позволяют автоматически монтировать диски при загрузке операционной системы.

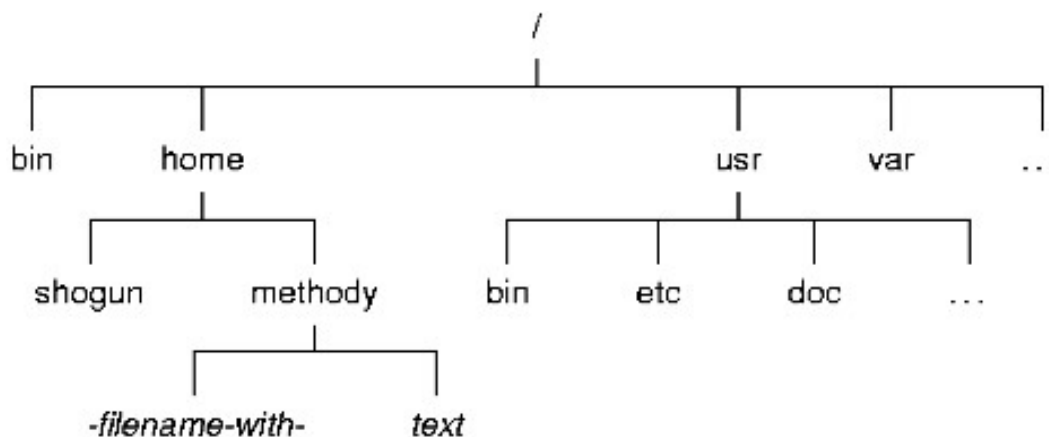


Рис. 1. Структура папок



### 3.5. Файловые системы

*Файловая система* (англ. file system) — регламент, определяющий способ организации, хранения и именования данных на носителях информации. Она определяет формат физического хранения информации, которую принято группировать в виде файлов. Конкретная файловая система определяет размер имени файла, максимальный возможный размер файла, набор атрибутов файла. Некоторые файловые системы предоставляют сервисные возможности, например, разграничение доступа или шифрование файлов.

Файловая система связывает носитель информации, с одной стороны, и API для доступа к файлам — с другой. Когда прикладная программа обращается к файлу, она не имеет никакого представления о том, каким образом расположена информация в конкретном файле, так же, как и на каком физическом типе носителя (CD, жёстком диске, магнитной ленте или блоке флеш-памяти) он записан. Всё, что знает программа — это имя файла, его размер и атрибуты. Эти данные она получает от драйвера файловой системы. Именно файловая система устанавливает, где и как будет записан файл на физическом носителе (например, жёстком диске).

С точки зрения операционной системы, весь диск представляет собой набор кластеров размером от 512 байт и выше. Драйверы файловой системы организуют кластеры в файлы и каталоги (реально являющиеся файлами, содержащими список файлов в этом каталоге). Эти же драйверы отслеживают, какие из кластеров в настоящее время используются, какие свободны, какие помечены как неисправные.

Однако файловая система не обязательно напрямую связана с физическим носителем информации. Существуют виртуальные и сетевые файловые системы, которые являются лишь способом доступа к файлам, находящимся на удалённом компьютере.

Дисковые файловые системы обычно являются поток-ориентированными. Файлы в поток-ориентированных файловых системах представляются последовательностью битов, часто предоставляющие такие функции, как чтение, запись, изменение данных и случайный доступ.

Перечислим самые популярные файловые системы, используемые в ПСПО:

- *Ext2 или 2я расширенная файловая система* — файловая система для ядра Linux. Она была разработана Rémy Card'ом в качестве замены для extended file system. Она достаточно быстра для того, что бы служить эталоном в тестах производительности файловых систем. Она не является журналируемой файловой системой и это её главный недостаток. Развитием ext2 стала журналируемая файловая система ext3, полностью совместимая с ext2.

- *Ext3 или 3-я расширенная файловая система* — журналируемая файловая система, используемая в операционных системах на ядре Linux, является файловой системой по умолчанию во многих дистрибутивах. Основана на ФС ext2.

Основное отличие от ext2fs состоит в том, что ext3 журналируема, то есть в ней предусмотрена запись некоторых данных, позволяющих восстановить файловую систему при сбоях в работе компьютера.

- *ReiserFS* — журналируемая файловая система, разработанная специально для Linux компанией Namesys под руководством Ганса Рейзера (Hans Reiser). Обычно под словом ReiserFS понимают третью версию (последняя — 3.6.19), а четвёртую называют Reiser4.
- *Network File System (NFS)* — протокол сетевого доступа к файловым системам, первоначально разработан Sun Microsystems в 1984 году. Основан на протоколе вызова удалённых процедур (ONC RPC, Open Network Computing Remote Procedure Call). Позволяет подключать (монтировать) удалённые файловые системы через сеть.

### 3.6. **Дополнительная настройка, конфигурирование и проверка оборудования в процессе установки ОС**

#### 3.6.1. Начало установки: загрузка системы

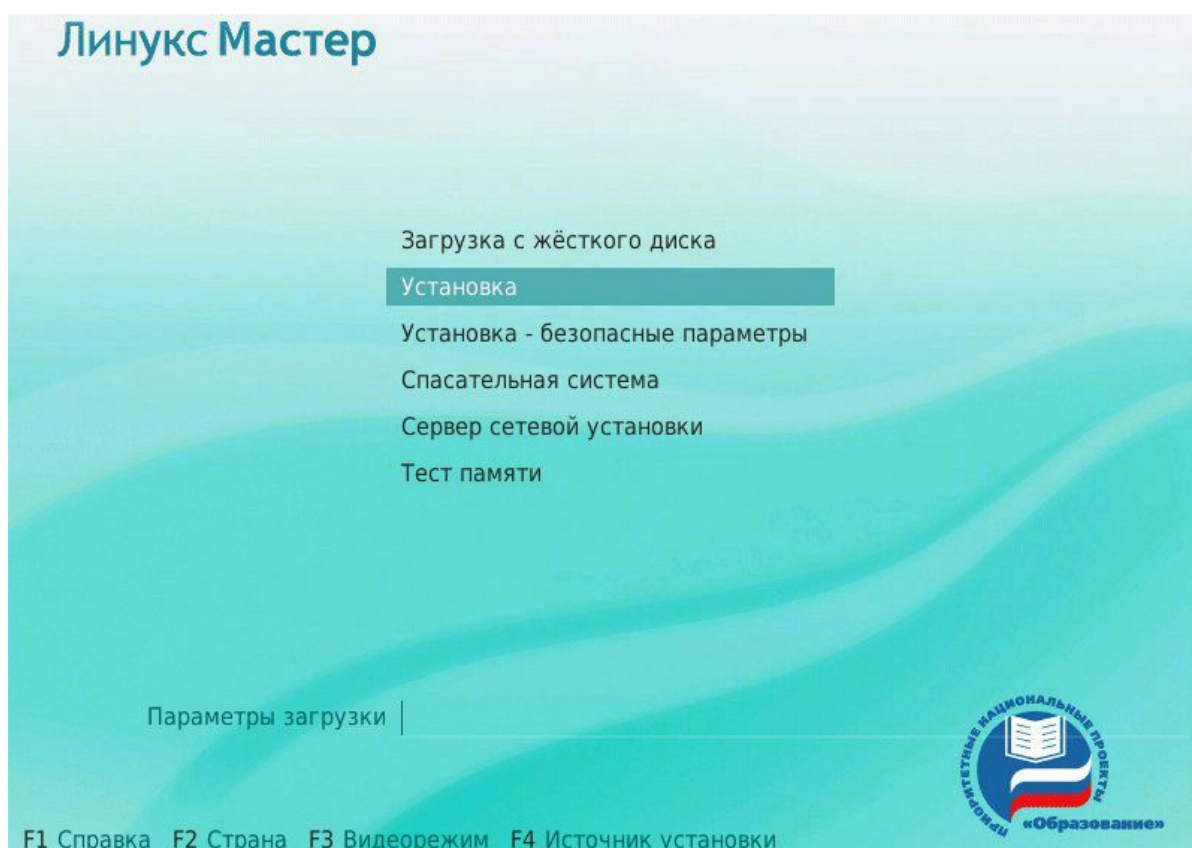


Рис. 2. Начало установки

Загрузка с установочного диска начинается с меню, в котором перечислено несколько вариантов загрузки, причём установка системы — это только одна из возможностей. Меню позволяет запустить программу для восстановления системы или проверки памяти. Мышь на этом этапе установки не поддерживается, поэтому для выбора различных вариантов и опций установки воспользуйтесь клавиатурой. Можно получить справку по любому пункту меню, выбрав этот пункт и нажав **F1**. Кроме установки с лазерного диска доступно несколько вариантов сетевой установки и установка с жёсткого диска.

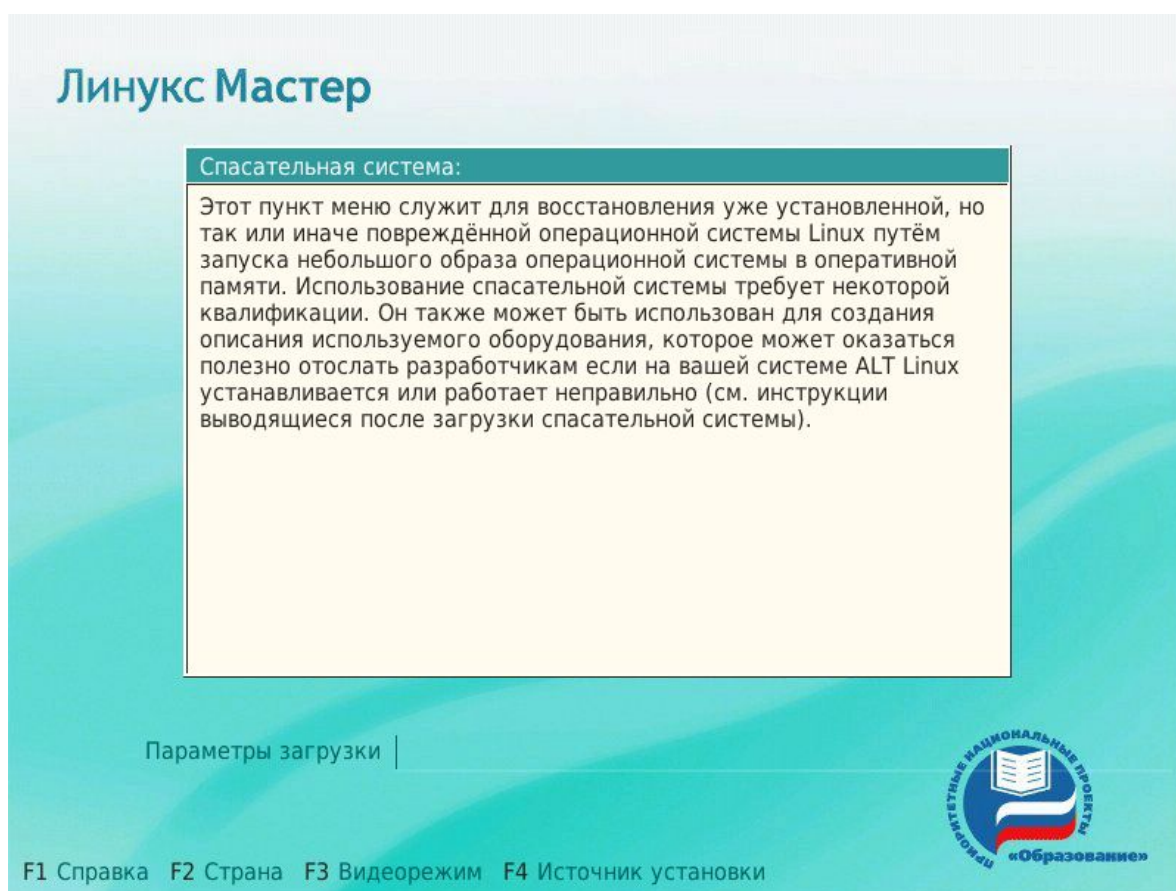


Рис. 3. Справка

Нажатием **F2** осуществляется выбор страны. От выбора страны в загрузчике зависит язык интерфейса загрузчика и программы установки а также какие языки будут доступны в списке языков установки — кроме основного для выбранной страны языка, в список будут включены и другие языки данной территории. По умолчанию предлагается **Россия**. Если выбрать вариант **Прочие**, то в списке языков установки будут перечислены все возможные языки.

По нажатию **F3** открывается меню доступных видеорежимов (разрешений экрана). Это разрешение будет использоваться во время установки и загрузки установленной системы.

Чтобы начать процесс установки, нужно клавишами перемещения курсора вверх, вниз выбрать пункт меню **Установка** и нажать **Enter**. В начальном загрузчике установлено небольшое время ожидания: если в этот момент не предпринимать никаких действий, то будет загружена та система, которая уже установлена на жёстком диске. Если вы пропустили нужный момент, перезагрузите компьютер и вовремя выберите пункт **Установка**.

Начальный этап установки не требует вмешательства пользователя: происходит автоматическое определение оборудования и запуск компонентов программы установки. Сообщения о том, что происходит на этом этапе, можно просмотреть, нажав клавишу **ESC**.

### **3.6.2. Другие источники установки**

#### **Сетевая установка**

Установка Линукс Мастер возможна не только с лазерного диска, её можно производить и по сети. Обязательное условие для этого — наличие на сервере дерева файлов, аналогичного содержимому установочного диска, и внешний носитель с начальным загрузчиком. Таким носителем может быть как сам загрузочный лазерный диск, так и, например, flash-накопитель, который можно сделать загрузочным, воспользовавшись утилитой **mkbootflash**.

Если вы желаете произвести установку по сети, загрузившись с лазерного диска, то сразу переходите к разделу **Запуск сетевой установки**. Для выполнения этого варианта установки необходим специальный компьютер – сервер установки, а также носитель (CD-ROM или флэш-накопитель) с которого будет производиться начальная загрузка системы. Описание подготовки сервера установки не является предметом рассмотрения в данном курсе, поэтому рекомендуем обратиться к документации проекта ПСПО.

#### **Запуск сетевой установки**

Кнопка **F4** позволяет выбрать источник сетевой установки: FTP, HTTP или NFS-сервер. Нужно указать имя или IP-адрес сервера и каталог (начиная с /), в котором размещён дистрибутив Линукс Мастер. В случае установки по протоколу FTP может понадобиться ввести имя пользователя и пароль.

Для получения подобного дерева каталогов на стороне сервера достаточно скопировать содержимое установочного лазерного диска в один из подкаталогов FTP-сервера (либо HTTP или NFS-сервера). При сетевой установке со стороны клиента (компьютера, на который производится установка) может понадобиться определить параметры соединения с сервером. В этом случае на экране будут появляться диалоги, например, с предложением выбрать сетевую карту (если их несколько) или указать тип IP-адреса: статический (потребуется вписать его самостоятельно) или динамический (DHCP).

После успешного соединения с сервером в память компьютера будет загружен образ установочного диска, после чего начнётся установка системы так же, как и при установке с лазерного диска.

### Установка с жёсткого диска

Аналогично установке по сети можно установить Линукс Мастер с жёсткого диска. Для этого понадобится подключить дополнительный жёсткий диск с дистрибутивом, чтобы выбрать подключенный диск в качестве источника установки, нужно в строке **Параметры загрузки** меню начального загрузчика указать метод установки, набрав: `automatic=method:disk`, где *automatic* — параметр, определяющий ход начальной стадии установки. По нажатию **Enter** через некоторое время на экране появится диалог выбора дискового раздела, а после — выбора пути к каталогу с дистрибутивом. После указания пути начнётся установка системы. При желании можно сразу указать путь к дистрибутиву, сделав в строке параметров загрузки запись вида: `automatic=method:disk,disk:hdb,partition:hdbX,directory:<путь_к_каталогу_с_дистрибутивом>`.

### Последовательность установки

До того, как будет произведена установка базовой системы на жёсткий диск, программа установки работает с образом системы, загруженным в оперативной памяти компьютера.

Если инициализация оборудования завершилась успешно, будет запущен графический интерфейс программы-установщика. Процесс установки состоит из шагов, каждый из которых посвящён настройке или установке определённого параметра системы. Шаги выполняются последовательно, переход к следующему шагу происходит по нажатию кнопки **Далее**. При помощи кнопки **Назад** при необходимости можно вернуться к уже пройденному шагу и внести изменения. Однако на этом этапе установки возможность перехода к предыдущему шагу ограничена теми шагами, где нет зависимости от данных, введённых ранее.

Если по каким-то причинам возникла необходимость прекратить установку, нажмите **Reset** на системном блоке компьютера. Помните, что совершенно *безопасно* прекращать установку только до шага **Подготовка диска**, поскольку до этого момента не производится никаких изменений на жёстком диске. Если прервать установку между шагами **Подготовка диска** и **Установка загрузчика**, вероятно, что после этого с жёсткого диска не сможет загрузиться ни одна из установленных систем.

Технические сведения о ходе установки можно посмотреть, нажав **Ctrl+Alt+F1**, вернуться к программе установки — **Ctrl+Alt+F7**. По нажатию **Ctrl+Alt+F2** откроется отладочная виртуальная консоль.

Каждый шаг сопровождается краткой справкой, которую можно вызвать, нажав **F1**.

*Во время установки системы выполняются следующие шаги:*

- Язык
- Уведомление о правах
- Настройка клавиатуры
- Часовой пояс
- Дата и время
- Подготовка диска
- Установка базовой системы
- Сохранение настроек
- Дополнительные диски
- Дополнительные пакеты
- Установка загрузчика
- Администратор системы
- Системный пользователь
- Настройка сети
- Настройка графической системы
- Завершение установки

## **Язык**

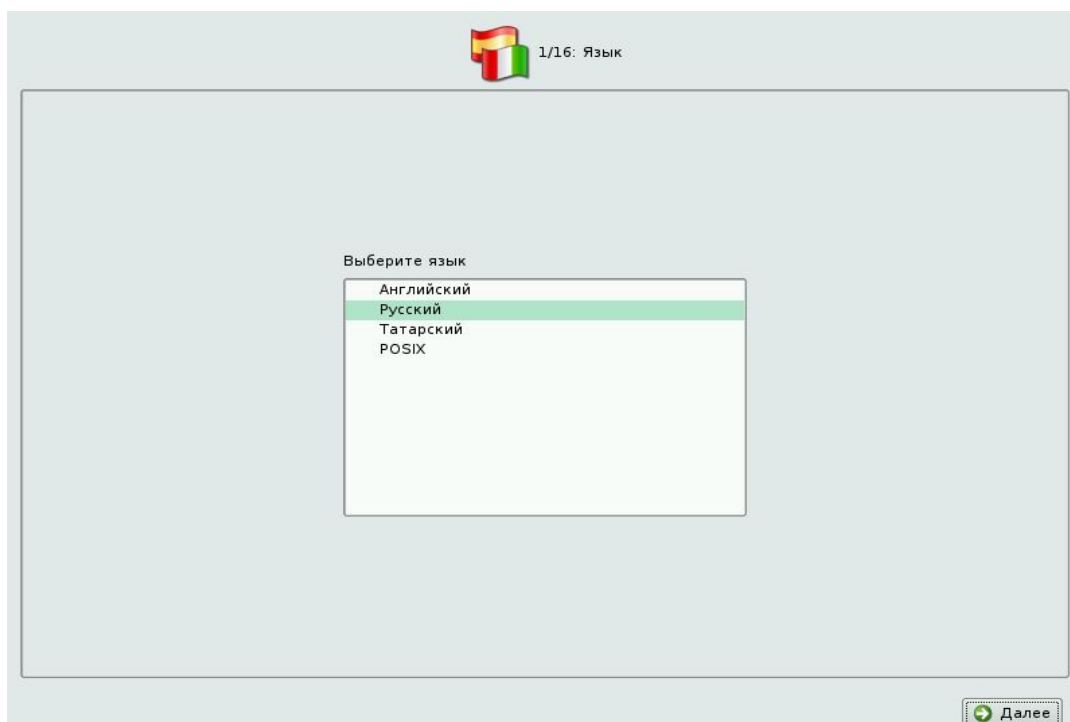


Рис. 4. Язык

Установка начинается с выбора основного языка — языка интерфейса программы установки и устанавливаемой системы. В списке, помимо доступных языков региона (выбранного на этапе начальной загрузки), указан и английский язык.

### Уведомление о правах

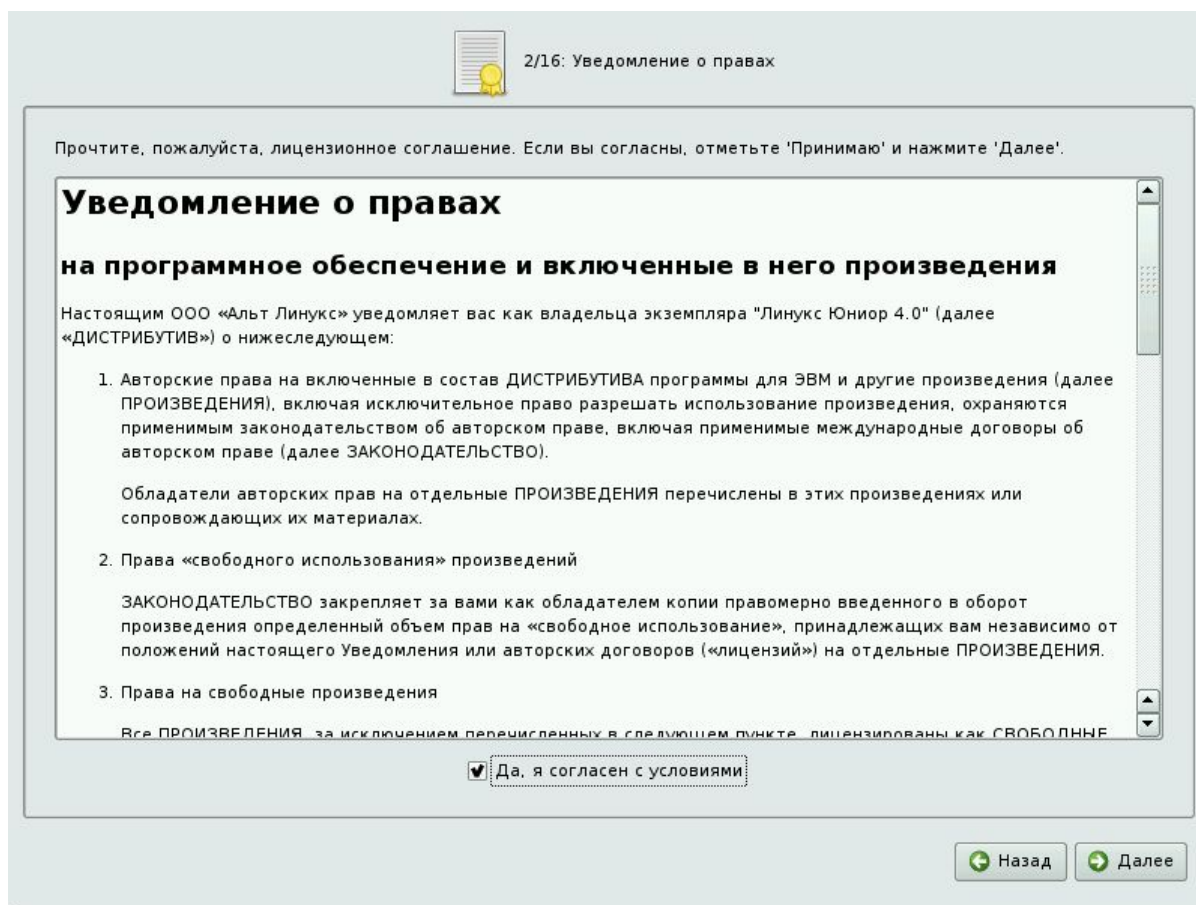


Рис. 5. Уведомление о правах

Перед продолжением установки следует внимательно прочитать условия лицензии. В лицензии говорится о ваших правах.

*В частности, за вами закрепляются права на:*

- эксплуатацию программ на любом количестве компьютеров и в любых целях;
- распространение программ (сопровождая их копией авторского договора);
- получение исходных текстов программ.

Лицензия относится ко всему дистрибутиву Линукс Мастер. Если вы согласны с условиями лицензии, отметьте пункт Да, я согласен с условиями и нажмите Далее.

## Настройка клавиатуры

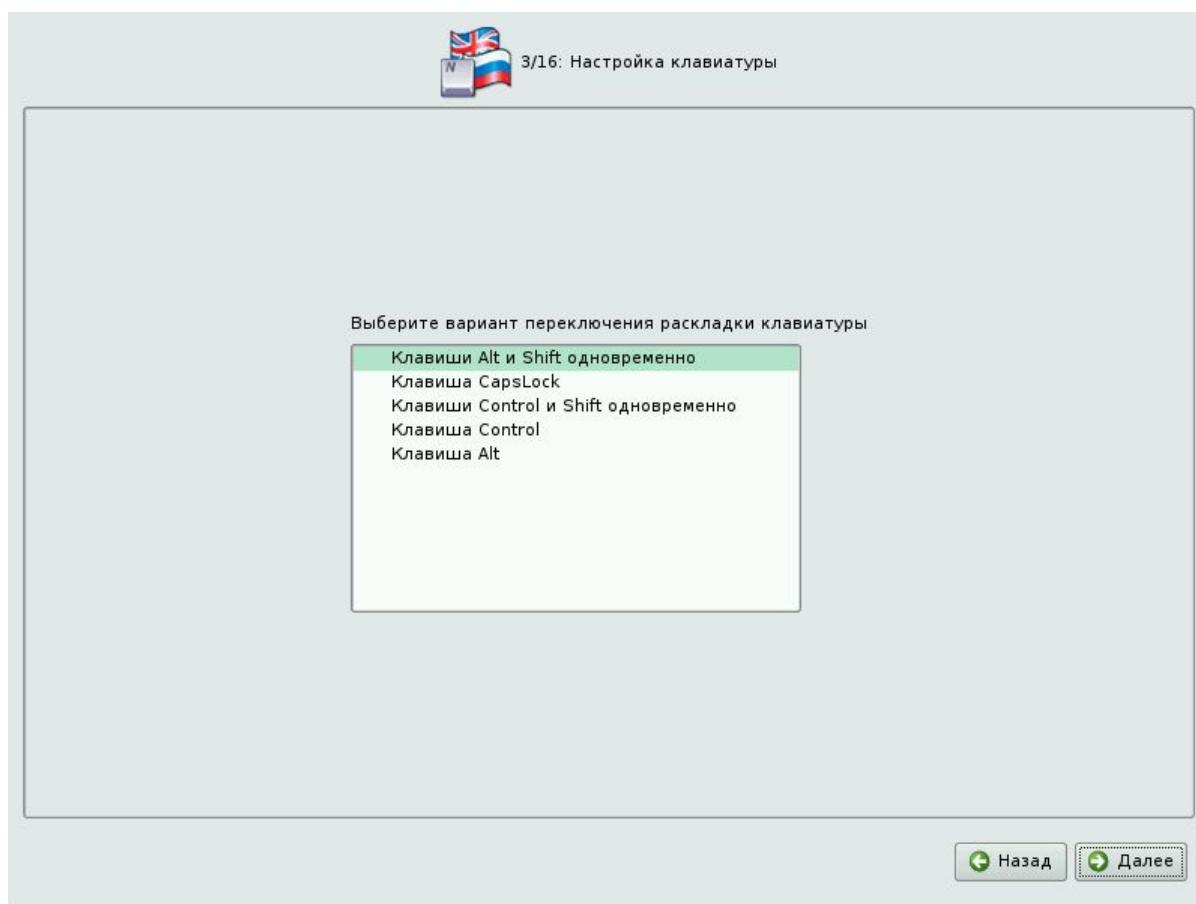


Рис. 6. Настройка клавиатуры

Раскладка клавиатуры — это привязка букв, цифр и специальных символов к клавишам на клавиатуре. Помимо ввода символов на основном языке, в любой системе Linux необходимо иметь возможность вводить латинские символы (имена команд, файлов и т. п.), для чего обычно используется стандартная английская раскладка клавиатуры. Переключение между раскладками осуществляется при помощи специально зарезервированных для этого клавиш.

Для русского языка доступны следующие комбинации клавиш:

- Клавиши Alt + Shift .
- Клавиша Capslock
- Клавиши Control + Shift
- Клавиша Control
- Клавиша Alt



## Часовой пояс

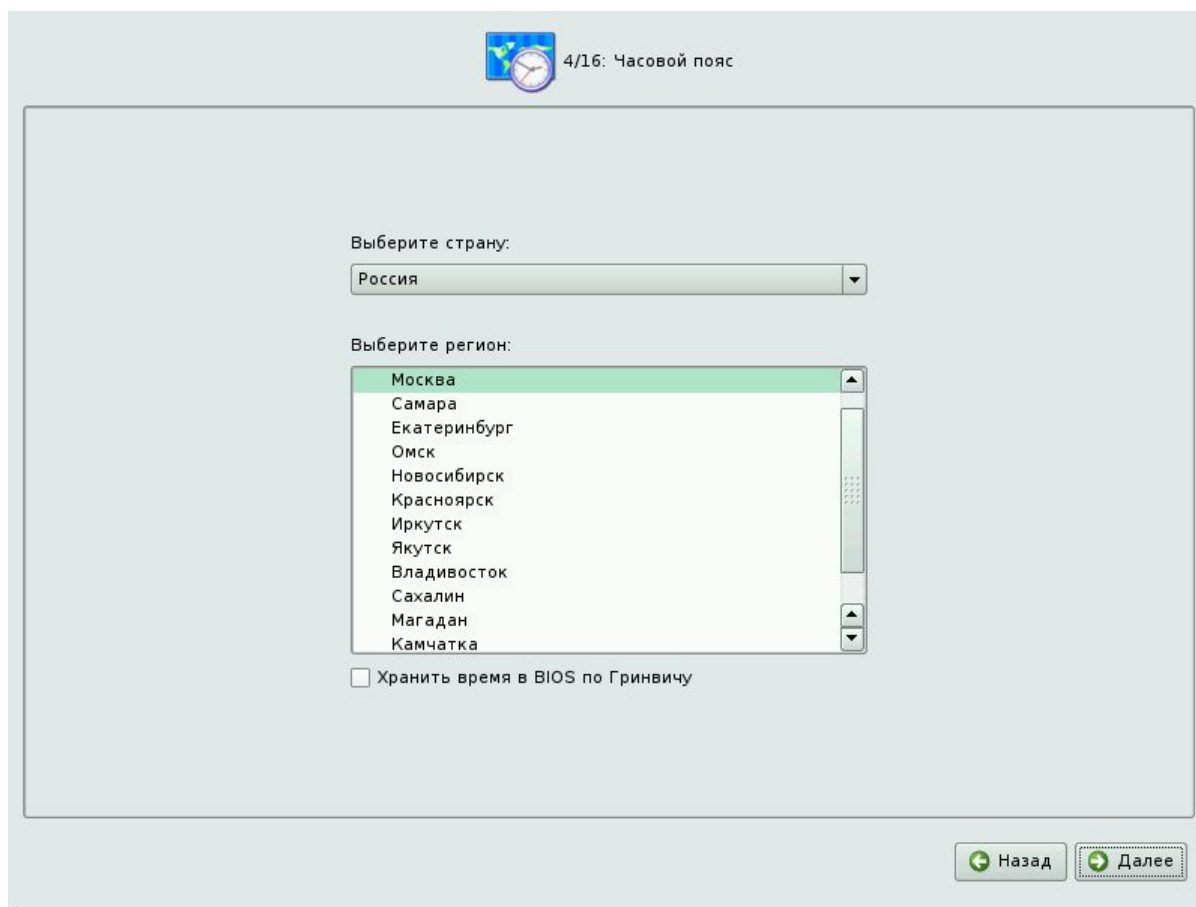


Рис. 7. Часовой пояс

Для корректной установки даты и времени достаточно правильно указать часовой пояс и выставить желаемые значения для даты и времени.

На этом шаге следует выбрать часовой пояс, по которому нужно установить часы. Для этого в соответствующих списках выберите страну, а затем регион. Поиск по списку можно ускорить, набирая на клавиатуре первые буквы искомого слова.

Обратите внимание на отметку **Хранить время в BIOS по Гринвичу**. В системных часах BIOS желательно устанавливать не локальное, а универсальное время по Гринвичу (GMT). При этом программные часы будут показывать локальное время в соответствии с выбранным часовым поясом, и системе не потребуется изменять настройки BIOS при сезонном переводе часов и смене часового пояса. Однако если вы планируете на этом же компьютере использовать другие операционные системы, отметку нужно снять, иначе при загрузке в другую операционную систему время может сбиваться.

## Дата и время

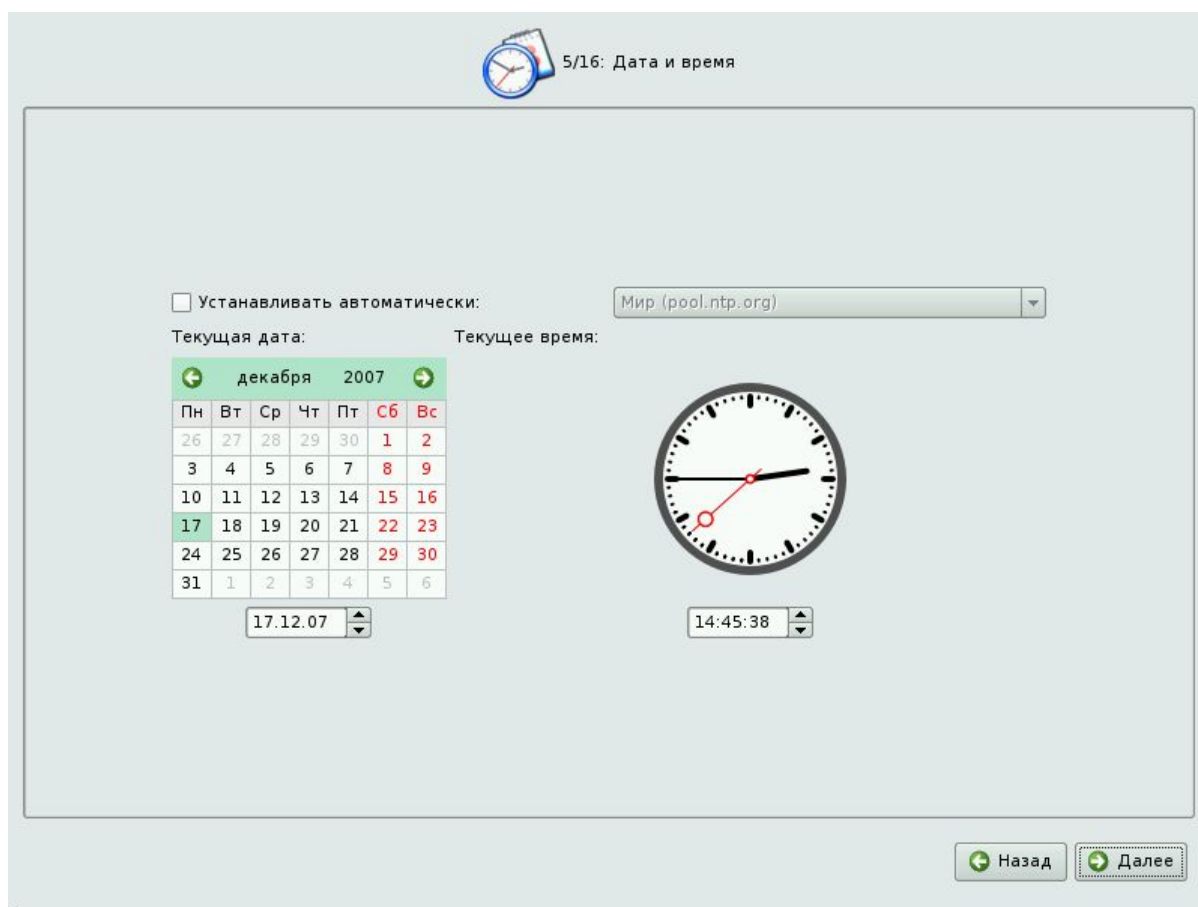


Рис. 8. Дата и время

Программа установки полагает, что системные часы (BIOS) отображают локальное время. Поэтому часы на этом шаге показывают либо время, соответствующее вашим системным часам, либо, если в предыдущем шаге была выставлена отметка **Хранить время в BIOS по Гринвичу**, время, соответствующее GMT, с учётом вашего часового пояса. Это значит, что, если системные часы отображают локальное время, а вы всё же выставили отметку **Хранить время в BIOS по Гринвичу** в предыдущем шаге, то часы будут отображать неверное время.

Проверьте, верно ли отображаются дата и время, и, при необходимости, выставьте правильные значения.

Если ваш компьютер подключён к локальной сети или к Интернет, можно включить синхронизацию системных часов (NTP) с удалённым сервером, для этого достаточно отметить пункт **Устанавливать автоматически** и выбрать из списка NTP-сервер.

## Подготовка диска

Переход к этому шагу может занять некоторое время. Время ожидания может быть разным и зависит от производительности компьютера, объёма жёсткого диска, количества разделов на нем и т. д.

На этом этапе подготавливается площадка для установки Линукс Мастер, в первую очередь — выделяется свободное место на диске. Для установки с выбором одного из автоматических профилей разметки потребуется не менее 7 Гб на одном или нескольких жёстких дисках компьютера. Программа установки автоматически создаёт три раздела: для корневой файловой системы (/) — 4 Гб, для области подкачки (swap) — 1 Гб и для пользовательских данных (/home) — оставшееся дисковое пространство. При подготовке разделов вручную необходимо выделить как минимум 4 Гб (3 Гб для корневого раздела и 1 Мб для swap). Рекомендуемое значение — 5 Гб без учёта домашнего раздела. Для компьютера учителя также рекомендуется дополнительно выделить 4 Гб для организации FTP, HTTP или NFS-сервера с образом диска, содержащего дистрибутив, для обеспечения возможности сетевой установки на другие компьютеры класса.

### Выбор профиля разбиения диска

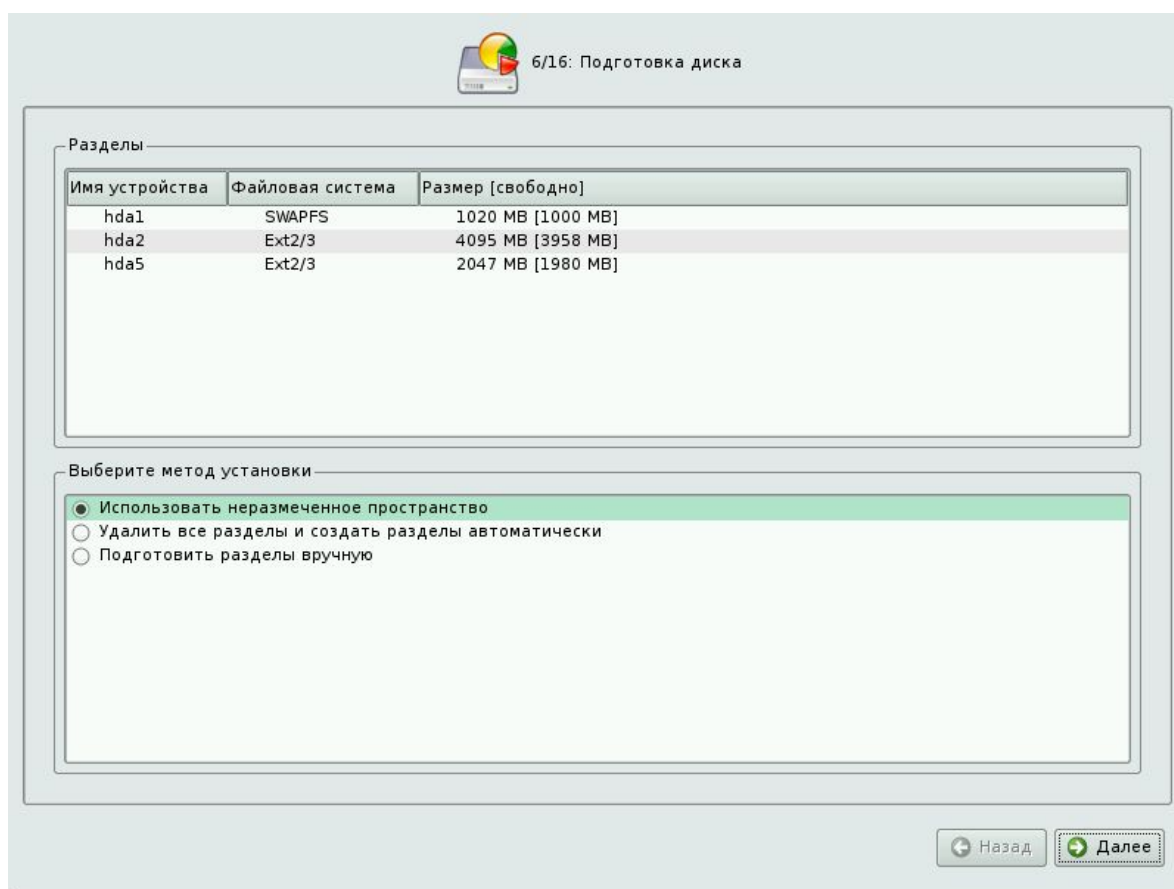


Рис. 8. Выбор профиля разбиения диска

В списке разделов перечислены уже существующие на жёстких дисках разделы (в том числе здесь могут оказаться съёмные USB-носители, подключённые к компьютеру в момент установки). Ниже перечислены доступные профили разбиения диска. Профиль — это шаблон распределения места на диске для установки Linux.

Можно выбрать один из трёх профилей:

- Использовать неразмеченное пространство
- Удалить все разделы и создать разделы автоматически
- Подготовить разделы вручную

Первые два профиля предполагают автоматическое разбиение диска. Они ориентированы на среднестатистические рабочие станции и должны подойти для большинства пользователей.

### **Автоматические профили разбиения диска**

Применение профилей автоматического разбиения происходит сразу по нажатию **Далее**, после чего непосредственно начинается этап установки базовой системы.

Если для применения одного из профилей автоматической разметки доступного места окажется недостаточно, то есть менее 7 Гб, будет выведено сообщение об ошибке:

**Невозможно применить профиль, недостаточно места на диске.**

Если данное сообщение появилось после попытки применить профиль **Использовать неразмеченное пространство**, то вы можете очистить место, удалив данные, которые уже есть на диске. Выберите пункт **Удалить все разделы и создать разделы автоматически**. При применении этого профиля сообщение о недостатке места связано с недостаточным объёмом всего жёсткого диска, на который производится установка. В этом случае необходимо воспользоваться режимом ручной разметки: профиль **Подготовить разделы вручную**.

Будьте осторожны при применении профиля **Удалить все разделы и создать разделы автоматически**! В этом случае будут удалены **все** данные со **всех** дисков без возможности восстановления. Рекомендуется использовать эту возможность только в том случае, если вы уверены, что диски не содержат **никаких ценных данных**.

### **Ручной профиль разбиения диска**

При необходимости освободить часть дискового пространства следует воспользоваться профилем разбиения вручную. Вы сможете удалить некоторые из существующих разделов или содержащиеся в них файловые системы. После этого можно создать необходимые разделы самостоятельно или вернуться к шагу выбора профиля и применить один из автоматических профилей.

Предположим, Вы решили подготовить диск самостоятельно, не используя автоматические профили. После выбора данной возможности, Вам будет предоставлена информация об имеющихся на компьютере дисках, разделах, логических томах и RAID-массивах различных конфигураций. В приведенном ниже примере рассматривается простейший случай – в наличии имеется один жесткий диск, на котором отсутствуют какие бы то ни было разделы (например, новый компьютер без операционной системы, или компьютер с новым диском).

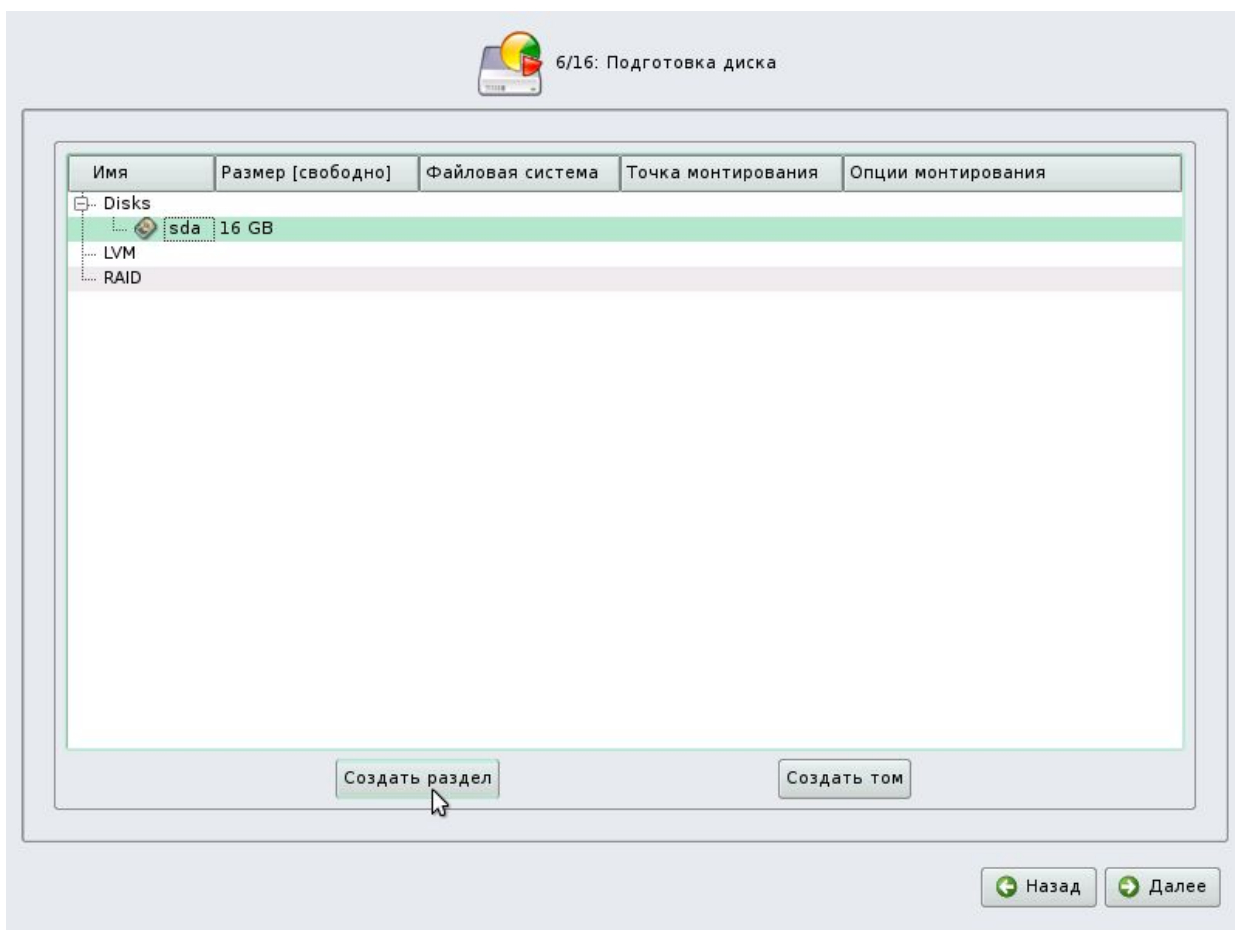


Рис. 9. Ручное разбиение диска

Итак, в списке дисков мы видим одно устройство – SDA (так обычно именуется диски с интерфейсом SATA или SCSI) емкостью 16Гб. Для успешной установки системы необходимо создать как минимум 2 раздела – один для файлов, второй – т.н. SWAP (своп) раздел, который будет использоваться системой для выгрузки из оперативной памяти отдельных процессов.

Выбираем опцию **Создать раздел**.

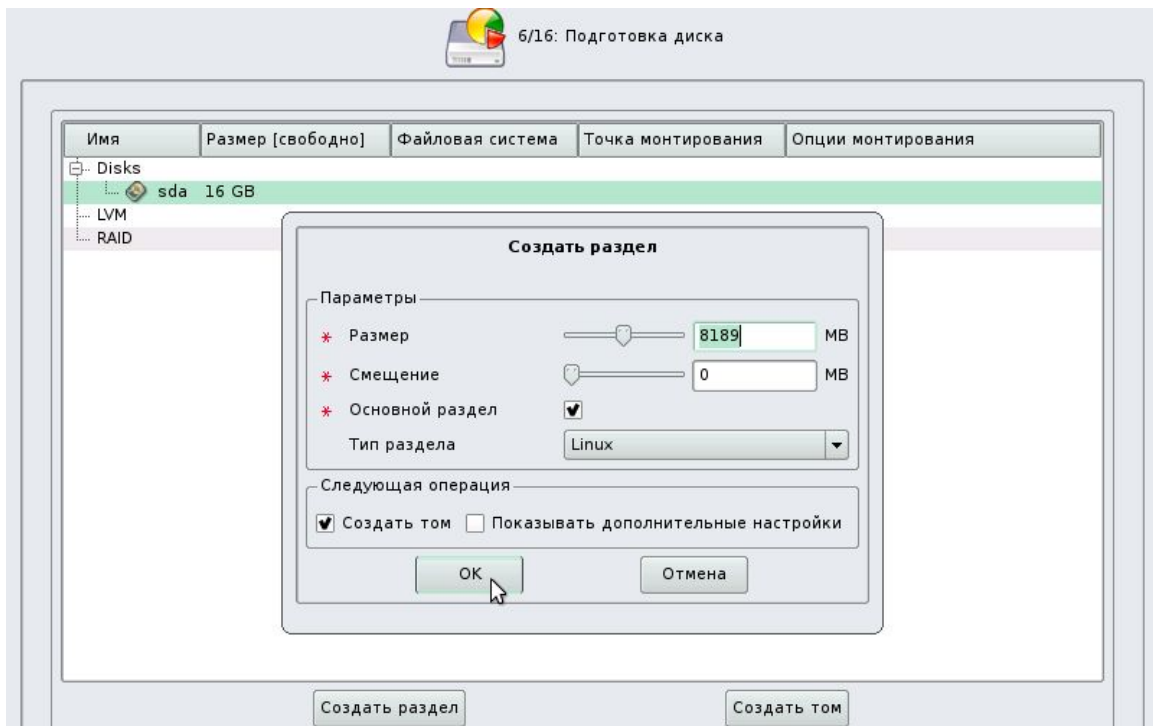


Рис. 10. Диалог создания раздела.

В открывшейся форме следует задать размер раздела (самое верхнее поле, в нашем случае – 8Гб), остальные параметры оставляем по умолчанию. Будет создан раздел, который займет половину нашего 16Гб диска, и этот раздел будет основным, предназначенным для хранения файлов операционной системы и данных пользователей. Далее потребуется выбрать для этого раздела тип файловой системы.

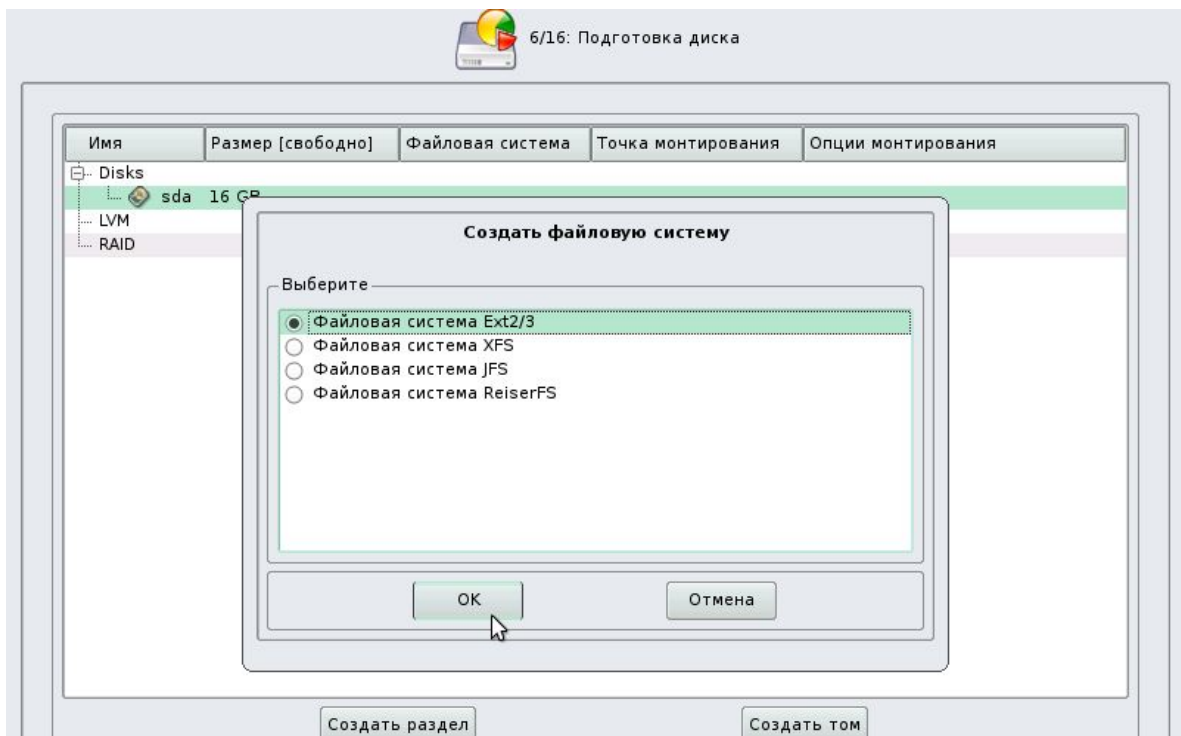


Рис. 11.. Диалог выбора файловой системы.

Рекомендуется оставить **Ext2/3**, т.е. значение по умолчанию.

После выбора типа ФС необходимо указать точку монтирования файловой системы нашего нового раздела.

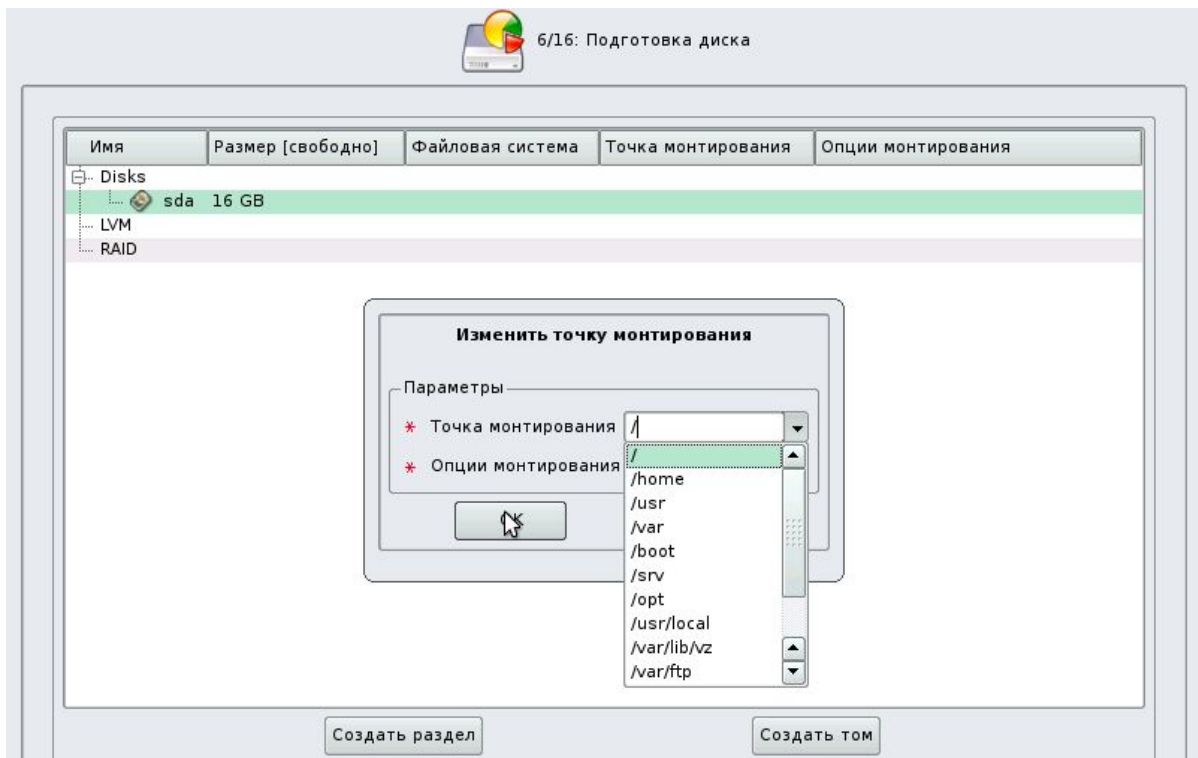


Рис. 12. Диалог выбора ФС.

Для нормальной работы процесса установки нам необходима точка монтирования /, остальные – опциональны. Выбираем / в выпадающем списке, нажимаем **Ок**.

После создания нового раздела мы возвращаемся в информационный экран, на котором уже виден вновь созданный раздел и остаток свободного пространства на диске. Обратите внимание – первый раздел на **SDA** диске получил имя **SDA1**.

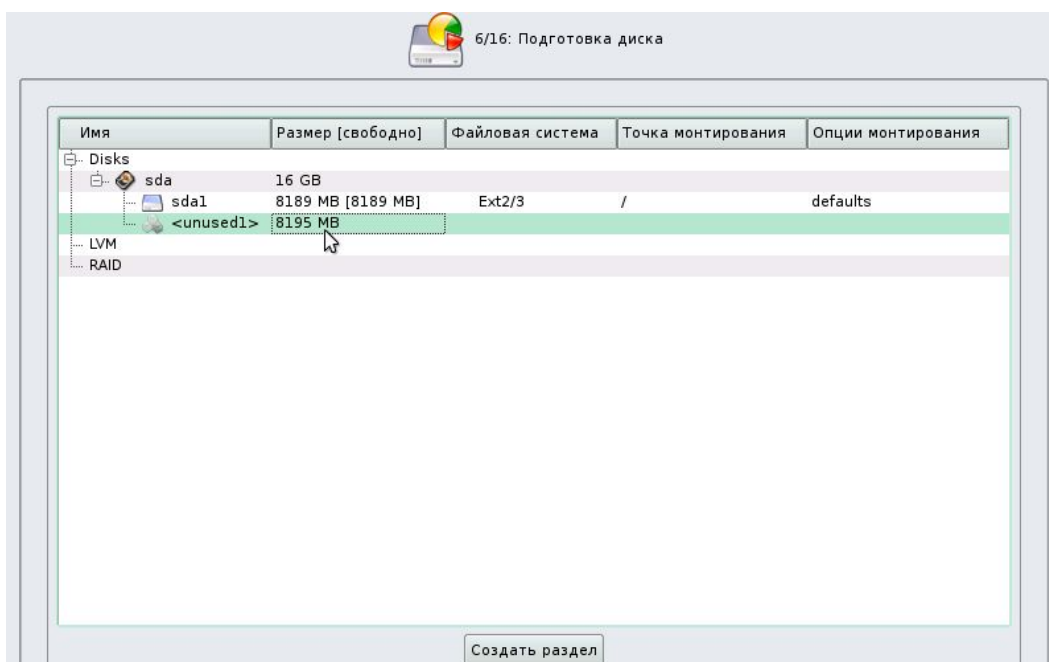


Рис. 13. Результат создания раздела и ФС.

Помимо создания основного раздела нам необходим **SWAP** раздел (аналог файла подкачки в windows-системах).

Для этого выбираем **Создать раздел**, и в экранной форме параметров нового раздела помимо его размера (в нашем случае 1Гб, общая рекомендация – устанавливать размер swap раздела равным удвоенному количеству оперативной памяти компьютера) устанавливаем тип раздела – **Linux Swap**.

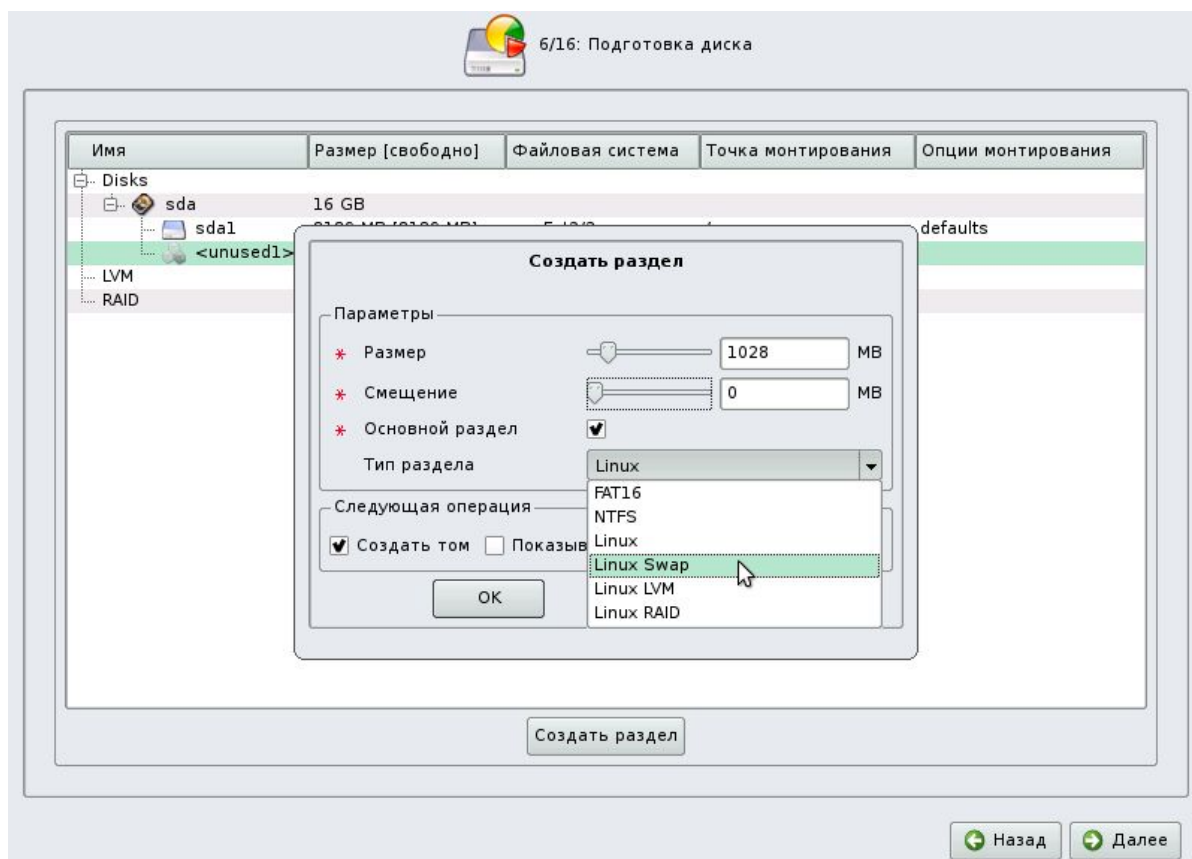


Рис. 14. Создание Swap-раздела.

Файловая система для swap -разделов не запрашивается (она фиксирована), точка монтирования не задается (для swap -разделов не используется).

В результате мы получаем еще один раздел на диске SDA с именем SDA2. На диске остается неиспользованным пространство порядка 7Гб, в котором можно создать еще один, либо несколько разделов, или оставить это место для разделов, которые будут создаваться уже после установки Линукс Мастер.



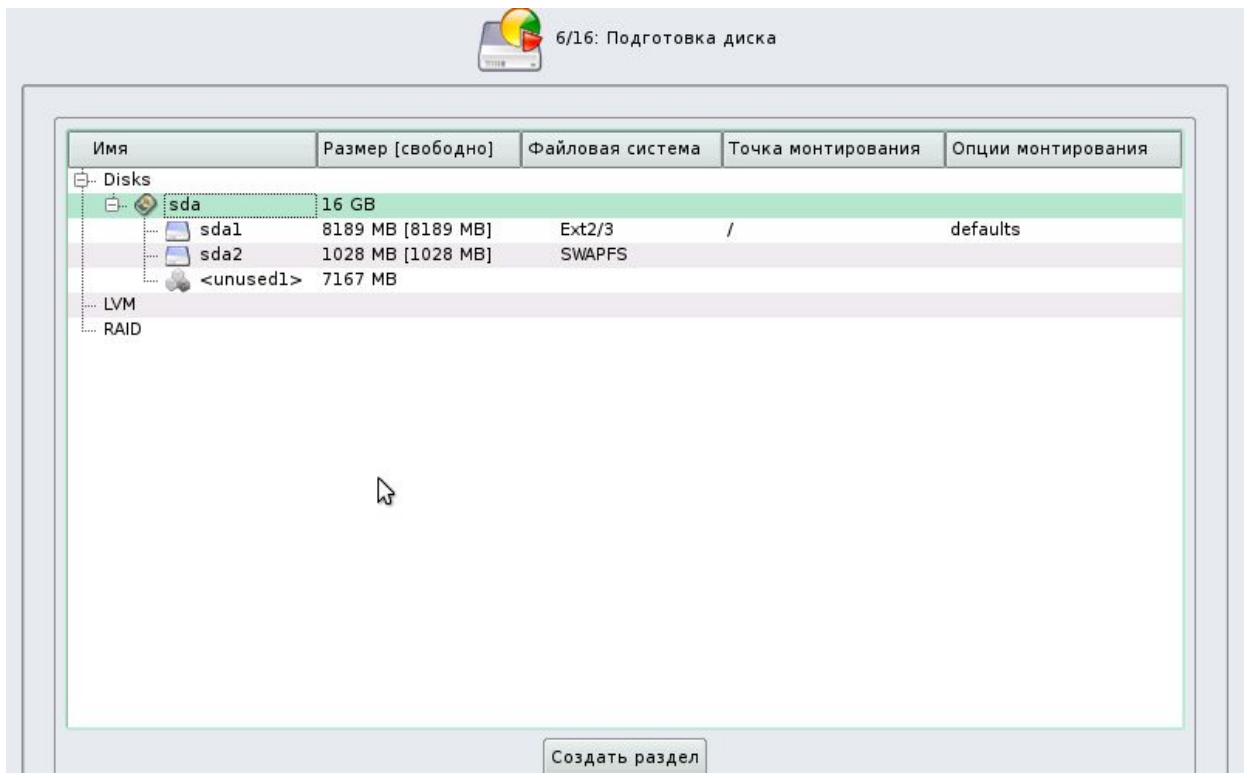


Рис. 15. Завершение ручного разбиения дисков.

Теперь можно нажать кнопку **Далее**, чтобы продолжить процесс установки.

Внимание! Если Вы используете шаблон **Удалить все разделы и создать разделы автоматически**, все данные на диске компьютера будут уничтожены. Соответствующее предупреждение выдается на экран.

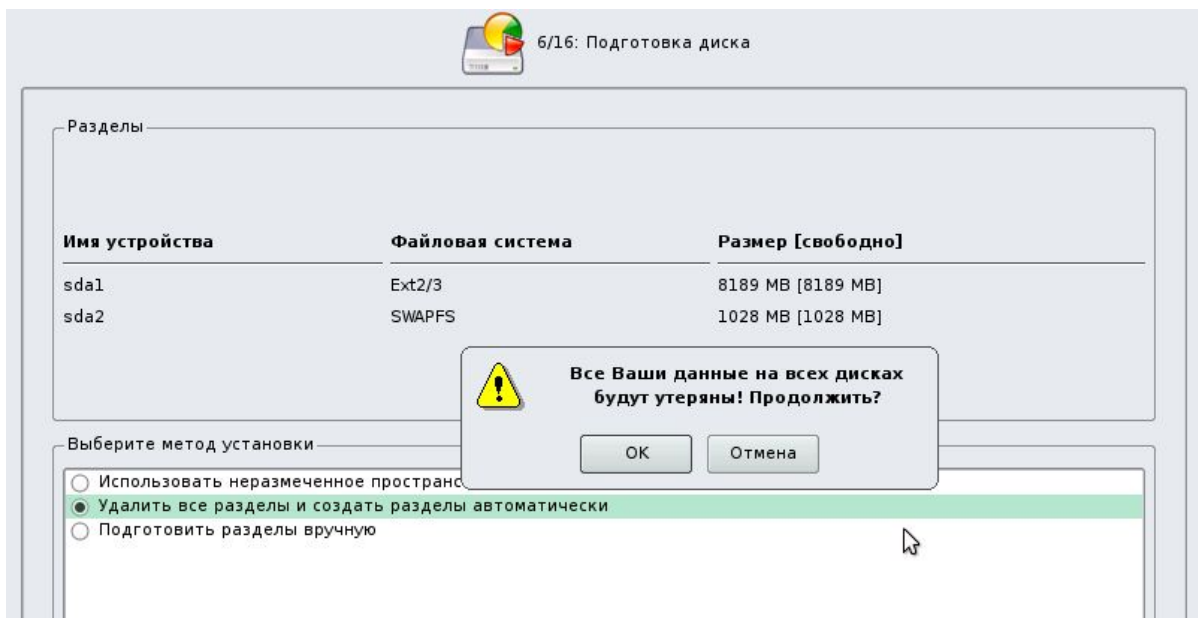


Рис. 16. Предупреждение о потере данных.

По нажатию **Далее** будет произведена запись новой таблицы разделов на диск и форматирование разделов. Разделы, только что созданные на диске программой установки, пока не содержат данных и поэтому форматируются без предупреждения. Уже

существовавшие, но изменённые разделы, которые будут отформатированы, помечаются специальным значком в колонке **Файловая система** слева от названия. Если вы уверены в том, что подготовка диска завершена, подтвердите переход к следующему шагу нажатием кнопки **ОК**.

Не следует форматировать разделы с теми данными, которые вы хотите сохранить, например, с пользовательскими данными (/home) или с другими операционными системами. С другой стороны, отформатировать можно любой раздел, который вы хотите очистить (т. е. удалить все данные).

### Установка Linux после установки Windows

Особо следует остановиться на случае, когда Linux устанавливается поверх установленной Windows, и важно сохранить возможность использовать обе операционные системы. В этом случае рекомендовано использовать шаблон **Использовать неразмеченное пространство**, но при использовании данной возможности возможна ситуация, когда процесс инсталляции считает что свободного места на диске недостаточно, или в результате возникновения ошибок при работе этого шаблона. Возможное решение – подготовка диска в ручном режиме. Проиллюстрируем ситуацию на примере:

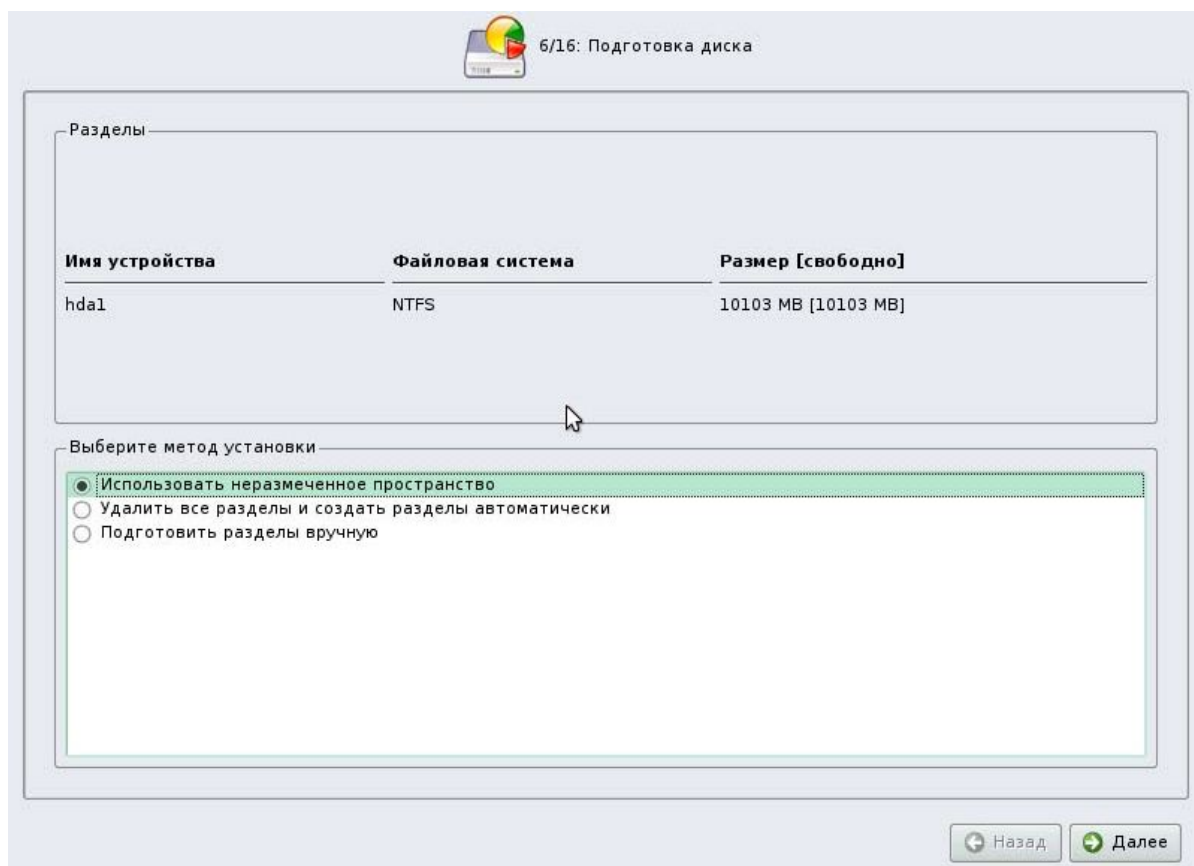


Рис. 17. Установка поверх Windows.

Итак, на приведенном рисунке видно, что на диске компьютера уже имеется раздел с файловой системой NTFS, на котором уже установлена Microsoft Windows.

При попытке использовать шаблон автоматического разбиения свободного пространства диска мы получили сообщение об ошибке – процессу установки оказалось недостаточно места.

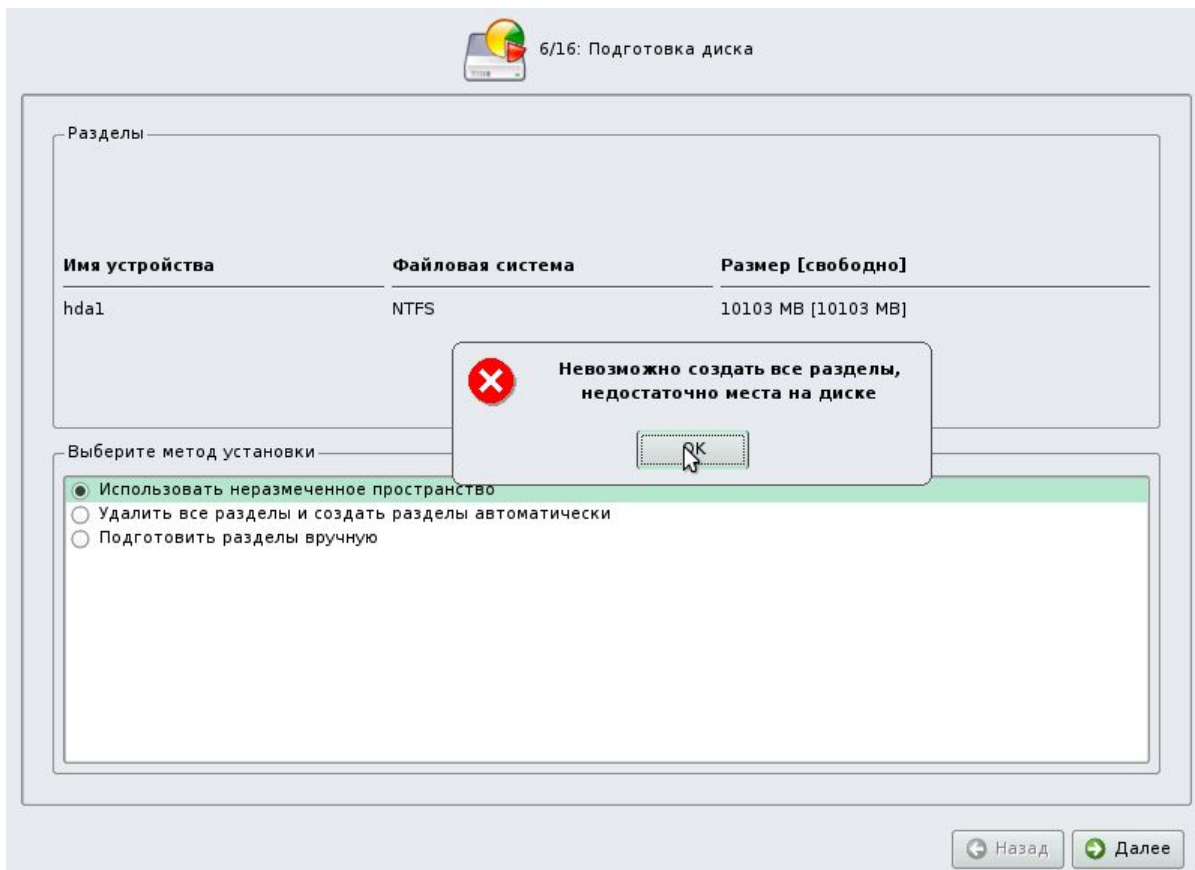


Рис. 18. Нехватка пространства.

Тем не менее, при входе в режим ручного разбиения диска можно видеть, что на диске имеется около 10Гб свободного пространства, что, в общем, вполне достаточно.

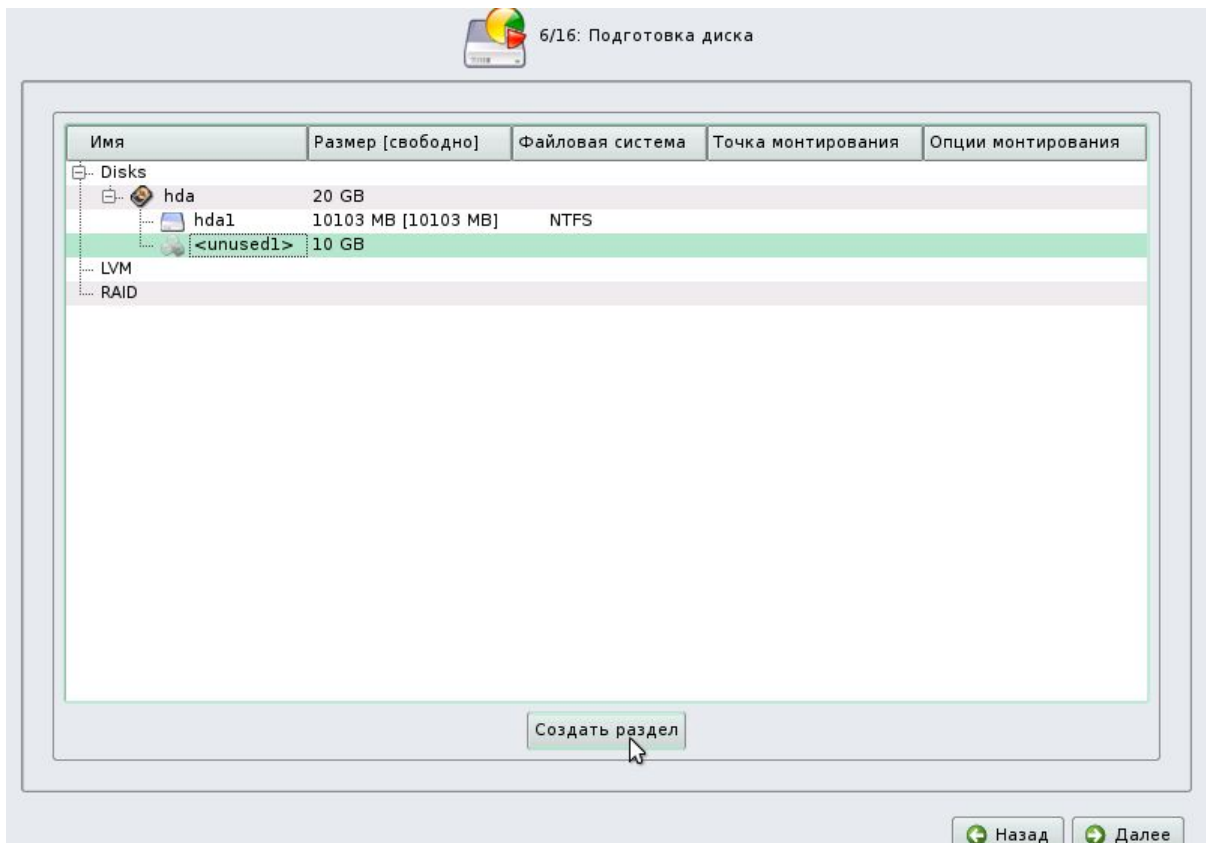


Рис. 19. Состояние дисковых разделов.

Можно попробовать изменить размеры существующих разделов, чтобы обеспечить достаточное количество свободного места, но этот способ имеет высокую вероятность возникновения ошибок и потери данных.

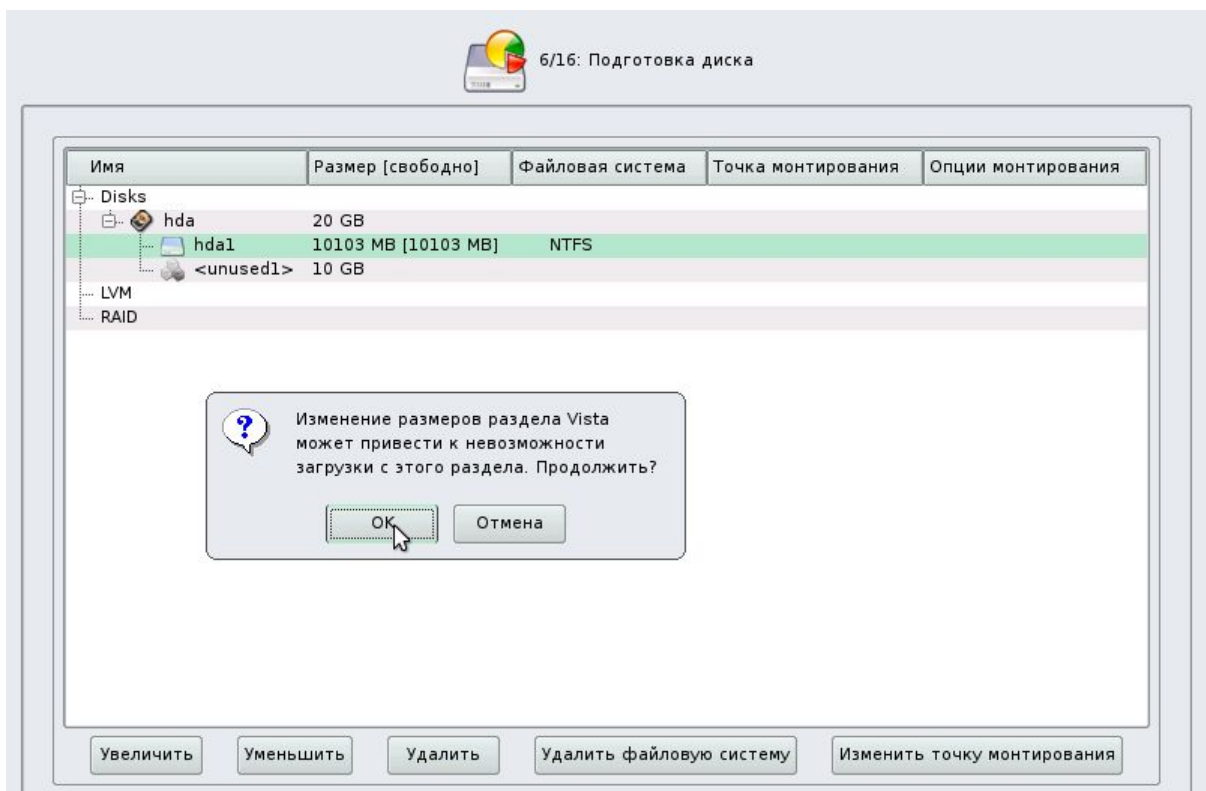


Рис. 20. Изменение разделов.

Принято решение создать разделы для Linux вручную, для чего действуем по описанной ранее методике – создаем **Ext2/3** и **SWAP-разделы** подходящих размеров, куда и будет установлена ОС Linux.

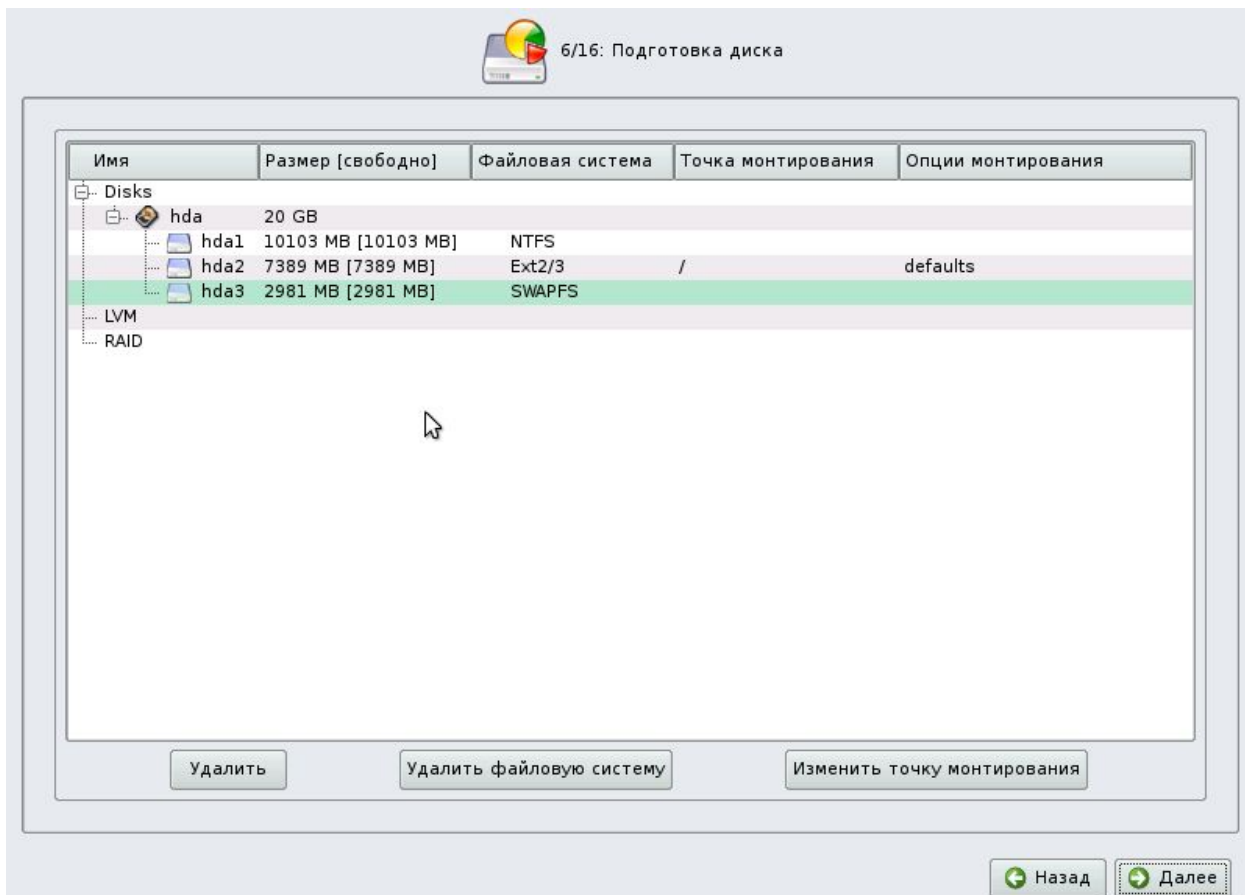


Рис. 21. Разделы созданы вручную.

Можно нажать **Далее** и продолжить установку.

### Установка базовой системы

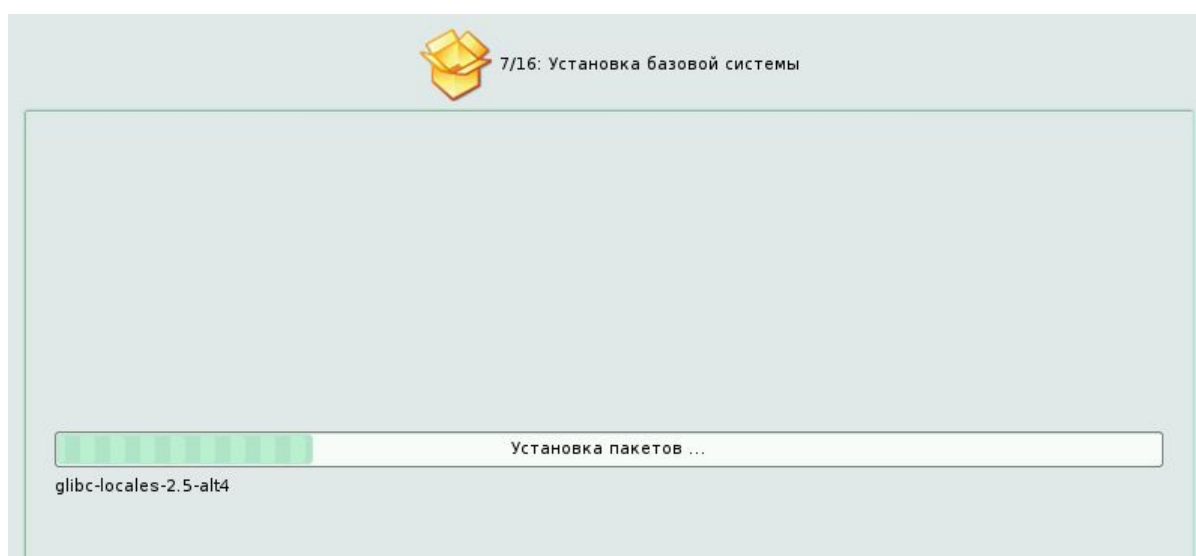


Рис. 22. Установка базовой системы

На этом этапе выполняется установка стартового набора программ, необходимых для запуска и первоначальной работы Linux. Впоследствии процесс установки позволит вам выбрать и установить все необходимые вам для работы приложения.

Установка происходит автоматически в два этапа:

- Получение пакетов
- Установка пакетов

Получение пакетов осуществляется из источника, выбранного на этапе начальной загрузки. При сетевой установке (по протоколу FTP или HTTP) время выполнения этого шага будет зависеть от скорости соединения и может быть значительно больше, чем при установке с лазерного диска.

Когда базовая система будет установлена, вы сможете произвести начальную настройку, выполнить конфигурацию сетевых адаптеров и сетевые подключения. Свойства системы, которые были заданы при установке (например, язык системы), могут быть изменены впоследствии в любой момент как при помощи стандартных для Linux средств, так и специализированных модулей управления, включённых в дистрибутив.

Установка базовой системы может занять некоторое время.

### Сохранение настроек

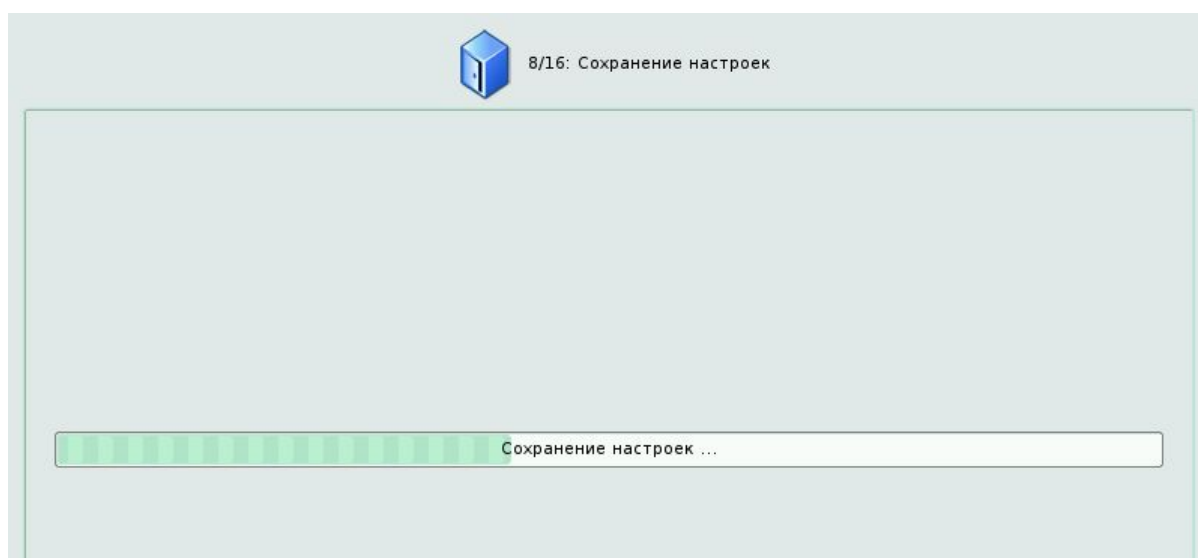


Рис. 23. Сохранение настроек

Начиная с этого шага, программа установки работает с файлами только что установленной базовой системы. Все последующие изменения можно будет совершить после завершения установки посредством редактирования соответствующих конфигурационных файлов или при помощи модулей управления, включённых в дистрибутив.

По завершении установки базовой системы начинается этап сохранения настроек. Он проходит автоматически и не требует вмешательства пользователя, на экране отображается прогресс выполнения.

На этом шаге производится перенос настроек, выполненных на первых пяти шагах установки (настройки языка, часового пояса, даты и времени, переключения раскладки клавиатуры), в только что установленную базовую систему. Также производится запись информации о соответствии разделов жёсткого диска смонтированным на них файловым системам (заполняется конфигурационный файл `/etc/fstab`). В список доступных источников программных пакетов добавляется репозиторий, находящийся на установочном лазерном диске, т. е. выполняется команда `apt-cdrom add`, осуществляющая запись в конфигурационный файл `/etc/apt/sources.list`. Подробнее об этом можно узнать в разделе электронной документации **Система управления пакетами АРТ**.

После того, как настройки сохранены, осуществляется автоматический переход к следующему шагу.

### Дополнительные диски



Рис. 23. Дополнительные диски

При наличии диска с дополнительным программным обеспечением, возможно использовать его и установить требуемые компоненты. Установите диск в привод

лазерных дисков и нажмите **Добавить**. Новый диск появится в списке доступных. Нажав кнопку **Далее**, вы попадаете на этап установки дополнительных пакетов.

### Дополнительные пакеты



Рис. 24. Дополнительные пакеты

В любом дистрибутиве Linux имеется большое количество программ, часть из которых составляет саму операционную систему, а все остальные — это *прикладные программы и утилиты*.

В операционной системе Linux все операции установки и удаления производятся над пакетами — *отдельными компонентами системы*. Пакет и программа соотносятся неоднозначно: иногда одна программа состоит из нескольких пакетов, иногда один пакет включает несколько программ.

В процессе установки системы обычно не требуется детализированный выбор компонентов на уровне пакетов — это требует слишком много времени и знаний от проводящего установку. Тем более, что комплектация дистрибутива подбирается таким образом, чтобы из имеющихся программ можно было составить полноценную рабочую среду для соответствующей аудитории пользователей. Поэтому в процессе установки системы пользователю предлагается выбрать из небольшого списка *групп пакетов*, объединяющих пакеты, необходимые для решения наиболее распространённых задач. Под



списком групп на экране отображается информация об объёме дискового пространства, которое будет занято после установки пакетов, входящих в выбранные группы.

Выбрав необходимые группы, следует нажать **Далее**, после чего начнётся установка пакетов.

### 3.7. Конфигурация системного загрузчика

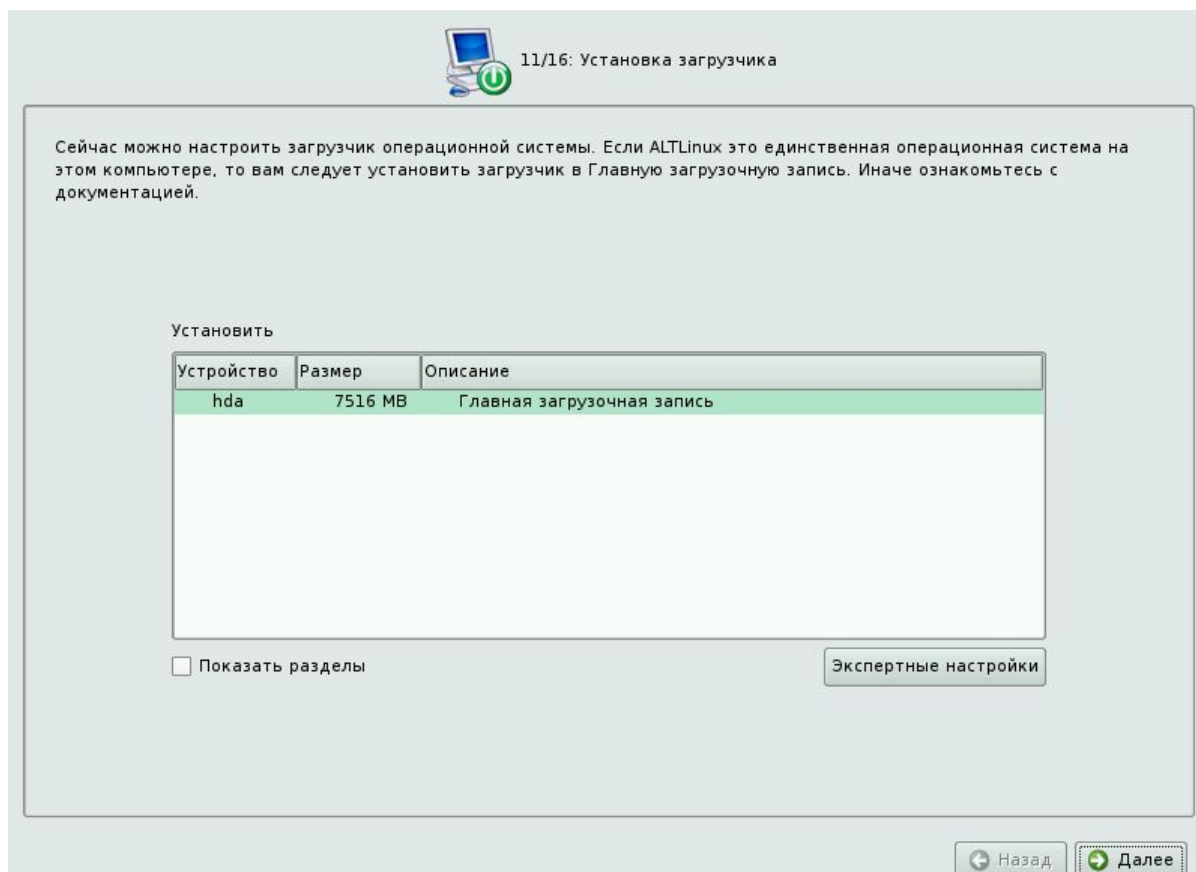


Рис. 25. Установка загрузчика

*Загрузчик Linux* — программа, которая позволяет загружать Linux и другие операционные системы. Если на вашем компьютере будет установлен только Linux, то здесь не нужно ничего изменять, просто нажмите **Далее**.

Если же вы планируете использовать и другие операционные системы, уже установленные на этом компьютере, тогда имеет значение, на каком жёстком диске или разделе будет расположен загрузчик.

Опытным пользователям может пригодиться возможность тонкой настройки загрузчика (кнопка **Экспертные настройки**). Параметры, которые можно здесь изменять, напрямую соотносятся с соответствующими параметрами конфигурационного файла загрузчика LILO (`/etc/lilo.conf`). Для простоты сохранены латинские названия параметров, об их значении можно справиться в документации по LILO (`lilo.conf(5)`).

В случае, когда Линукс устанавливался поверх Windows, при последующих загрузках Вам будет предложено экранное меню, позволяющее выбрать какую именно ОС загружать:

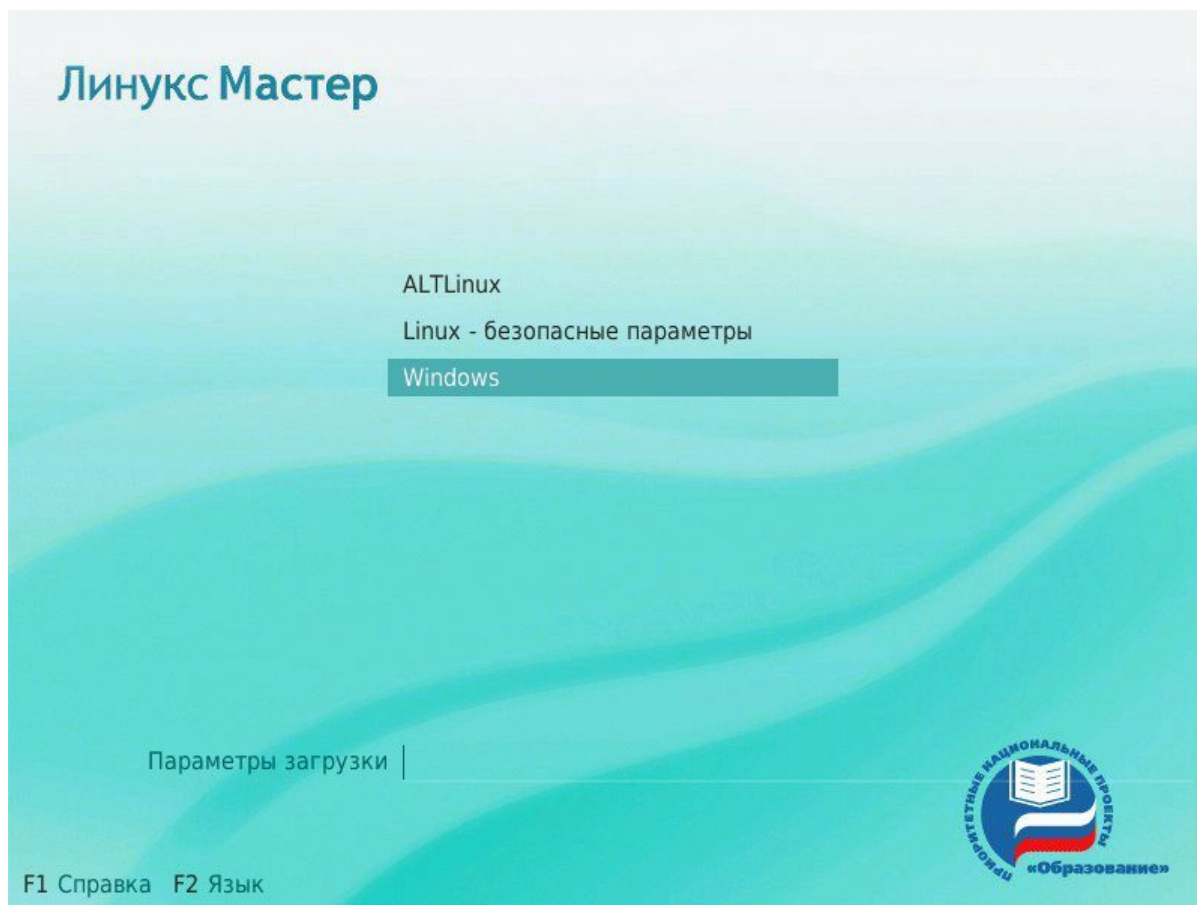


Рис. 26. Стартовое меню.

### 3.8. Начальные установки (сеть, пользователи, сообщения диагностики)

Linux — это многопользовательская система. На практике это означает, что для работы в системе нужно в ней *зарегистрироваться*, т. е. дать понять системе, кто именно находится за монитором и клавиатурой. Наиболее распространённый способ регистрации на сегодняшний день — использование **системных имён** (login name) и паролей. Это надёжное средство убедиться, что с системой работает тот, кто нужно, если пользователи хранят свои пароли в секрете и если пароль достаточно сложен и не слишком короток (иначе его легко угадать или подобрать).

## Администратор системы

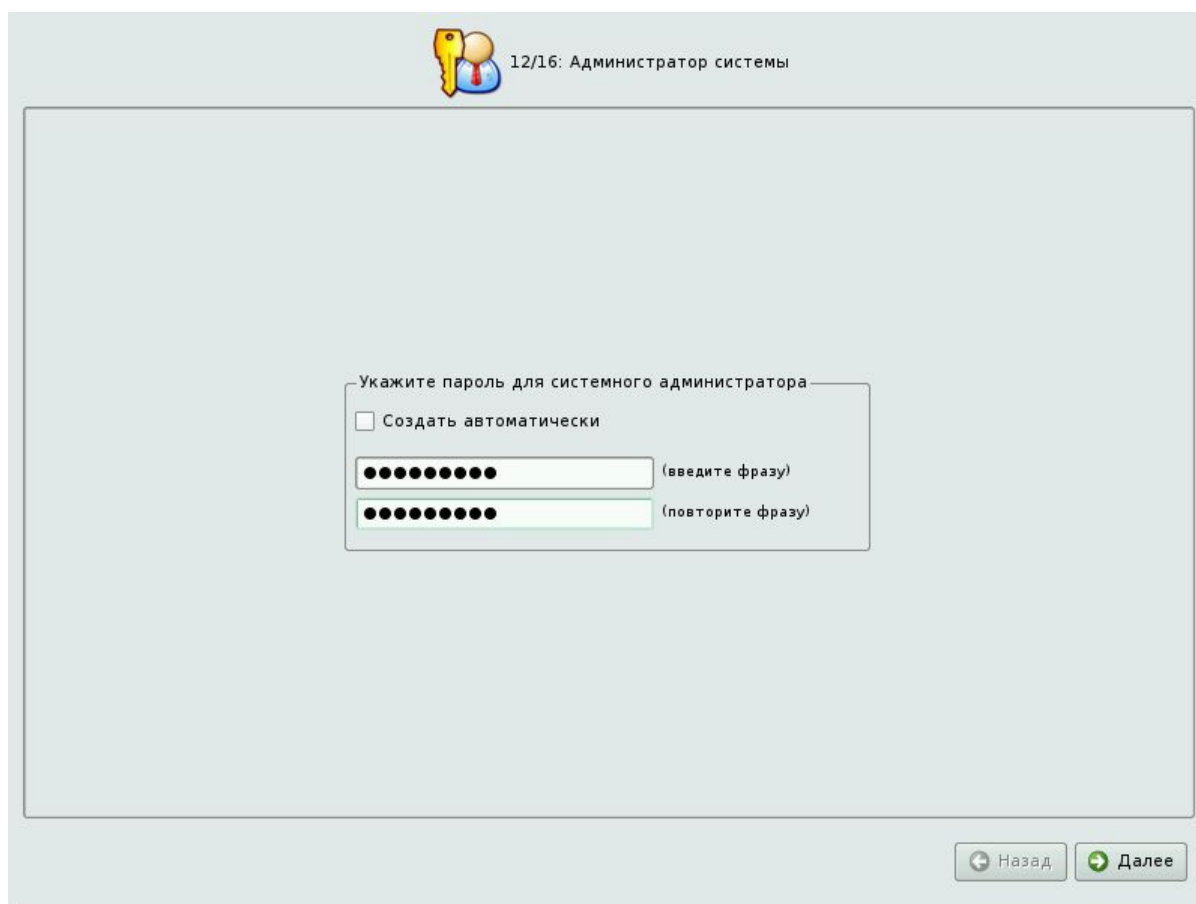


Рис. 27. Администратор системы

В любой системе Linux всегда присутствует один специальный пользователь — администратор, он же **суперпользователь**, для него зарезервировано стандартное системное имя — **root**.

Стоит запомнить пароль root — его нужно будет вводить, чтобы получить право изменять настройки системы с помощью стандартных средств настройки Linux.

При наборе пароля вместо символов на экране высвечиваются звёздочки. Чтобы избежать опечатки при вводе пароля, его предлагается ввести дважды. Можно воспользоваться автоматическим созданием пароля, выбрав **Создать автоматически**. Вам будет предложен случайно сгенерированный и достаточно надёжный вариант пароля. Можно принять автоматически сгенерированный пароль (не забудьте при этом запомнить пароль!) или запросить другой вариант пароля при помощи кнопки **Сгенерировать**.

Администратор отличается от всех прочих пользователей тем, что ему позволено производить любые, в том числе самые разрушительные, изменения в системе. Поэтому выбор пароля администратора — очень важный момент для безопасности: любой, кто сможет ввести его правильно (узнать или подобрать), получит неограниченный доступ к

системе. Даже ваши собственные неосторожные действия от имени root могут иметь катастрофические последствия для всей системы.

### Системный пользователь

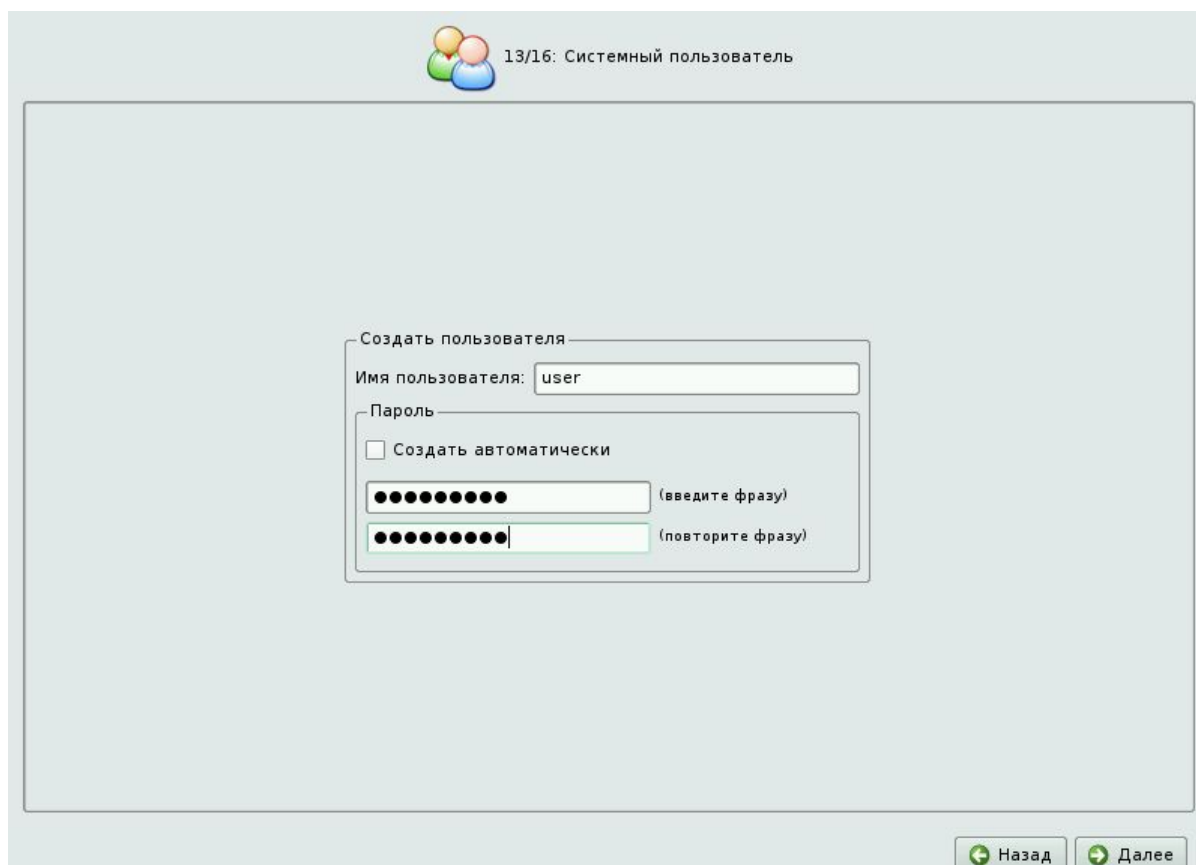


Рис. 28. Системный пользователь

Помимо администратора (root) в систему необходимо добавить, по меньшей мере, одного **обычного пользователя**. Работа от имени администратора считается опасной (можно по неосторожности повредить систему), поэтому повседневную работу в Linux следует выполнять от имени обычного пользователя, полномочия которого ограничены.

При добавлении пользователя предлагается ввести имя учётной записи (login name) пользователя. Имя учётной записи всегда представляет собой одно слово, состоящее только из строчных латинских букв (заглавные запрещены), цифр и символа подчёркивания “\_” (причём цифра и символ “\_” не могут стоять в начале слова). Чтобы исключить опечатки, пароль пользователя вводится дважды. Так же, как при выборе пароля администратора (root), можно создать пароль автоматически.

В процессе установки предлагается создать только одну учётную запись обычного пользователя — чтобы от его имени системный администратор мог выполнять задачи, которые не требуют привилегий суперпользователя.

Учётные записи для всех прочих пользователей системы можно будет создать в любой момент после её установки.

## Настройка сети



Рис. 27. Настройка сети

Существует ряд сетевых параметров, которые являются общими для всех подключений к сети и должны быть определены даже тогда, когда компьютер не подключён ни к какой сети. Для подключения к локальной сети необходимо к тому же настроить **сетевое подключение**, которое обычно привязывается к определённому физическому устройству — сетевой карте (Ethernet).

В поле **Полное доменное имя компьютера** необходимо указать сетевое имя компьютера вида **computer.domain**. Несмотря на то, что этот параметр никому из соседних компьютеров в сети не передаётся (в отличие, скажем, от имени компьютера в Windows-сети), его используют многие сетевые службы, например, почтовый сервер. Если компьютер не подключён к локальной сети, доменное имя может выглядеть как угодно, можно оставить значение по умолчанию (localhost.localdomain).

В случае локальной сети программа установки автоматически настроит все установленные на компьютере **сетевые интерфейсы** (сетевые карты). При наличии в сети DHCP-сервера все необходимые параметры (IP-адрес, маска сети, шлюз по умолчанию, адреса серверов DNS) будут получены автоматически. Если сервера DHCP нет, при подключении к сети интерфейс будет настроен при помощи IPv4LL- технологии,

позволяющей автоматически подбирать свободный IP-адрес, не используемый другими компьютерами в сети, и присваивать его сетевому интерфейсу.

Можно отказаться от автоматической настройки сетевых подключений, убрав отметку с пункта **автоматически сконфигурировать все сетевые карты**. После завершения установки вы сможете в любой момент настроить доступ к сети самостоятельно.

### Настройка графической системы

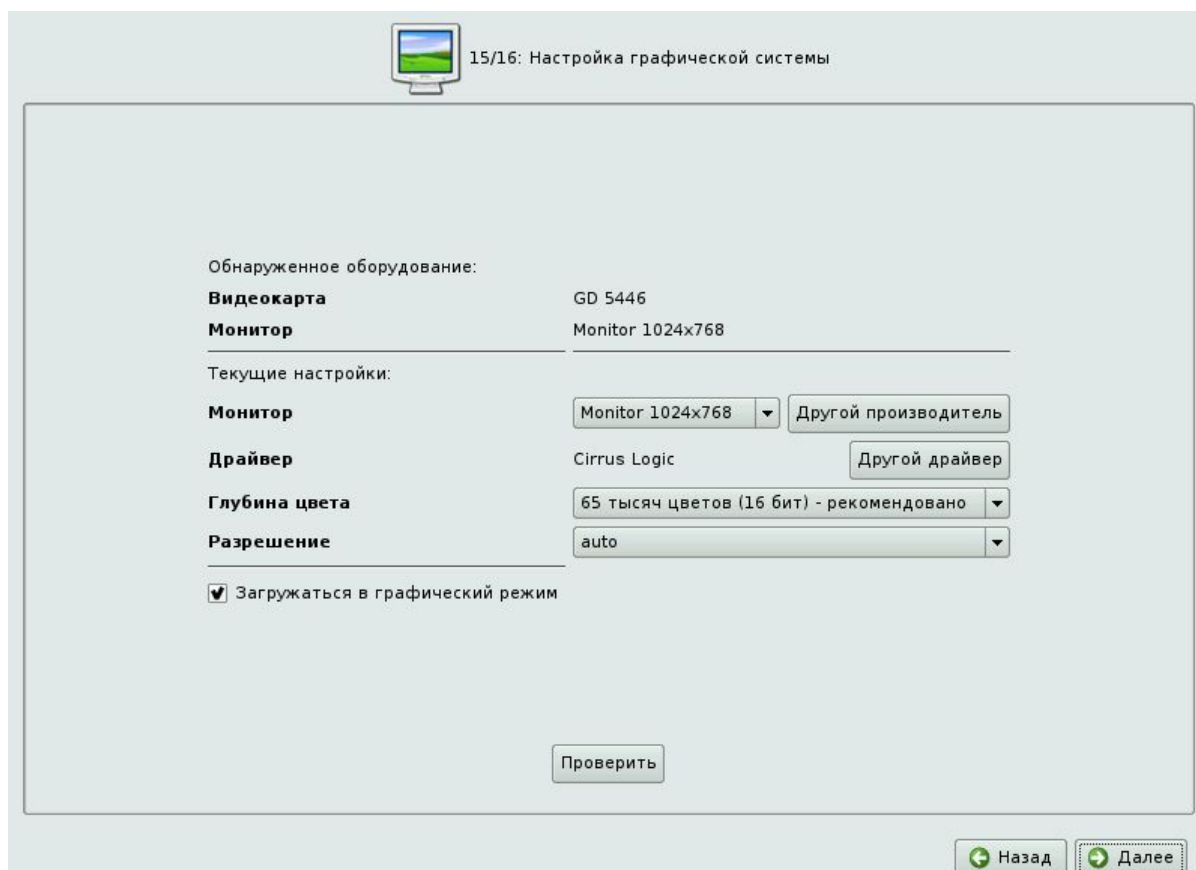


Рис. 28. Настройка графической системы

Современное графическое оборудование в большинстве случаев поддается автоматическому определению, хотя некоторое очень новое или редкое оборудование может отсутствовать в базе данных. Автоматически определённые видеокарта и монитор будут указаны в разделе Обнаруженное оборудование. В разделе Текущие настройки будут предложены наиболее подходящие настройки графического режима — их стоит испробовать в первую очередь. Довольно часто видеокарта может работать с несколькими разными драйверами. По умолчанию предлагается тот, который считается наилучшим для данной модели.

Нужно заметить, что оптимальные настройки — это не всегда максимальные значения из возможных (разрешение, глубина цвета и т. п.). При указании рекомендуемых значений учитываются свойства конкретного оборудования и драйвера, поэтому выбор более высоких значений не обязательно приведёт к улучшению качества изображения.

Если оборудование автоматически не определилось, то драйвер для видеокарты и модель монитора придётся выбрать вручную.

Проверить работоспособность выбранных параметров можно, нажав на кнопку **Проверить**. В случае успешной активации графического режима с новыми параметрами, вы увидите сообщение на чёрном экране, где можете либо подтвердить работоспособность графического режима нажатием кнопки **Да**, либо отказаться от текущих настроек, нажав **Нет**. Кнопка **Стоп** служит для приостановки счётчика времени задержки перед возвращением в диалог настройки графического режима. Если нажимать никаких кнопок в окне тестирования видеорежима, к примеру, если из-за неверных настроек графического режима данное сообщение вообще не отобразилось на экране, то через несколько секунд будет возвращено исходное состояние, где вы можете выбрать более подходящие настройки.

Обратите внимание на отметку **Загружаться в графический режим**: новичку в Linux нужно проследить, чтобы она была установлена. В противном случае загрузка будет заканчиваться приглашением к регистрации в системе (login:) в текстовом режиме.

### **Смена драйвера видеокарты**

При необходимости вы можете сменить драйвер видеокарты. В списке перечислены названия доступных драйверов с указанием через дефис производителя и, в некоторых случаях, моделей видеокарт. Вы можете выбрать тот их них, который считаете наиболее подходящим. Драйвер, рекомендуемый для использования, помечен рекомендовано.

Если в списке нет драйвера для вашей модели видеокарты, можно попробовать один из двух стандартных драйверов: **vga — Generic VGA Compatible** или **vesa — Generic VESA Compatible**.

### **Выбор модели монитора**

Модели мониторов можно выбирать по производителям: кнопка **Другой производитель**. Ускорить передвижение по спискам можно, набирая первые буквы искомого слова. После выбора производителя в списке становятся доступны модели мониторов данного производителя. Не всегда обязательно подбирать монитор с точностью до номера модели: некоторые пункты в списке не содержат конкретного номера модели, а указывают на целый ряд устройств, например **Dell 1024x768 Laptop Display Panel**.

Если в списке не нашлось производителя или близкой модели, то можно попробовать один из стандартных типов монитора. Для этого в списке производителей нужно выбрать **Generic CRT Display** (для электронно-лучевых мониторов) либо **Generic**

**LCD Display** (для жидкокристаллических мониторов), а далее выбрать модель, руководствуясь желаемым разрешением.

### Завершение установки

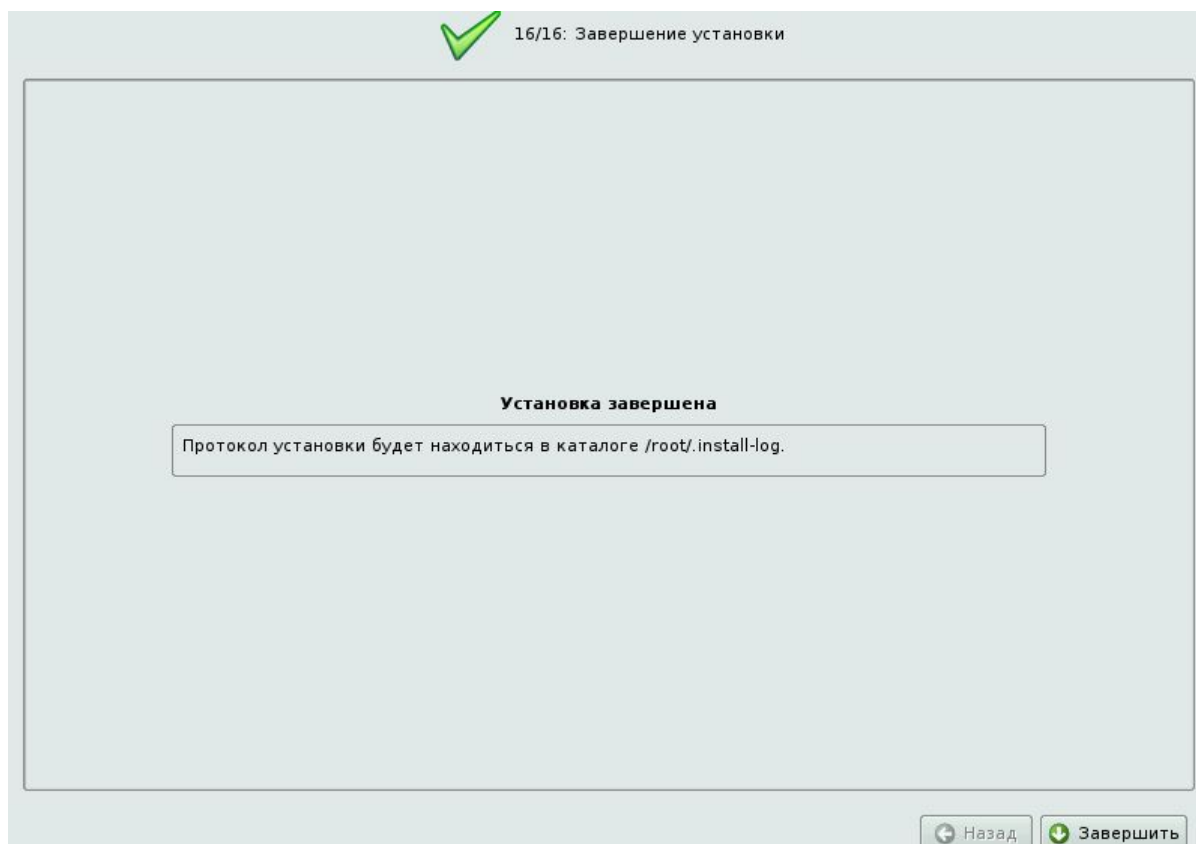


Рис. 29. Завершение установки

На экране последнего шага установки отображается информация о местонахождении протокола установки (каталог `/root/.install-log`). После нажатия кнопки **Завершить** и перезагрузки компьютера можно загрузить установленную систему в обычном режиме.

## 4. Установка и настройка прикладного программного обеспечения

В современных системах на базе ПСПО огромное число общих ресурсов, которыми пользуются сразу несколько программ: разделяемых библиотек, содержащих стандартные функции, исполняемых файлов, сценариев и стандартных утилит и т. д. Удаление или изменение версии одного из составляющих систему компонентов может повлечь неработоспособность других, связанных с ним компонентов, или даже вывести из строя всю систему. В контексте системного администрирования проблемы такого рода называют нарушением целостности системы. Задача администратора — обеспечить наличие в системе согласованных версий всех необходимых программных компонентов (обеспечение целостности системы).



Для установки, удаления и обновления программ и поддержания целостности системы в ПСПО в первую очередь стали использоваться менеджеры пакетов. С точки зрения менеджера пакетов программное обеспечение представляет собой набор компонентов — программных пакетов. Такие компоненты содержат в себе набор исполняемых программ и вспомогательных файлов, необходимых для корректной работы программного обеспечения. Менеджеры пакетов облегчают установку программ: они позволяют проверить наличие необходимых для работы устанавливаемой программы компонент подходящей версии непосредственно в момент установки, а также производят необходимые процедуры для регистрации программы во всех операционных средах пользователя: сразу после установки программа может быть доступна пользователю из командной строки и — если это предусмотрено — появляется в меню всех графических оболочек.

Дистрибутивы ПСПО выпускаются на основе репозитория **Sisyphus**. Следует иметь в виду, что Sisyphus не является самостоятельным дистрибутивом, а отражает текущее состояние разработки и может содержать нестабильные версии пакетов. Периодически на базе этого проекта выпускаются отдельные оттестированные срезы — дистрибутивы.

В отличие от Sisyphus, ежедневно обновляемого разработчиками, такие срезы являются замороженными — разработка в них не ведётся, и сами срезы сохраняются в целях обеспечения целостности среды дистрибутива, в которой уже не должны обновляться версии пакетов. Единственное исключение делается для обновлений, исправляющих проблемы в безопасности системы, однако такие обновления помещаются в отдельном репозитории для каждого дистрибутива.

Пользователи стабильных дистрибутивов не всегда готовы переходить на нестабильную ветку разработки, и в то же время заинтересованы в обновлении версий некоторых прикладных программ (в которых появляется новая функциональность и т. п.). Для этих целей заинтересованными пользователями для каждого дистрибутива ПСПО создаются и поддерживаются специальные репозитории с обновлёнными версиями программ — **backports**. Репозитории backports являются согласованными по пакетной базе с основным репозиторием соответствующего дистрибутива, и могут быть без опасений подключены параллельно с ним. Непосредственно после установки дистрибутива ПСПО в **/etc/apt/sources.list** обычно указывается несколько репозиториев:

- репозиторий обновлений в системе безопасности дистрибутива;
- полный срез репозитория Sisyphus, подмножеством которого является

дистрибутив.

## 4.1. Установка пакетов

Synaptic запускается из меню **К — Настройка — Менеджер пакетов** (Программа управления пакетами Synaptic). При запуске программы запрашивается пароль суперпользователя (root). Если вы хотите установить пакет, выполните следующие шаги:

1. Обновите информацию о пакетах, чтобы узнать о последних доступных версиях, выбрав в меню **Редактирование — Получить сведения о пакетах** или нажав **Получить сведения** на панели инструментов.

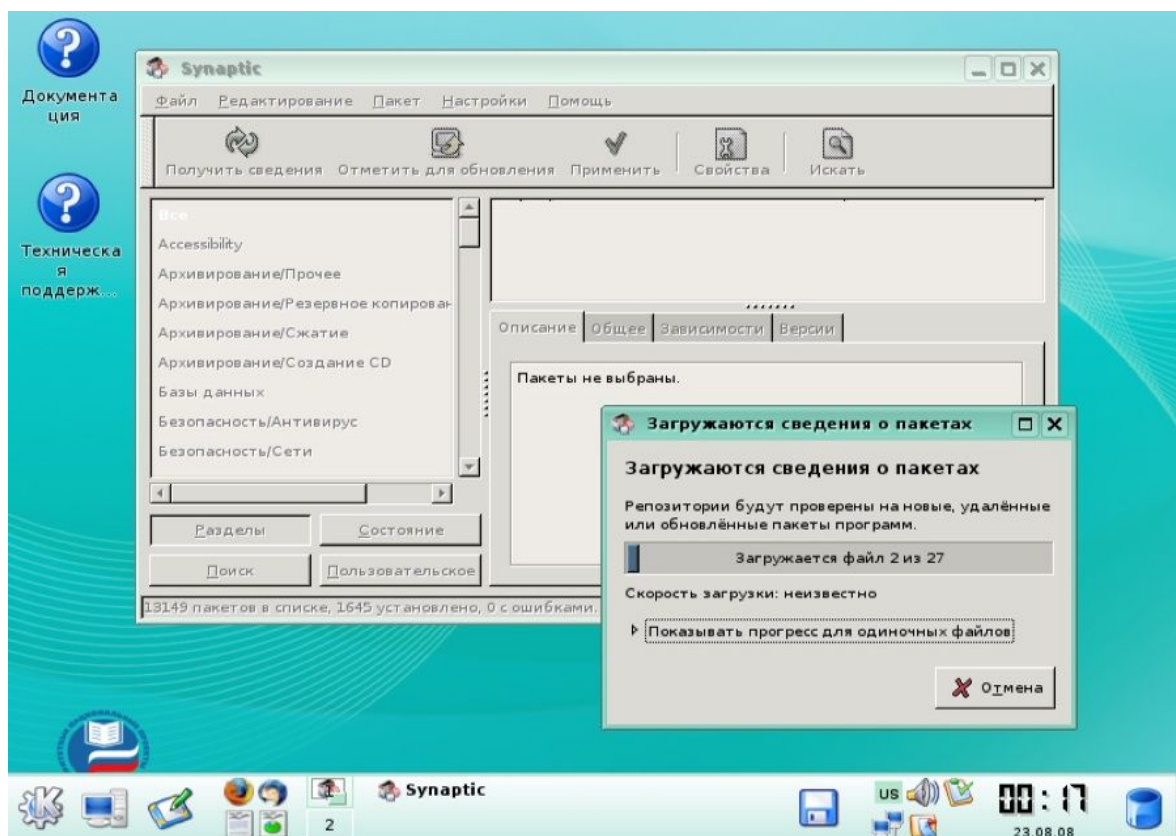


Рис. 1. Получение информации о пакетах

2. Отметьте пакет для установки:

- Двойным щелчком мыши на названии пакета в списке пакетов.
- Нажав правой кнопкой мыши на пакете и выбрав **Отметить для установки** в контекстном меню.
- Выделив пакет и выбрав в меню **Пакет — Отметить для установки**. Если установка пакетов требует дополнительных изменений, вас спросят о подтверждении. Чтобы отметить дополнительные изменения, нажмите **Применить**.

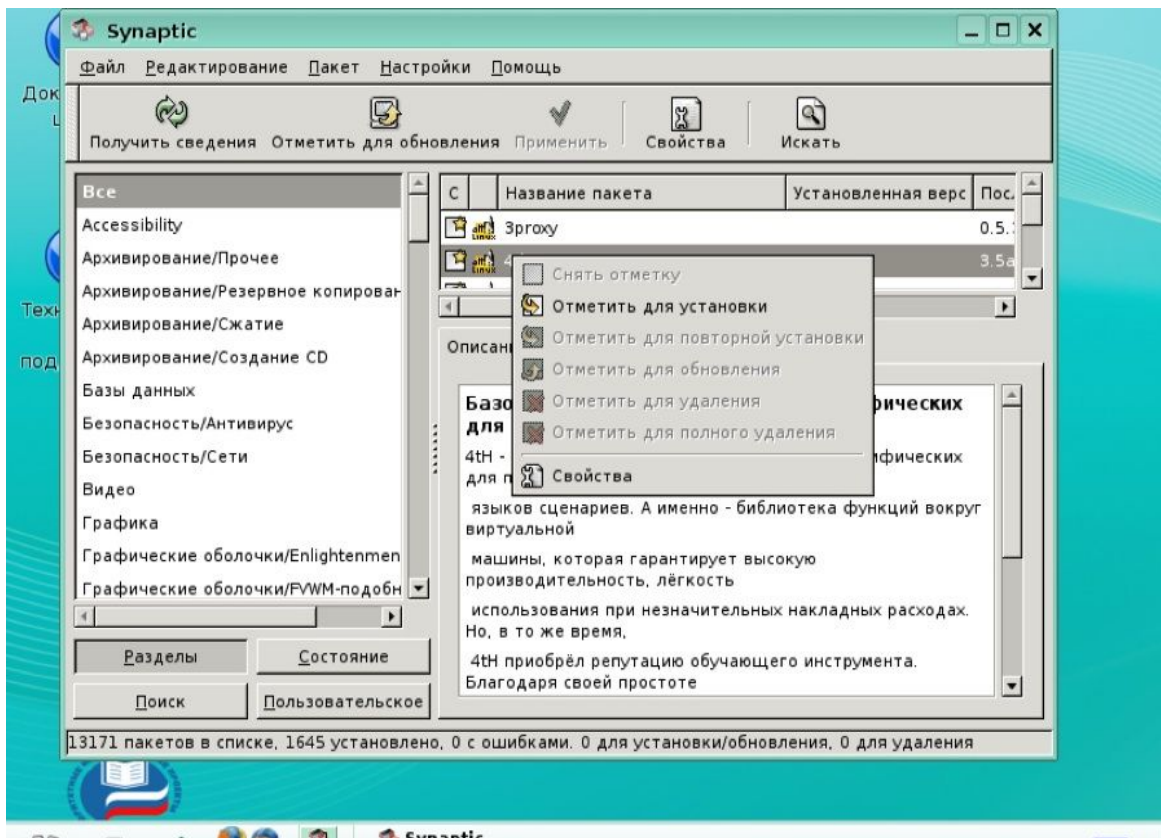


Рис. 2. Выбор пакета для установки

3. Примените отмеченные изменения, чтобы установить пакеты, нажав **Применить** на панели инструментов или выбрав в меню **Редактирование** — **Внести отмеченные изменения**.

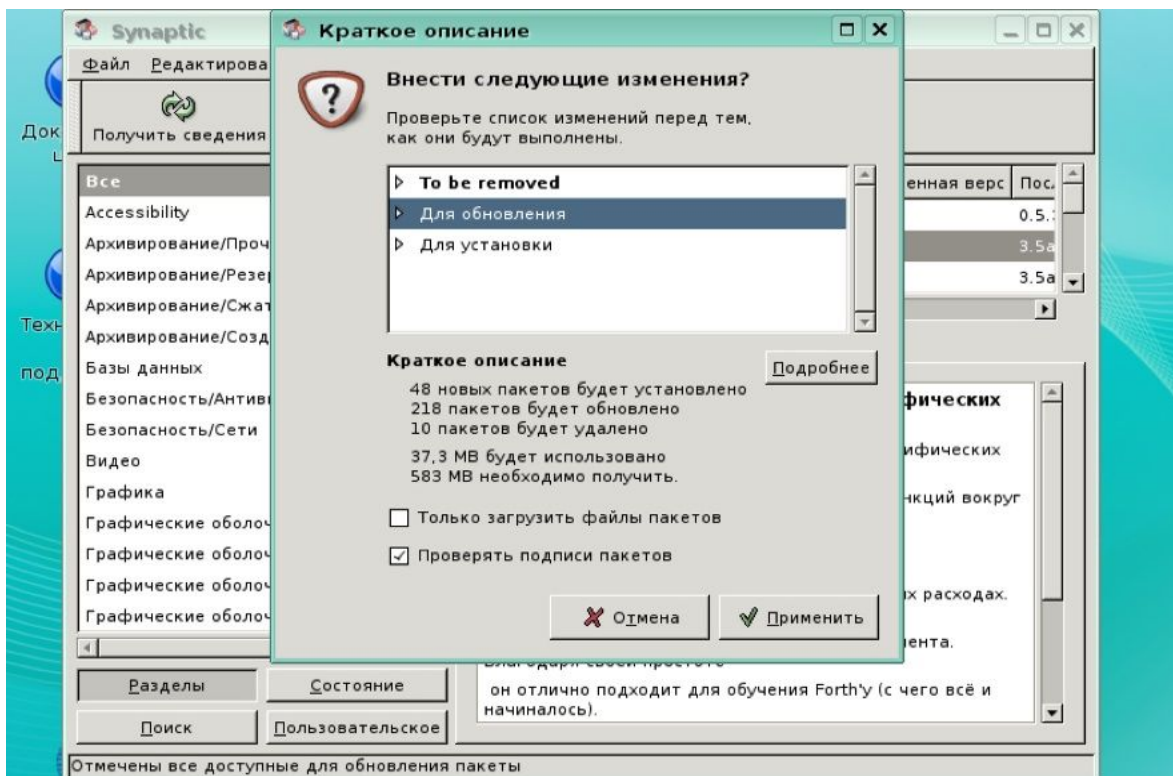


Рис. 3. Внести отмеченные изменения

4. Вас спросят о подтверждении. Проверьте итоговые изменения, которые будут применены. Чтобы продолжить установку подтвердите изменения, нажав **Применить**.

5. Во время проведения изменений вы увидите строку состояния. Подождите, пока изменения будут применены. Это может занять некоторое время в зависимости от количества изменений. После этого вы вернётесь в основное меню.

## 4.2. Удаление пакетов

Чтобы удалить пакет, выполните следующие шаги:

1. Отметьте пакет для удаления:
  - Двойным щелчком мыши на названии установленного пакета в списке пакетов.
  - Нажав правой кнопкой мыши на пакете и выбрав **Отметить для удаления** в контекстном меню.
  - Выделив пакет и выбрав в меню **Пакет — Отметить для удаления**.
2. Примените отмеченные изменения, чтобы удалить пакеты (так же, как при установке пакетов).
3. Вас спросят о подтверждении. Проверьте итоговые изменения, которые будут применены, после чего подтвердите изменения.
4. Во время проведения изменений вы увидите строку состояния. Когда изменения будут применены, вы вернётесь в основное меню.

## 4.3. Установка доступных обновлений

Существует возможность установки более новых версий уже установленных пакетов, или новой установки пакетов, не включенных в дистрибутив. Чтобы установить доступные обновления, выполните следующие шаги:

1. В параметрах конфигурации менеджера пакетов определите, с каким именно репозитарием Вы собираетесь работать. Для этого выберите в меню **Настройки—Репозитории** и в открывшемся списке доступных сетевых ресурсов отметьте галочкой тот, с которым собираетесь работать.
2. Обновите информацию о пакетах, чтобы узнать о последних доступных версиях, как в случае установки пакетов.
3. Отметьте пакет для обновления:
  - Двойным щелчком мыши на названии пакета с последней доступной версией в списке пакетов.

- Нажав правой кнопкой мышки на пакете и выбрав **Отметить для обновления** в контекстном меню.
- Выделив пакет и выбрав в меню **Пакет — Отметить для обновления**. Как и в описанных выше случаях, при необходимости дополнительных изменений вас спросят о подтверждении.
- 4. Примените отмеченные изменения, чтобы обновить пакеты.
- 5. Вас спросят о подтверждении. Проверьте итоговые изменения, которые будут применены. Чтобы продолжить обновление, подтвердите изменения, нажав **Применить**.
- 6. Во время проведения изменений Вы увидите строку состояния. Когда изменения будут применены, вы вернётесь в основное меню.

#### 4.4. Обновление всей системы

Менеджер пакетов Synaptic предусматривает два метода выделения пакетов для обновления:

1. *Обновление по умолчанию*. По умолчанию обновляются только установленные пакеты. Если более свежая версия пакета зависит от неустановленных пакетов или конфликтует с уже установленным пакетом, обновление не будет отмечено.

7. *Умное обновление (Dist-Upgrade)*. Метод умного обновления старается разрешить конфликты между пакетами интеллектуально. Это включает установку дополнительных требуемых пакетов и предпочтение пакетов с более высоким приоритетом. Умное обновление также известно как dist-upgrade при использовании apt-get в интерфейсе командной строки.

*Замечание:* обновления до более свежих выпусков операционной системы должны производиться с помощью метода умного обновления.

Чтобы обновить вашу систему до самой последней версии выполните следующие шаги:

1. Обновите информацию о пакетах, чтобы узнать о последних доступных версиях.
2. Отметьте все возможные изменения, нажав **Отметить для обновления** на панели инструментов или выбрав в меню **Редактирование — Отметить для обновления**.
3. Выберите метод обновления (умное обновление).
4. Примените отмеченные изменения, чтобы обновить пакеты.
5. Проверьте итоговые изменения. Чтобы продолжить обновление подтвердите изменения, нажав **Применить**.

6. Во время проведения изменений вы увидите строку состояния. После применения изменений вы вернётесь в основное меню.

Совет: вы можете изменить метод обновления по умолчанию в настройках synaptic (меню **Настройки — Параметры**).

Synaptic — это очень мощная программа, обладающая множеством функций. Для более детального ознакомления со всеми её возможностями необходимо ознакомиться с документацией. Она доступна в html-формате: /usr/share/synaptic/html/index.html.

## 5. Администрирование Linux

### 5.1. Начало работы с операционной системой Linux

#### 5.1.1 Варианты входа пользователя в систему

##### *Пользователи системы*

Между включением питания компьютера и моментом, когда система готова к работе с пользователем, происходит процедура загрузки системы. В Linux во время загрузки на экран выводятся диагностические сообщения о происходящих событиях, и если всё в порядке, загрузка завершится выводом на экран приглашения «**login:**».

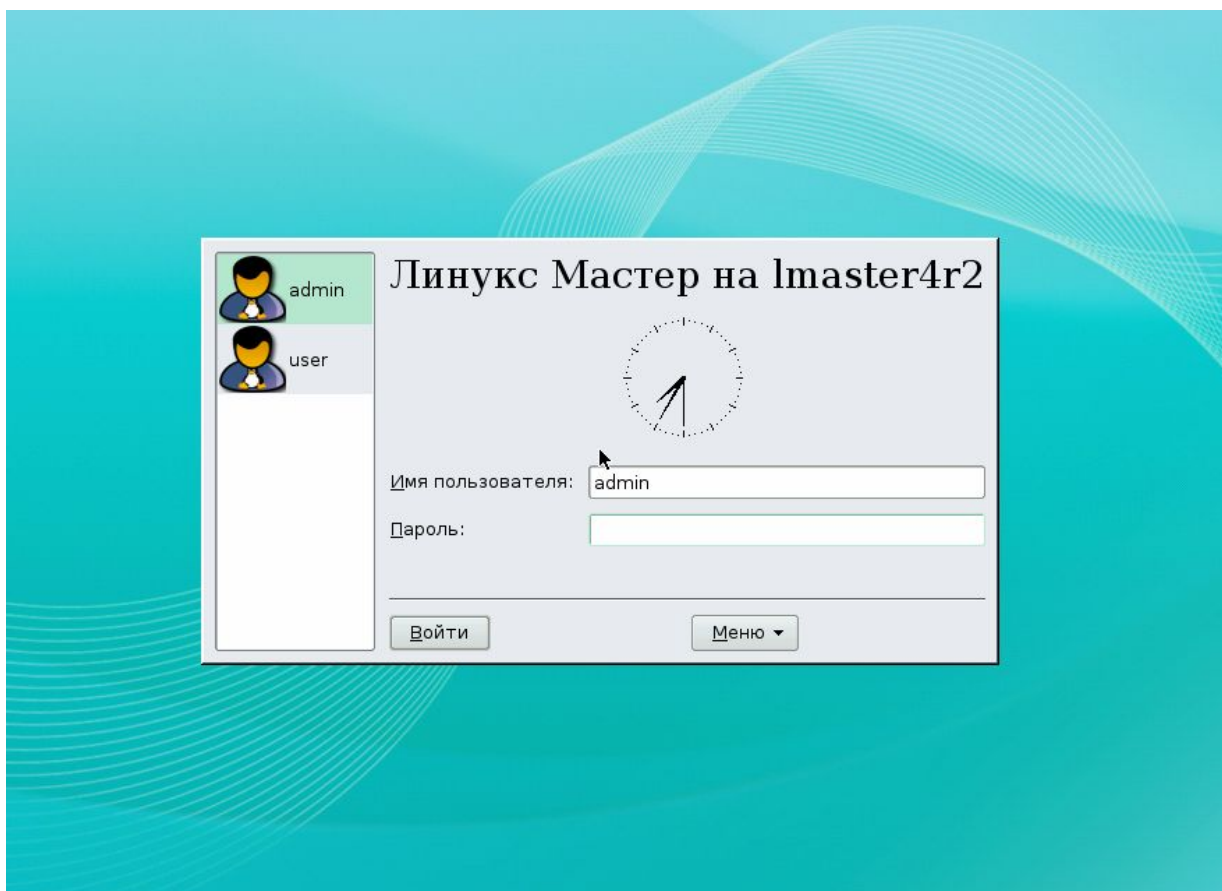


Рис. 1

Это приглашение к регистрации в системе: система ожидает, что в ответ на это будет введено системное имя пользователя, который начинает работу.

Процедура регистрации в системе обязательна. Linux – многопользовательская система, поэтому пользователь – ключевое понятие при доступе в систему. Для каждого пользователя определена сфера его полномочий в системе. Для персонального компьютера, с которым работают несколько человек, довольно важно обеспечить каждому независимую рабочую среду. Это снижает вероятность случайного повреждения чужих данных, а также позволяет каждому пользователю настроить внешний вид рабочей среды. Эта задача решается в многопользовательской модели так: организуется домашний каталог, где хранятся данные пользователя, настройки внешнего вида и поведения его системы и т. п., доступ остальных пользователей к этому каталогу ограничивается..

#### *Учётные записи*

После успешного прохождения процедуры регистрации Linux связывает пользователя с учётной записью, в которой хранится информация о каждом пользователе системы: системное имя пользователя (о нём спрашивает система, выводя приглашение «login:») и некоторые другие сведения о пользователе, необходимые системе для работы. Ниже приведён список этих сведений.

*Системное имя* - название учётной записи пользователя, которое нужно вводить при регистрации пользователя в системе. Оно может содержать только латинские буквы и знак ”\_”.

*Идентификатор пользователя* - UID (User IDentificator). UID — положительное целое число, идентифицирующее учётную запись пользователя в Linux. Обычно это число выбирается автоматически при регистрации учётной записи. Номер UID от «0» до «100» зарезервированы для псевдопользователей. Обычные пользователи имеют номера от 500. *UID* снабжены все процессы Linux и все объекты файловой системы. Используется для персонального учёта действий пользователя и определения прав доступа к другим объектам системы

*Идентификатор группы* *GUID* (*Group User IDentificator*). Кроме идентификационного номера пользователя с учётной записью связан идентификатор группы. Группы пользователей применяются для организации доступа нескольких пользователей к некоторым ресурсам. У группы, как и у пользователя, есть имя и идентификационный номер. В Linux пользователь должен принадлежать как минимум к одной первичной группе. Пользователь может входить более чем в одну группу, но в учётной записи указывается только номер первичной группы.

*Полное имя* (*GECOS*) – поле примечания, в которое обычно заносят имя и фамилию пользователя.

*Домашний каталог* - файлы всех пользователей хранятся отдельно, у каждого пользователя есть собственный домашний каталог, в котором он может хранить свои данные. Доступ других пользователей к домашнему каталогу пользователя может быть ограничен. Информация о домашнем каталоге обязательно должна присутствовать в учётной записи, потому что именно с него начинается работа пользователя, зарегистрировавшийся в системе.

*Командная оболочка (Shell)* - каждому пользователю нужно предоставить способ взаимодействовать с системой: передавать ей команды и получать её ответы. Для этой цели служит специальная программа — командная оболочка (или интерпретатор командной строки), она должна быть запущена для каждого пользователя, зарегистрировавшегося в системе. Поскольку в Linux доступно несколько разных командных оболочек, в учётной записи указано, какую из командных оболочек нужно запустить для данного пользователя. Если специально не указывать командную оболочку при создании учётной записи, она будет назначена по умолчанию `bash`.

*Интерпретатор командной строки* - программа, используемая в Linux для организации диалога человека и системы. Командный интерпретатор это: (1) редактор и анализатор команд в командной строке, (2) высокоуровневый системно-ориентированный язык программирования, (3) средство организации взаимодействия команд друг с другом и с системой.

*Понятие «суперпользователь»* - в Linux есть только один пользователь, полномочия которого в системе отличаются от полномочий остальных пользователей — это пользователь с идентификатором «0». Обычно учётная запись пользователя с `UID=0` называется `root` (англ., «корень»). Пользователь `root` — это администратор системы Linux, учётная запись для `root` обязательно присутствует в любой системе Linux. Пользователю с таким `UID` разрешено выполнять любые действия в системе. Работать в `root` следует только тогда, когда это действительно необходимо: при настройке и обновлении системы, восстановлении после сбоев. Пользователь `Root` обладает достаточными полномочиями для создания новых учётных записей.

### **5.1.2. Основы работы с командными интерпретаторами**

Необходимость работы с командной строкой в интерпретаторе вызвана тем, что администратору приходится восстанавливать систему после сбоев. Один из возможных вариантов сбоя – это отказ в работе графической среды. Один из основных принципов построения Linux заключается в том, что все конфигурационные файлы имеют текстовый формат. Это означает, что с помощью утилит, запущенных из интерпретатора, можно выявить причину сбоя и через текстовый редактор исправить поврежденный



конфигурационный файл. Если же графическая среда работает нормально, то многие действия могут быть выполнены в ней, с помощью графических утилит. Еще одна причина работать с командами – недостаточная гибкость графических утилит. В дальнейшем мы будем рассматривать приемы работы администратора как через команды интерпретатора, так и с помощью графических утилит.

Работа в Linux при помощи командной строки напоминает диалог с системой: пользователь вводит команды, получая ответы, содержащие сведения о произведённых операциях, дополнительные вопросы к пользователю, сообщения об ошибках или просто молчаливое согласие выполнить следующую команду.

К примеру при вводе команды **\$ whoami** на экран выведется

```
filipp  
[filipp@localhost filipp]$
```

Название этой команды происходит от английского выражения «Who am I?» («Кто я?»). В ответ на эту команду система вывела только одно слово: «filipp» и завершила свою работу, о чём свидетельствует вновь появившееся приглашение командной строки. Команда **whoami** возвращает название учётной записи того пользователя, от имени которого она была выполнена. Эта команда полезна в системах, в которых работает много разных пользователей, чтобы не воспользоваться по ошибке чужой учётной записью.

Команда **who** выводит список пользователей, которые в настоящий момент зарегистрированы в системе, по одной строке на каждого зарегистрированного пользователя: в первой колонке указывается имя пользователя, во второй — «точка входа» в систему, далее следует дата и время регистрации и имя хоста. Из приведенного примера можно заключить, что в системе дважды зарегистрирован пользователь **methody**: на первой виртуальной консоли (**tty1**), а примерно через сорок минут — на второй (**tty2**).

Ещё одна команда, возвращающая информацию о пользователях, работавших в системе — **last**. Выводимые этой командой строки напоминают результат команды **who**, но здесь перечислены и те пользователи, которые уже завершили работу.

*Терминал* - устройство последовательного ввода и вывода символьной информации, способное воспринимать часть символов как управляющие для редактирования ввода, послышки сигналов и т. п. Используется для взаимодействия пользователя и системы

Свойство терминала передавать только символьную информацию приводит к тому, что некоторые из передаваемых символов должны восприниматься не как текстовые, а как управляющие (например, символы, возвращаемые клавишами **Backspace** и **Enter**).

Многие из этих символов не имеют специальной клавиши на клавиатуре, поэтому их необходимо извлекать с помощью клавиатурного модификатора **Ctrl**.

Команды, подаваемые с клавиатуры с помощью **Ctrl**, принято обозначать знаком «^», после которого следует имя клавиши, нажимаемой вместе с **Ctrl**: например, одновременное нажатие **Ctrl** и «a» обозначается «^A».

Например, для завершения работы программы **cat**, которая считывает построчно данные с клавиатуры и выводит их на терминал, можно воспользоваться командой «^C» или «^D».

### 5.1.3. Справочные системы (**man**, **info**, **HowTo**, **FAQs**)

Все утилиты, все сервисы (демоны) Linux, все функции ядра и библиотек, структура большинства конфигурационных файлов и важные понятия системы описаны в руководствах, в **info**-страницах, в сопроводительной документации. Больше всего полезной информации содержится в страницах руководства (**manpages**). Для просмотра руководства используется команда **man**:

```
$ man cal
CAL(1) General Commands Manual          CAL(1)
NAME
    cal - displays a calendar
SYNOPSIS
    cal [-smjy13] [[month] year]
DESCRIPTION
    Cal displays a simple calendar. If arguments are not specified, the current month is displayed. The options are as follows:
    . . .
```

Перелистывать страницы можно клавишами **Page Up/Page Down**, для сдвига на одну строку вперёд можно применять **Enter** или стрелку вниз, а на одну строку назад — стрелку вверх. Переход на начало и конец текста выполняется по командам «g» и «G» соответственно (Go).

В системе может встретиться несколько объектов разного типа, но с одинаковым названием. Часто совпадают, например, имена системных вызовов (функций ядра) и команд, которые позволяют пользоваться этими функциями из командной строки (т. н. утилит).

```
[filipp@localhost filipp]$ whatis passwd
passwd          (1) - update a user's authentication tokens(s)
passwd          (5) - password file
passwd          (8) - manual page for passwd wrapper version 1.0.5
```

Описания объектов, выводимые **whatis**, отличаются числом в скобках — номером раздела. В системе руководств Linux девять разделов, каждый из которых содержит страницы руководства к объектам определённого типа. Все разделы содержат по одному руководству с именем «**intro**», в котором в общем виде и на примерах рассказано, что за объекты имеют отношение к данному разделу:

```
george@localhost:~> whatis intro
```

```
intro          (1) - Introduction to user commands
intro          (2) - Introduction to system calls
intro          (3) - Introduction to library functions
intro          (4) - Introduction to special files
intro          (5) - Introduction to file formats
intro          (6) - Introduction to games
intro          (7) - Introduction to conventions and miscellany section
intro          (8) - Introduction to administration and privileged
commands
intro          (9) - Introduction to kernel interface
```

1. Команды пользователя.
2. Системные вызовы (пользовательские функции ядра Linux; руководства рассчитаны на программиста, знающего язык Си).
3. Библиотечные функции
4. Внешние устройства и работа с ними (в Linux они называются специальными файлами, см. лекцию Работа с внешними устройствами).
5. Форматы различных стандартных файлов системы (например конфигурационных).
6. Игры, безделушки и прочие вещи, не имеющие системной ценности.
7. Теоретические положения, договорённости и всё, что не может быть классифицировано.
8. Инструменты администратора.
9. Интерфейс ядра (внутренние функции и структуры данных ядра Linux).

В частности, пример с **passwd** показывает, что в системе есть программа **passwd**, файл **passwd**, содержащий информацию о пользователях и администраторская программа **passwd**, обладающая более широкими возможностями. По умолчанию **man** просматривает все разделы и показывает первое найденное руководство с заданным именем. Чтобы посмотреть руководство по объекту из определённого раздела, необходимо в качестве первого параметра команды **man** указать номер раздела:

```
[filipp@localhost filipp]$ man 8 passwd
```

```
PASSWD(8)          System Administration Utilities          PASSWD(8)
```

```

. . .
[filipp@localhost filipp]$ man -a passwd
PASSWD(1)                Some Linux                PASSWD(1)
. . .
PASSWD(8)                System Administration Utilities    PASSWD(8)
. . .
PASSWD(5)                Linux Programmer's Manual          PASSWD(5)
. . .

```

Если в качестве первого параметра **man** использовать «-a», будут последовательно выданы все руководства с заданным именем. Внутри страниц руководства принято непосредственно после имени объекта ставить в круглых скобках номер раздела, в котором содержится руководство по этому объекту: man(1), less(1), passwd(5) и т. д.

Другой источник информации о Linux и составляющих его программах — справочная подсистема **info**. Документ **info** — это настоящий гипертекст, в котором множество небольших страниц объединены в дерево. В каждом разделе документа **info** всегда есть оглавление, из которого можно перейти сразу к нужному подразделу, откуда всегда можно вернуться обратно. Кроме того, info-документ можно читать и как непрерывный текст, поэтому в каждом подразделе есть ссылки на предыдущий и последующий подразделы.

```

[filipp@localhost filipp]$ info info
File: info.info, Node: Top, Next: Getting Started, Up: (dir)
Info: An Introduction
. . .
* Menu:
* Getting Started::          Getting started using an Info reader.
* Expert Info::             Info commands for experts.
* Creating an Info File::    How to make your own Info file.
* Index::                   An index of topics, commands, and variables.
. . .

```

Программа **info** использует весь экран: на большей его части она показывает текст документа, а первая и две последних строки отведены для ориентации в его структуре.

Одна или несколько страниц, которые можно перелистывать клавишей Пробел или **Page Up/Page Down** — это узел (node). Узел содержит обычный текст и меню (menu) — список ссылок на другие узлы, лежащие в дереве на более низком уровне. Ссылки внутри документа имеют вид «\* имя\_узла:» и перемещать по ним курсор можно клавишей Tab, а переходить к просмотру выбранного узла — клавишей Enter. Вернуться к предыдущему просмотренному узлу можно клавишей «l» (от «Last»). И, главное, выйти из программы

info можно, нажав «q» (Quit). Более подробную справку об управлении программой info можно в любой момент получить у самой info, нажав «?».

Узлы, составляющие документ info, можно просматривать и подряд, один за другим (с помощью команд «n», Next, и «p», Previous), однако это бывает нужно нечасто. В верхней строке экрана info показывает имя текущего узла, имя следующего узла и имя родительского (или верхнего) узла, в котором находится ссылка на текущий. Показанные имя узла Top и имя верхнего узла (dir) означают, что просматривается корневой узел документа, выше которого — только каталог со списком всех info-деревьев. В нижней части экрана расположена строка с информацией о текущем узле, а за ней — строка для ввода длинных команд (например, для поиска текста с помощью команды «/»).

Система **info** может содержать больше, чем **man**, поэтому в неё часто включают и учебники («**tutorial**»), и т. н. «**howto**» (примеры постановки и решения типовых задач), и даже статьи по теме. Таким образом info-документ может стать полным сводом сведений. Если некоторый объект системы не имеет документации ни в формате **man**, ни в формате info, в этом случае можно надеяться, что при нём есть сопроводительная документация обычно помещается в каталог **/usr/share/doc/имя\_объекта**.

### **Ключи**

Работая в системе и изучая руководства, мы заметили, что параметры команд можно отнести к двум различным категориям. Некоторые параметры имеют собственный смысл: это имена файлов, названия разделов и объектов в **man** и **info**, числа и т. п. Другие параметры собственного смысла не имеют, их значение можно истолковать, лишь зная, к какой команде они относятся. Например, параметр «**-a**» можно передать не только команде **man**, но и команде **who**, и команде **last**, при этом значить для них он будет разное. Такого рода параметры называются модификаторами выполнения или ключами (options).

```
[filipp@localhost filipp]$ date
Вск Сен 19 23:01:17 MSD 2004
[filipp@localhost filipp]$ date -u
Вск Сен 19 19:01:19 UTC 2004
```

Для решения разных задач одни и те же действия необходимо выполнять слегка по-разному. Например, для синхронизации работ в разных точках земного шара лучше использовать единое для всех время (по Гринвичу), а для организации собственного рабочего дня — местное время (с учётом сдвига по часовому поясу и разницы зимнего и летнего времени). И то, и другое время показывает команда **date**, только для работы по Гринвичу ей нужен дополнительный параметр-ключ «**-u**» (он же «**--universal**»).

## Однбуквенные ключи

```
[filipp@localhost filipp]$ who -m  
filipp tty1 Oct 10 15:50 (localhost)
```

Для **who** ключ «-m» означает «Me», то есть «Я», и в результате **who** работает похоже на **whoami**.

Свойство ключа быть, с одной стороны, предельно коротким, а с другой стороны — информативным. Ключ «-h» («Help») заставляет команды выдать краткую справку.

## Полнословные ключи

Аббревиативность ключей трудно соблюсти, когда их у команды слишком много. Некоторые буквы латинского алфавита (например, «s» или «o») используются очень часто, и могли бы служить сокращением сразу нескольких команд, а некоторые (например, «z») — редко, под них и название-то осмысленное трудно придумать. На такой случай существует другой, полнословный формат: ключ начинается на два знака «-», за которыми следует полное имя обозначаемой им сущности. Таков, например, ключ «--help» (аналог «-h»):

## *Интерпретатор командной строки (shell)*

Командных интерпретаторов в Linux несколько. Самый простой из них, появившийся в ранних версиях UNIX, назывался sh, или «Bourne Shell». Со временем его — везде, где только можно — заменили на более мощный, bash, «Bourne Again Shell». Помимо sh и bash в системе может быть установлен «The Z Shell», zsh, самый мощный на сегодняшний день командный интерпретатор или tcsh, обновлённая и тоже очень мощная версия старой оболочки «C Shell», синтаксис команд которой похож на язык программирования Си.

Какая бы задача, связанная с управлением системой, ни встала перед пользователем Linux, она должна иметь решение в терминах командного интерпретатора. Фактически, решение пользовательской задачи — это описание её на языке shell. Язык общения пользователя и командного интерпретатора — это высокоуровневый язык программирования, дополненный, с одной стороны, средствами организации взаимодействия команд и системы, а с другой стороны — средствами взаимодействия с пользователем, облегчающими и ускоряющими работу с командной строкой.

В действительности собственных команд в командном интерпретаторе немного. В основном это — операторы языка программирования и прочие средства управления самим интерпретатором. Shell занимается только тем, что подготавливает набор параметров в командной строке (например, раскрывая шаблоны), запускает программы и обрабатывает результаты их работы.

При разборе командной строки shell использует понятие разделитель (delimiter). Разделитель — это символ, разделяющий слова; таким образом командная строка — это последовательность слов (которые имеют значение) и разделителей (которые значения не имеют). Для shell разделителями являются символ пробела, символ табуляции и символ перевода строки (который всё-таки может попасть между словами способом, описанным в лекциях Работа с текстовыми данными и Возможности командной оболочки).

Вызов shell по Ctrl+Alt+FN, где N от 1 до 6 или в графической среде: **меню К - Службные-Терминал**.

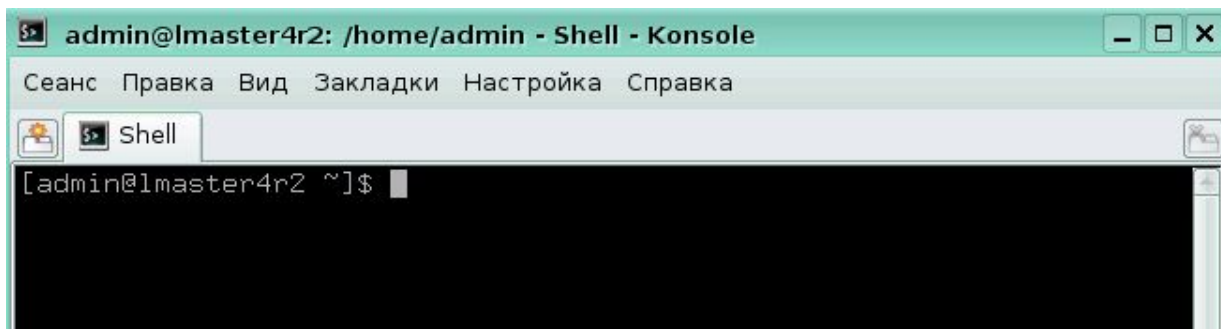


Рис. 2

#### 5.1.4. Обзор основных команд интерпретатора.

Все команды, отдаваемые пользователем при помощи shell, наследуют текущий каталог shell, т. е. «работают» в том же каталоге. По этой причине пользователю важно знать текущий каталог shell. Для этого служит утилита **pwd**:

```
[methody@localhost methody]$ pwd
/home/methody
[methody@localhost methody]$
```

pwd (аббревиатура от print working directory) возвращает полный путь текущего каталога командной оболочки. В данном случае мы узнали, что текущим является каталог «/home/methody».

Почти все утилиты, по умолчанию читают и создают файлы в текущем каталоге. Например, используем утилиту cat, чтобы вывести на экран содержимое файла «text».

```
[methody@localhost methody]$ cat text
File: info.info, Node: Help-Cross, Up: Cross-refs
The node reached by the cross reference in Info
. . .
```

*Относительный путь* - путь к объекту файловой системы, начинающийся не с корневого каталога. Для каждого процесса Linux определён текущий каталог, с которого система начинает относительный путь при выполнении файловых операций.

Относительный путь строится точно так же, как и полный — перечисление через «/» всех названий каталогов, встретившихся при движении к искомому каталогу или

файлу. Относительный путь любого файла или каталога в файловой системе может иметь любую конфигурацию: чтобы добраться до искомого файла можно двигаться как по направлению к корневому каталогу, так и от него (см. раздел Filesystem\_use.Перемещение по дереву каталогов). Linux различает полный и относительный пути: если имя объекта начинается на «/» — это полный путь, в любом другом случае — относительный.

Так, пути «text» и «./text» тоже приводят к одному и тому же файлу, однако в первом случае в строке пути не содержится ничего, кроме имени файла.

В Linux у каждого пользователя обязательно есть свой собственный каталог, который и становится текущим сразу после регистрации в системе — домашний каталог.

*Домашний каталог* - каталог, предназначенный для хранения собственных данных пользователя Linux, является текущим после регистрации пользователя в системе. Полный путь к домашнему каталогу хранится в переменной окружения HOME.

Каждый пользователь располагает своим собственным каталогом и по умолчанию работает в нём, решается задача разделения файлов разных пользователей.

Обычно доступ других пользователей к чужому домашнему каталогу ограничен.

Чтобы иметь возможность ориентироваться в файловой системе, нужно знать, что содержится в каждом каталоге. Запомнить всю структуру файловой системы невозможно и не нужно: в любой момент можно просмотреть содержимое любого каталога при помощи утилиты ls (сокращение от англ. «list» — «список»):

ls - эта команда выводит список файлов в каталоге. При простом выполнении ls выведет список файлов в текущем каталоге:

```
$ cd /
$ ls
bin  cdrom      home  mnt  sbin  usr
boot dev  lib  proc  suncd  var
cdr  etc  lost+found  root  tmp  vmlinuz
```

В таком выводе неудобно то, что вы не можете определить, где файлы, а где каталоги. Многие пользователи предпочитают, чтобы ls выводила так же идентификатор типа, что-то вроде этого:

```
$ ls -FC
bin/  cdrom/      home/      mnt/  sbin/      usr/
boot/      dev/  lib/  proc/      suncd/      var/
cdr/  etc/  lost+found/      root/      tmp/  vmlinuz
```

Эта команда добавляет слэш после имён каталогов, звезду после имён выполняемых файлов, и т.д.



ls так же может быть использована для получения более подробной информации о файлах. Например, чтобы увидеть, даты создания, имена владельцев, права доступа, вы воспользуетесь подробным списком:

```
$ ls -l
drwxr-xr-x  2   root  bin   4096  May   7    1994  bin/
drwxr-xr-x  2   root  root  4096  Feb   24   03:55  boot/
...
drwxr-xr-x 21   root  root  4096  Aug   24   1999  usr/
drwxr-xr-x 18   root  root  4096  Mar   8    03:32  var/
-rw-r-r-   1   root  root 461907      Feb  22   20:04  vmlinuz
```

Предположим, вы хотите увидеть список скрытых файлов в текущем каталоге. Для этого воспользуйтесь такой командой:

```
$ ls -a
.   bin  cdrom  home  mnt  sbin  usr
..  boot  dev    lib   proc  suncd  var
.pwrchute_tmp  cdr  etc    lost+found  root  tmp  vmlinuz
```

Файлы, имя которых начинается с точки не показываются при простом выполнении ls, поэтому называются "скрытыми". Вы увидите их только, если добавите опцию -a.

«..» — это сокращённый способ сослаться на родительский каталог

Команда **cd** используется для смены рабочего каталога. Просто наберите cd и затем имя каталога, в который вы хотите перейти. Вот несколько примеров:

```
darkstar: $ cd /bin
darkstar:/bin$ cd usr
bash: cd: usr: No such file or directory
darkstar:/bin$ cd /usr
darkstar:/usr$
```

Обратите внимание, что если вы не указываете слэш в начале пути, команда пытается переместиться в под-каталог текущего каталога.

Команда cd отличается от остальных тем, что она встроена в оболочку. Команды, встроенные в оболочку обсуждаются в разделе. Команды нет man страницы. Вместо этого вам надо обратиться к help вашей оболочки. Например:

```
$ help cd
```

Эта команда выдаст вам опции команды cd.

Необходимость вернуться в домашний каталог из произвольной точки файловой системы возникает довольно часто, поэтому командная оболочка поддерживает обозначение домашнего каталога при помощи символа «~». Поэтому чтобы перейти в домашний каталог из любого другого, достаточно выполнить команду «cd ~». При

исполнении команды символ «~» будет заменён командной оболочкой на полный путь к домашнему каталогу пользователя.

```
methody@localhost methody]$ cd ..  
[methody@localhost home]$ cd ~  
[methody@localhost methody]$ cd ~shogun  
[methody@localhost shogun]$ cd  
[methody@localhost methody]$
```

При помощи символа «~» можно ссылаться и на домашние каталоги других пользователей: «~имя пользователя». В примере \$cd.. Команда cd, поданная без параметров, эквивалента команде «cd ~» и делает текущим каталогом домашний каталог пользователя.

**more** – утилита для разбиения на страницы. Часто вывод определённой команды больше, чем размер экрана. Отдельные команды не знают, как разбить свой вывод на несколько экранов. Они предоставляют это утилите разбиения на страницы.

Команда more разбивает вывод на отдельные экраны и ждёт, пока вы нажмёте пробел до того, как выводить следующий экран. Нажатие ввода сместит экран на одну строчку вниз. Вот хороший пример:

```
$ cd /usr/bin  
$ ls -l
```

Вывод будет гораздо больше экрана. Чтобы разбить вывод на экраны, просто пропустите (pipe) команду через more:

```
$ ls -l | more
```

Это pipe символ (Shift и бэкслэш). В двух словах, pipe означает "возьми вывод команды ls и пропусти его через more". Вы можете пропускать практически всё через more, не только ls. **Less** - команда more очень удобна, но зачастую вы встретитесь с ситуацией, когда вы пропустили тот экран, который вам необходим. more не позволяет вернуться. Команда less(1) позволяет вам сделать это.

**cat** это сокращение от "concatenate" (сцеплять, связывать). Изначально была разработана для объединения нескольких текстовых файлов в один, но может быть использована и для других целей.

Чтобы объединить два или несколько файлов в один, просто перечислите файлы после cat команды и пере направьте вывод в новый файл. cat работает со стандартным вводом и выводом, поэтому вам надо воспользоваться символами оболочки для пере направления. Например:

```
$ cat file1 file2 file3 > bigfile
```

Эта команда возьмёт содержимое файлов file1, file2 и file3 и объединит их в файл bigfile.

Вы можете также использовать cat для просмотра содержимого файлов. Многие пользователи для просмотра текстовых файлов выполняют cat, имя файла, и затем пропускают вывод через more или less:

```
$ cat file1 | more
```

Так же cat часто используется для копирования файлов. Вы можете скопировать любой файл, выполнив:

```
$ cat /bin/bash > ~/mybash
```

Файл /bin/bash будет скопирован в ваш домашний каталог, под именем mybash.

**touch** используется для изменения временных атрибутов файла. С помощью этой команды вы можете изменить отметку времени доступа и отметку времени изменения файла. Если указанный файл не существует, то touch создаст пустой файл с указанным именем. Чтобы пометить файл текущим системным временем, выполните такую команду:

```
$ touch file1
```

Эта команда имеет несколько опций, включая опции для уточнения, какую временную метку изменить, какое время использовать, и ещё много других. man страница содержит описание всех этих опций.

Команда **echo** выводит указанный текст на экран. Строку, которую вы хотите вывести, следует указать после echo команды. По умолчанию echo выведет текст и символ перевода строки после него. Вы можете указать опцию -n, чтобы строка не переводилась. Опция -e укажет команде искать эскейп символы в строке и выполнить их.

**mkdir** создаёт новый каталог. Просто укажите имя каталога, который вы хотите создать после имени команды. Вот пример создания под-каталога hejaz в текущем каталоге:

```
$ mkdir hejaz
```

Так же вы можете указать путь:

```
$ mkdir /usr/local/hejaz
```

Опция -p укажет mkdir создавать "вложенные" каталоги. Так, если в выше приведённом примере каталог /usr/local не существует, то программа выдаст сообщение об ошибке. Указание параметра -p заставит программу создать /usr/local и hejaz каталоги:

```
$ mkdir -p /usr/local/hejaz
```

Каждый файл представляет собой область данных на жёстком диске компьютера или на другом носителе информации, которую можно найти по имени. В файловой системе Linux содержимое файла связывается с его именем при помощи жёстких ссылок. Создание файла с помощью любой программы означает, что будет создана жёсткая

ссылка — имя файла, и открыта новая область данных на диске. Причём количество ссылок на одну и ту же область данных (файл) не ограничено, т. е. у файла может быть несколько имён.

Пользователь Linux может добавить файлу ещё одно имя (создать ещё одну жёсткую ссылку на файл) при помощи утилиты `ln` (сокращение от англ. «link» — «соединять, связывать»). Первый параметр — это имя файла, на который нужно создать ссылку, второй — имя новой ссылки. По умолчанию ссылка будет создана в текущем каталоге.

`ln` используется для создания ссылок на файлы. Это могут быть как жёсткие - `hard`, так и гибкие или символические - `soft`, `symbolic` ссылки. Разница между двумя типами ссылок обсуждалась в разделе 5.2.3. Если вы хотите создать символическую ссылку на каталог `/var/media/mp3` и расположить ссылку в вашем домашнем каталоге, вам необходимо выполнить такую команду:

```
$ ln -s /var/media/mp3 ~/mp3
```

Опция `-s` указывает команде `ln`, что ссылка должна быть символическая. Затем задаётся цель ссылки, и в конце указывается как она будет называться. В приведённом случае будет создан файл `mp3`, указывающий на `/var/media/mp3`. Вы можете назвать ссылку как вам заблагорассудится, просто изменив последний из параметров команды `ln`.

Создание жёстких ссылок производится аналогично. Только опускается опция `-s`. Для примера, создадим жёсткую ссылку на тот же каталог, что и в примере выше:

```
$ ln /var/media/mp3 ~/mp3
```

`cp` применяется для копирования файлов. Пользователи DOS найдут её похожей на команду `copy`. Эта команда имеет очень много опций. Загляните в `man` страницу, если вам интересно.

Обычный вариант использования команды это копирование файла из одного места в другое. Например:

```
$ cp hejaz /tmp
```

Эта команда создать копию файла `hejaz` из текущего каталога в `/tmp` каталог.

Многие пользователи предпочитают сохранять временные отметки файлов при копировании. Вот вам пример:

```
$ cp -a hejaz /tmp
```

Это сохранит все временные отметки оригинального файла для копии.

Для того, чтобы рекурсивно скопировать содержимое каталога в другой каталог, вы воспользуетесь такой командой:

```
$ cp -R adirectory /tmp
```

Это скопирует `adirectory` в каталог `/tmp`.

ср имеет гораздо больше опций, которые детально описаны в man странице.

**mv** - перемещает файлы из одного места в другое. Пользователи DOS найдут её похожей на move команду. При использовании команды надо указывать источник и назначение. Вот пример обычного использования mv:

```
# mv myfile /usr/local/share/hejaz
```

mv имеет несколько опций, которые так же обсуждаются в man странице.

**rm** удаляет файлы и деревья каталогов. Аналогия для пользователей DOS - del и deltree. Использование rm может быть очень опасным, если вы не отдаёте себе отчёта в том, что вы делаете. В отличие от DOS и Windows в Linux удалённые файлы восстановлению не подлежат.

Чтобы удалить отдельный файл, укажите его имя после имени команды:

```
$ rm file1
```

Если у вас нет прав записи для этого файла, то вы получите сообщение об ошибке доступа. Чтобы удалить файл, несмотря ни на что, воспользуйтесь -f опцией:

```
$ rm -f file1
```

Чтобы целиком удалить каталог, воспользуйтесь обеими -r и -f опциями. Вот вам хороший пример того, как удалить всё содержание вашего диска. Надеемся, вы не захотите выполнить такую команду. В любом случае, вот пример команды:

```
# rm -rf /
```

Будьте очень осторожны при использовании rm; вы можете подстрелить себя в ногу. За получением дополнительных опций, обращайтесь к man страничке.

**rmdir** удаляет каталоги из файловой системы. Каталог должен быть пустым, иначе команда не сможет удалить его. Синтаксис прост:

```
$ rmdir <directory>
```

Этот пример удалит hejaz под-каталог в текущем рабочем каталоге:

```
$ rmdir hejaz
```

Если этот каталог не существует, rmdir скажет вам об этом. Вы так же можете указать полный путь к каталогу, который хотите удалить:

```
$ rmdir /tmp/hejaz
```

Этот пример попытается удалить hejaz каталог, являющийся под-каталогом /tmp каталога.

Вы так же можете удалить каталог, и все его родительские каталоги, указав -p опцию.

```
$ rmdir -p /tmp/hejaz
```

Эта команда вначале попытается удалить каталог hejaz, внутри /tmp. Если это удастся, команда попытается удалить /tmp каталог. rmdir продолжит удаление указанных каталогов, пока не столкнется с ошибкой, или не удалит все указанные каталоги.

## 5.2. Понятие системного администрирования

В современном мире компьютер стал привычным, доступным и незаменимым инструментом. Поэтому любое современное предприятие, от небольших офисов до крупных заводов, оснащено компьютерной техникой. Область применения компьютеров может быть различной, но почти всегда они объединены сетью в комплексную систему, локальную или глобальную, с возможностью выхода в Интернет. Одновременно с распространением компьютерных систем появилась необходимость эти системы обслуживать. Возникло понятие системного администрирования.

*Системное администрирование* — управление компьютерными системами, в том числе: операционными системами, программным обеспечением, базами данных, оборудованием, пользователями.

### **Основными целями системного администрирования являются:**

#### 1. Сопровождение пользователей:

- создание и удаление учётных записей пользователей;
- изменение учетных записей и паролей пользователей;
- распределение и ограничение доступа к файлам и каталогам.

#### 2. Сопровождение операционной системы:

- установка и обновление программного обеспечения;
- установка Web-, FTP-, DNS- и других серверов с последующим управлением ими; конфигурирование параметров;
- настройка производительности;
- мониторинг, диагностика и устранение неисправностей системы

#### 3. Сопровождение сети:

- создание и конфигурирование компьютерной сети;
- распределение ресурсов сети с целью минимизации нагрузок на сеть и сервера, а так же устранение узких мест;
- мониторинг, диагностика и устранение неисправностей сети

4. Обеспечение информационной безопасности (защита информации от разрушения, несанкционированного доступа, изменения и удаления):

- настройка служб безопасности;
- применение эффективной антивирусной защиты.

## 5. Обеспечение безопасности системы:

- обеспечение стабильной работы системы за счет уменьшения количества сбоев и ошибок;
- выполнение резервного копирования;
- минимизация времени на восстановление системы.

### 5.3. Система авторизации и управление правами доступа пользователей

#### 5.3.1 Средства аутентификации/авторизации пользователей Linux.

##### Учетные записи пользователей и группы

Любой файл в системе обязательно должен принадлежать пользователю. *Все устройства системы* - накопители, устройства вывода и прочая периферия - также представлены для нее в виде специальных файлов, и они также должны кому-то принадлежать. Причем, в зависимости от атрибутов каждого файла к нему могут иметь доступ либо все пользователи системы, либо только его непосредственный владелец - такая гибкая настройка позволяет избежать несанкционированного доступа к важным данным и нежелательных изменений конфигурационных файлов.

По умолчанию в любом Linux создается несколько привилегированных пользователей, которые владеют системными файлами и большинством процессов, запускаемых в системе.

Именно человек, владеющий паролем **root**, является администратором, и именно ему принадлежат, среди всех прочих, права управлять остальными пользователями в системе.

В ОС Linux очень удобно реализована поддержка пользователей. Для того чтобы человек мог выполнять какие-то действия с системой, администратор должен зарегистрировать его в системе - выдать ему имя и пароль и создать для него каталог, в котором пользователь будет оказываться сразу же после успешного входа в систему.

**Внимание.** Если приглашение командной строки оканчивается символом `<$>` или этот символ стоит перед командой, в данном учебном пособии, это означает, что последующие команды будут выполняться от имени обычного пользователя, если используется символ `<#>`, то команды выполняются от имени **root** и надо быть осторожнее.

Список пользователей можно посмотреть в файле `/etc/passwd`.

```
[admin@lmaster4r2 ~]$ cat /etc/passwd
root:x:0:0:System Administrator:/root:/bin/bash
bin:x:1:1:bin:/:/dev/null
daemon:x:2:2:daemon:/:/dev/null
adm:x:3:4:adm:/var/adm:/dev/null
lp:x:4:7:lp:/var/spool/lpd:/dev/null
mail:x:8:12:mail:/var/spool/mail:/dev/null
news:x:9:13:news:/var/spool/news:/dev/null
uucp:x:10:14:uucp:/var/spool/uucp:/dev/null
games:x:12:100:games:/usr/games:/dev/null
ftp:x:14:50:FTP User:/var/ftp:/dev/null
```

С помощью программы **getenv** также можно получить подробные сведения об учетной записи пользователя:

```
#getenv passwd root
root:x:0:1:Super-User:/:/sbin/sh
```

Список групп хранится в файле **/etc/group**:

```
[root@lmaster4r2 ~]# cat /etc/group
root:x:0:
bin:x:1:root
daemon:x:2:root
sys:x:3:root,bin,adm
adm:x:4:root
tty:x:5:
disk:x:6:root
lp:x:7:
```

Пароли пользователей хранятся в отдельных файлах в зашифрованном виде. В Линукс Мастер применяется альтернативная теневая (shadow passwords) система хранения паролей, т.е. для каждого пользователя пароль хранится в отдельном shadow-файле в каталоге **/etc/tcb/<имя\_пользователя>/shadow**.

Таким образом, в **/etc/passwd** хранится большая часть информации кроме пароля. Пароли - ключевой момент администрирования пользователей Linux. К их подбору следует подходить с тщательностью и серьезностью, потому что неверно выбранный пароль может привести к непоправимым последствиям.

### 5.3.2. Настройки операционной системы и среды пользователя.

#### Утилиты администрирования учетных записей пользователей

При добавлении нового пользователя или изменении пароля уже существующего в ОС пользователя, используется команда **passwd** или в графической среде программа **Центр управления системой**. Недостаток графической программы в том, что не всегда удается установить атрибуты пользователя такими, какие необходимы администратору.

В каждой ОС Linux обязательно имеются команды администрирования пользователей – добавления, удаления, а также изменения свойств. Команда добавления **useradd**, а команда удаления - **userdel**. В большинстве систем имеется также скрипт **adduser**, который является надстройкой над командой **useradd** и позволяет вам



производить добавление пользователей с большим комфортом за счет интерактивности. Ниже мы рассмотрим несколько примеров применения команд `useradd` и `userdel`. Простейший вариант:

```
#useradd looser
```

где *looser* - это имя пользователя. Команда выдаст максимальный идентификатор пользователя - UID, установит в качестве группы по умолчанию группу `looser`, и создаст необходимые записи в файлах `/etc/passwd` и прочих, оставив поле GECOS пустым, домашнего каталога пользователя `/home/looser`, оболочка `/bin/bash`.

Позже информацию о пользователе можно изменять с помощью специальных команд:

```
#man useradd
```

Данная команда выдаст вам подробную информацию обо всех ключах команды.

```
[root@lmaster4r2 ~]# useradd -d /home/student -s /bin/bash -c "Student" student
```

Данная строка добавляет пользователя с именем `student`, указывая в качестве его полного имени `Student`, устанавливая ему оболочку `/bin/bash`, и показывает, что его домашним каталогом будет считаться `/home/student`. Пользователь будет принадлежать группе `student`, которая является его первичной группой.

Пароль у пользователя не установлен, и потому, в зависимости от конкретной системы и ее настроек, либо пользователя при указании его имени попросят сразу же установить его пароль, либо его просто не пустят в систему (любой пароль будет считаться неверным).

Получить информацию об идентификаторе пользователя и групп, к которым пользователь принадлежит можно с помощью команды `id`.

```
[root@lmaster4r2 ~]# id user
uid=501(user) gid=100(users) groups=100(users)
```

Удалить пользователя нужно командой **`userdel`**

```
#userdel grrr
```

Команда удалит пользователя *grrr* из системы, лишив его возможности входить в систему, но оставит нетронутым его домашний каталог и все файлы в нем.

Группы перечислены в файле `/etc/group`. Каждая запись этого файла состоит из нескольких полей, разделенных двоеточиями. Это следующие поля:

- имя группы (оно должно быть уникальным)
- пароль, обычно это "x", который позволяет входить в состав группы любому пользователю; для управления доступом можно добавить пароль
- идентификатор группы, номер, которым система обозначает данную группу
- пользователи, список пользователей, относящихся к данной группе

Ниже приведен пример записи из файла `/etc/group`. Группа называется `engines`, пароля нет, идентификатор группы - 100, в группу входят пользователи `chris`, `robert`, `valerie` и `aleina`.

```
engines:x:100:chris,robert,valerie,aleina
```

Как и в случае с файлом `/etc/passwd`, файл `/etc/group` можно редактировать непосредственно, пользуясь текстовым редактором

В графической среде администрирование пользователей происходит с помощью программы **Центр управления системой**. После вызова этой программы нужно ввести пароль системного администратора `root`.

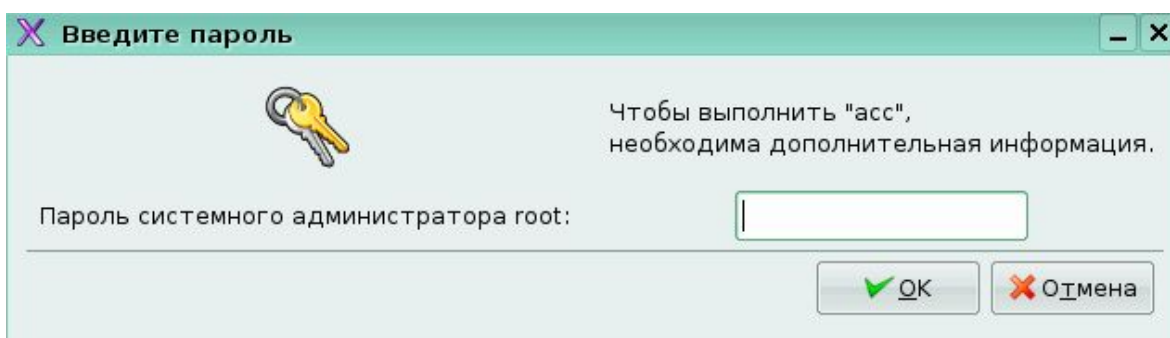


Рис. 3. Ввод пароля

После успешного ввода пароля нужно выбрать поле **Учетные записи**.

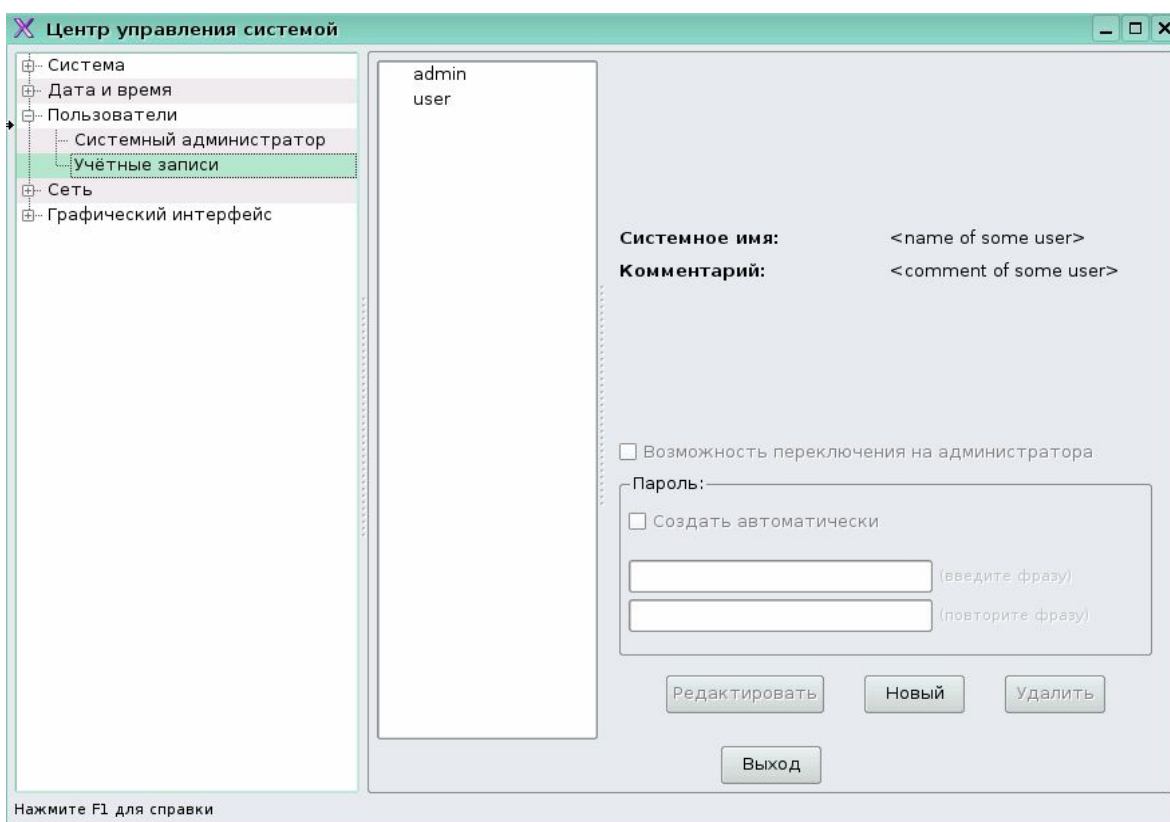


Рис. 4. Учетные записи Центра управления системой

Далее используя кнопку **Новый** – создать нового пользователя, через кнопку **Редактировать** – изменить атрибуты, а **Удалить** – удаление пользователя. Все эти действия можно проделать с пользователями, у которых номер UID 500 и больше.

Также в этой программе есть возможность изменить пароль пользователя **root**.

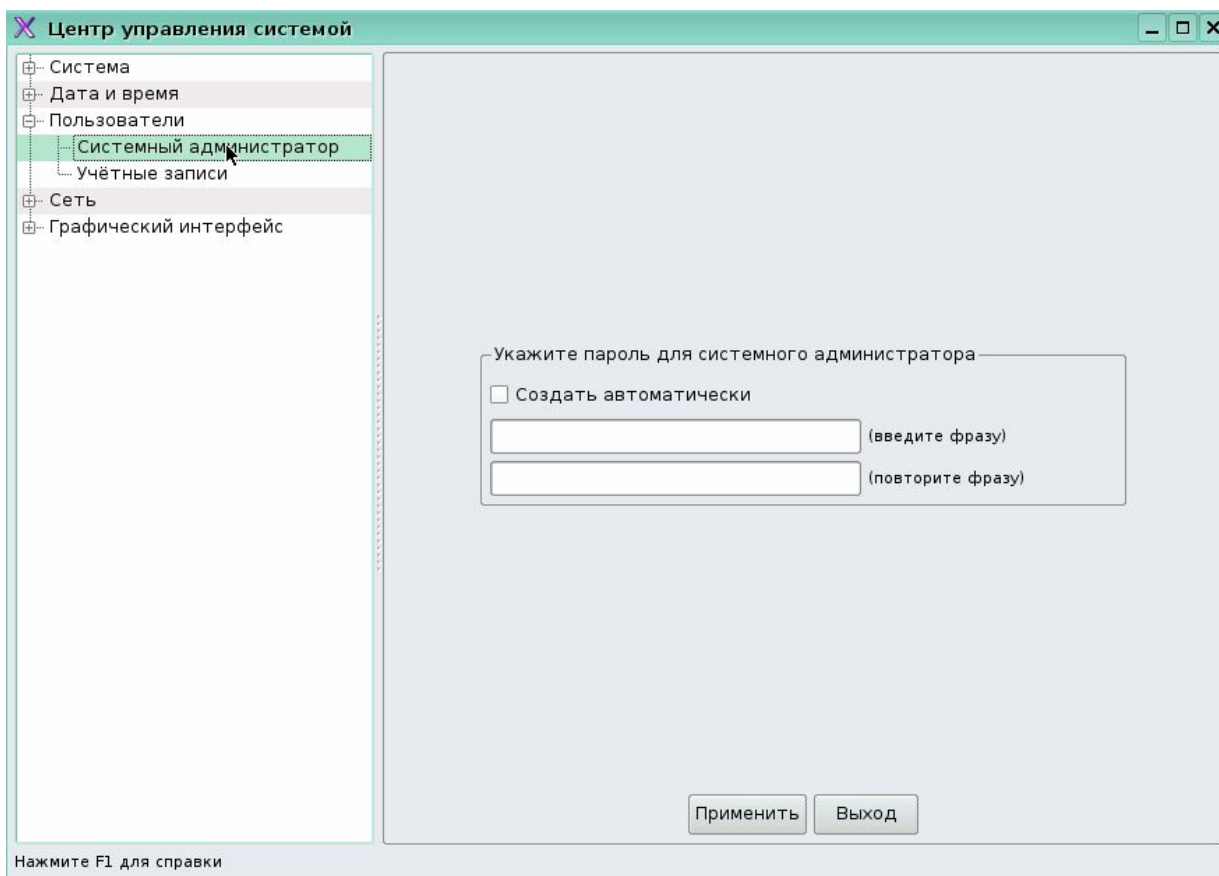


Рис. 5. Создание новой учетной записи

## Смена пароля

Пользователь может в любой момент поменять свой пароль. Единственное, что требуется для смены пароля — знать текущий пароль. Для смены пароля набрать в командной строке команду **passwd** и нажать **Enter**. В приглашении **Enter current password:** ввести текущий пароль, а при приглашениях **Enter new password** и **Re-type new password** ввести новый пароль.

```
[admin@lmaster4r2 ~]$ passwd
Changing password for admin.
Enter current password:
```

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and other characters. You can use an 8 character long password with characters from at least 3 of these 4 classes, or a 7 character long password containing characters from all the classes. An upper case letter that begins the password and a digit that ends it do not count towards the number of character classes used.

A passphrase should be of at least 3 words, 12 to 40 characters long and contain enough different characters.

Alternatively, if noone else can see your terminal now, you can pick this as your password: "bad\_sure!lord".

```
Enter new password:
Re-type new password:
passwd: all authentication tokens updated successfully.
[admin@lmaster4r2 ~]$ █
```

Пароль пользователя **root** изначально назначается при установке системы, однако он может быть изменён в любой момент точно так же, как и пароль обычного пользователя.

В графической оболочке пароль меняется программой **ChangePassword (Пуск-Прочие-Change Password)**.

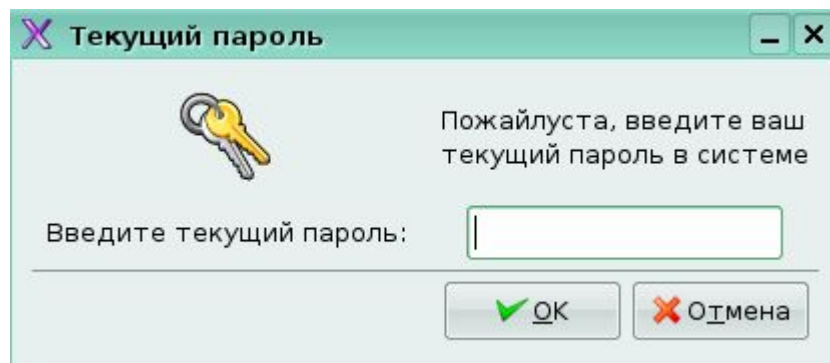


Рис. 6. Ввод текущего пароля

После запуска программы необходимо ввести текущий пароль пользователя, а затем два раза ввести новый пароль.

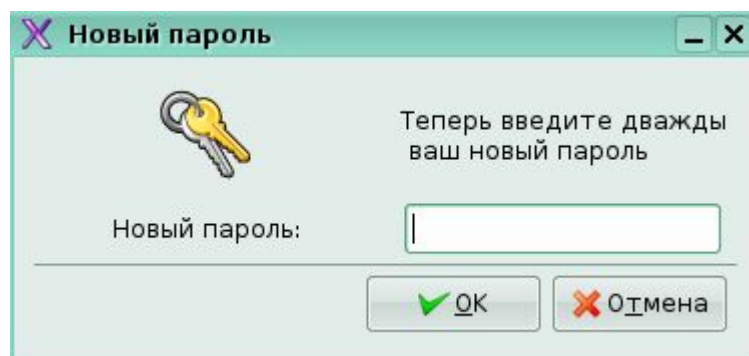


Рис. 7. Ввод нового пароля

Если пароль окажется для программы слишком простым, потребуется ввести более сложный пароль.

### 5.3.3. Управление правами доступа к файлам

*Linux* – многопользовательская система и у нее предусмотрен механизм прав доступа к файлам и директориям. Он позволяет разграничить полномочия пользователей. У файла есть владелец и группа. Пользователями *Linux* на самом деле являются не люди, а программы (процессы), выполняющие различные действия. Эти программы выполняются от имени пользователя. Возможности программ по работе с файлами и директориями определяются правами доступа, заданными для пользователя. Каждый файл имеет права доступа. Права доступа включают список из девяти пунктов (три тройки): по три вида доступа для трех групп — пользователя-владельца, группы-владельца и всех остальных. Каждый пункт в этом списке может быть либо разрешён, либо запрещён (равен 0 либо 1). Таким образом, для хранения информации о правах доступа достаточно 9 бит.

Просмотреть список файлов со всеми атрибутами можно командой `ls`:

```
[admin@lmaster4r2 ~]$ ls -l
lrwxrwxrwx 1 root root 4333 Сен  8 17:36 vmtools.txt
drwxr-xr-x 4 admin admin 4096 Сен  8 12:24 RPM
drwxr-xr-x 2 admin admin 4096 Сен  8 16:35 mnt
drwx----- 2 admin admin 4096 Сен  7 12:44 Documents
drwx----- 2 admin admin 4096 Окт  1 13:53 Desktop
[admin@lmaster4r2 ~]$
```

Права доступа указаны в первой колонке, в 3 и 4 колонках указаны имя владельца и имя группы, в последней колонке имя самого файла, остальные атрибуты пока не рассматриваем. Далее на рисунке показаны значения всех атрибутов прав доступа.



Рис. 8. Права доступа пользователей.

Запись `rw-rw-r-` означает, что владелец может данный файл читать и производить в него запись, но не может исполнять (`rw-`), группа также может читать и записывать

данные в файл (rw-). Последние 3 символа определяют права доступа для остальных пользователей. Как видно все остальные пользователи могут только читать файл.

Заметим, что выше указывалось не 9, а 10 символов. Самый первый символ примет значение d, если данный файл является каталогом. То есть запись -rw-rw-r- означает, что данный файл не является каталогом, а drw-rw-r- будет говорить, что это каталог, директория (d).

В Linux типы файлов определяются первой строкой содержимого файла. Отдельно необходимо сказать об атрибуте выполнения. Windows (DOS) запускает файлы на исполнение, если они имеют расширение "exe", Unix же судит о возможности выполнения файла по его атрибуту выполнения.

Если у файла права доступа <-----> никто ничего не сможет с ним сделать, но владелец сможет изменить права на другие, и файл снова будет выполнять своё назначение.

```
[admin@lmaster4r2 ~]$ ls -l test
----- 1 admin admin 0 Окт  7 12:21 test
```

Права доступа к файлам может изменять только владелец данного файла и суперпользователь (**root**).

### Основные команды

Изменения владельца файла: **chown**

```
$chown <имя_нового_владельца> <имя_файла>
```

Изменения группы: **chgrp**

```
$chgrp <имя_новой_группы> <имя_файла>
```

Определение прав доступа по умолчанию: **umask**.

```
$umask 022
```

Изменение прав доступа: **chmod**.

```
$chmod {augo}{+-}{rwx} <имя_файла>
```

Где *a(all)* - все, *u(user)* - владелец, *g(group)* - группа владельца, *o(other)* - остальные, кроме владельца и группы. Если мы хотим изменить права доступа одновременно и для владельца и для его группы и для всех остальных, то мы пишем не "ugo", а "a" - то есть сразу для всех.

"+" или "-" - придавать или отбирать права

*Пример.* Файлу filly установить атрибут выполнения для всех пользователей:

```
$chmod a+x filly
```

Установить право исполнения для владельца:

```
$chmod u+x filly
```

Каталоги - это тоже файлы. Права для них назначаются также как для файлов, но атрибуты прав означают другое:

- право на выполнение(x) означает, что можно входить в каталог;
- право на чтение(r) означает, что имеете право читать содержимое каталога;
- право на запись(w) означает, что вы можете добавлять файлы в каталог, удалять.

Если каталог нужно использовать совместно, необходимо разрешить запись в него либо группе пользователей, либо всем пользователям. Любой пользователь сможет удалить в нём любой файл. Во избежание этой проблемы был добавлен специальный атрибут — *sticky-bit*. При установке этого атрибута пользователь, имеющий доступ на запись в этот каталог, может изменять только *принадлежащие ему* файлы.

В Linux существует механизм *подмены идентификатора* (SetUID), позволяющий пользователям запускать процессы с идентификаторами других пользователей. Этот механизм применяется в тех случаях, когда процессу для выполнения определённых операций необходимо предоставить повышенные права (например, суперпользователя) или права другого пользователя.

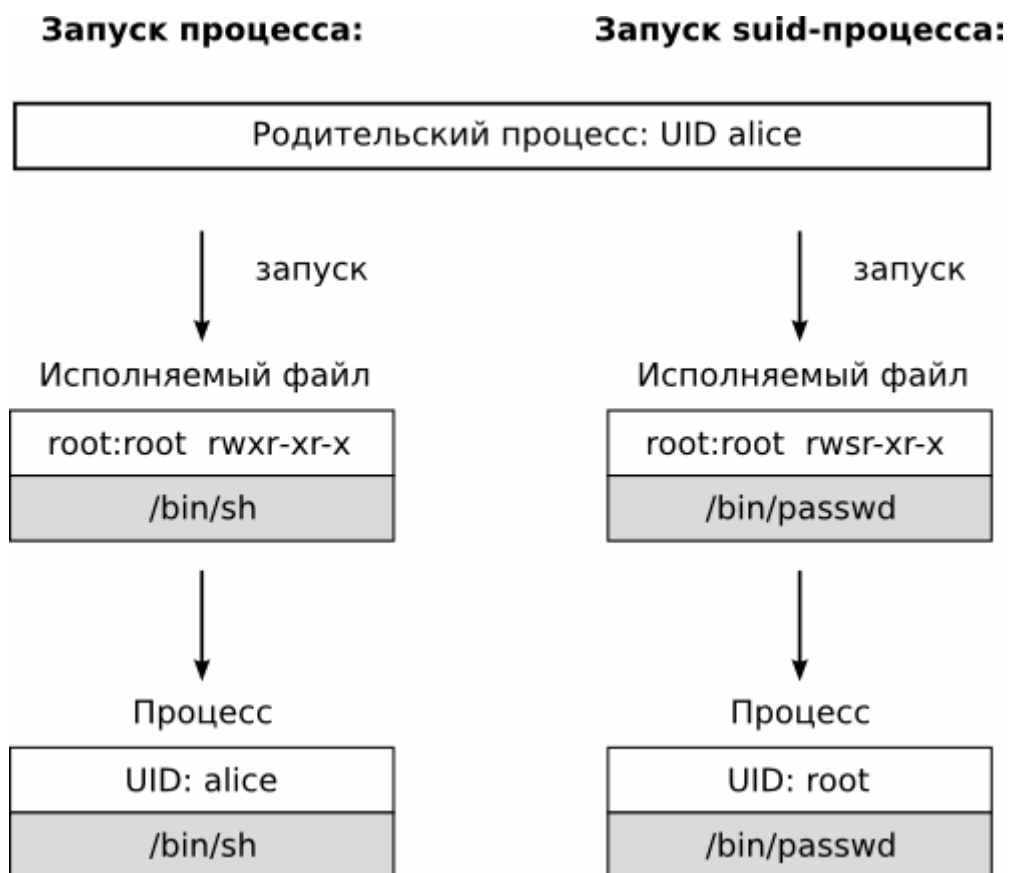


Рис.9. Механизм подмены идентификатора.

Подмена идентификатора необходима, если процесс запустит вместо себя программу из файла, в правах доступа которого установлен *бит подмены идентификатора пользователя* (SUID-бит, обозначается *s* в символьной записи прав доступа). Запущенный из этого файла процесс получит исполнительный идентификатор владельца (EUID) файла вместо идентификатора владельца процесса-родителя, благодаря чему UID процесса сохраняет информацию о том, кто на самом деле запустил программу.

Пример:

установить SUID для файла `test`:

```
chmod u+s test
```

установить SGUID для файла `test`:

```
chmod g+s test
```

Особое значение имеют атрибуты подмены идентификатора (SetUID и SetGID), установленные на каталогах. Для каталогов тоже используются права на исполнение, хотя и имеют другой смысл, чем у исполняемых файлов. Атрибут **SetGID**, установленный на каталоге, указывает, что файлы и подкаталоги, создаваемые внутри этого подкаталога любыми процессами, будут получать тот же идентификатор группы, что и сам каталог. Причем подкаталоги будут также наследовать атрибут SetGID. Такой механизм используется для организации общих каталогов, файлы в которых должны быть доступны на равных условиях группе пользователей. Атрибут **SetUID**, установленный на каталоге, просто игнорируется.

Атрибуты подмены идентификаторов пользователя и группы несут потенциальную угрозу безопасности системы и должны использоваться с осторожностью.

Такие права доступа Linux приводит к некоторым ограничениям. Стандартными средствами невозможно создать такое положение вещей, когда одна группа пользователей могла бы только читать из файла, другая — только запускать его, а всем остальным файл вообще не был бы доступен. Такое положение вещей встречается нечасто.

В Linux существует расширенные права доступа, позволяющие устанавливать права на отдельные объекты - *списки доступа* (ACL), с помощью которых можно для каждого объекта задавать права всех субъектов на доступ. Списки доступа используются редко, например, для временного понижения или предоставления доступа некоторым пользователям или при работе с очень важными файлами.

В графической среде с правами файлов можно работать через программу **Konqueror** или **mc**.



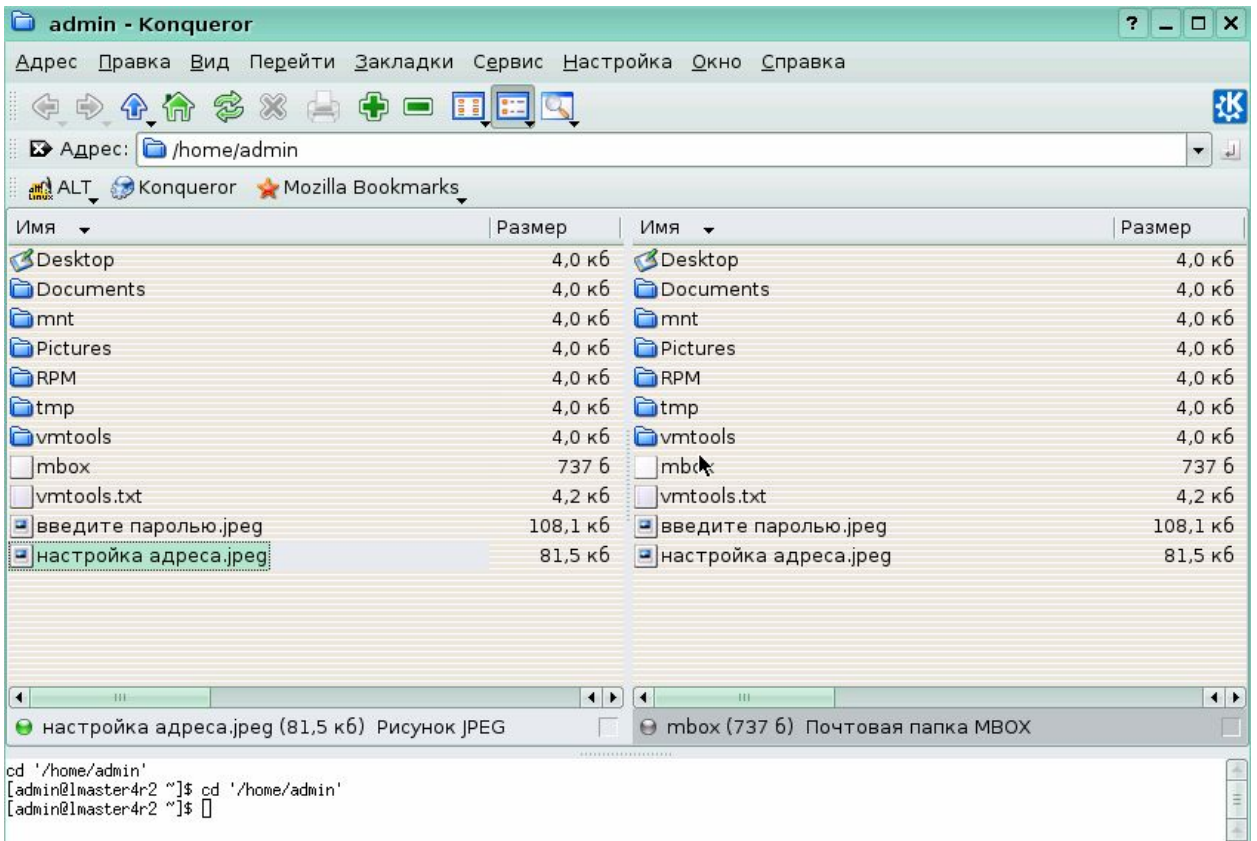


Рис. 8 Программа Konqueror

Переместиться мышкой к нужному файлу или директории и нажав правую кнопку на мышке перейти в **Свойства**, затем на закладку **Права**.

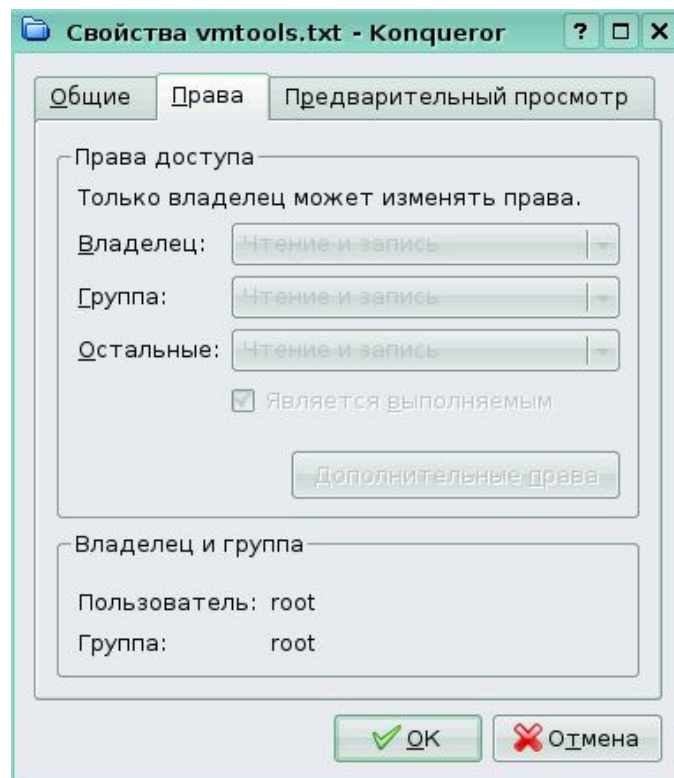


Рис. 9. Изменение права доступа

После изменения прав нажать на клавишу **ОК**.

## 5.4. Процесс загрузки системы и управление процессами

### 5.4.1. Процесс загрузки ОС

#### Загрузчик в ПЗУ

Сразу после включения, оперативная память компьютера находится в случайном, непредсказуемом состоянии. Для того чтобы начать работать, процессору необходимы специальные программы (*BIOS*). Эти программы автоматически загружаются в память из постоянного запоминающего устройства, ПЗУ (или ROM, *read-only memory*).

Операционная система Linux вынуждена пользоваться BIOS на этапе начальной загрузки, когда ей нужно прочитать образ ядра с диска или иного внешнего устройства. BIOS выполняет следующие действия:

- тесты аппаратной части компьютера для определения устройств;
- инициализация аппаратных устройств;
- поиск устройства с загружаемой ОС (в определенном порядке);
- копирование содержимого первого сектора загружаемого устройства в оперативную память и выполнение загруженного кода.

#### Загрузочный сектор и первичный загрузчик

*Загрузчик* – это программа, вызывающая процедуры BIOS для загрузки образа ОС в оперативную память. Первый раздел жесткого диска, *MBR (Master Boot Record)*, содержит таблицу разделов диска и программу для загрузки первого сектора раздела (512 байт) в котором находится образ ОС. Такой подход позволяет выбирать пользователю какую систему загружать. Для загрузки Linux нужен двух ступенчатый загрузчик. ОС Linux для архитектур 80x86 использует *LILLO (Linux Loader)* или *GRUB (GRund Unified Bootloader)*. ALTLinux предпочитает работать с LILLO, хотя GRUB более универсальный и развитый.

Основная задача первичного загрузчика LILLO загрузить вторичный загрузчик. Первичный загрузчик обычно записывается в MBR, он умеет загружать некоторые последовательности секторов с диска, указанных в карте размещения, в оперативную память. Вторичный загрузчик по объему гораздо больше первичного и может располагаться где угодно. Если менять вторичный загрузчик, надо переписать и первичный. Вторичный загрузчик должен загрузить ядро и стартовый виртуальный диск *initrd (Initial RAM disk)*, передать ядру параметры.

**Карта размещения** - это список секторов файловой системы, в которых лежат ядро и *initrd*. Карту размещения необходимо перестраивать всякий раз, когда что-то относящееся к загрузке меняется. Карта размещения, ядро и стартовый виртуальный диск лежат в директории */boot*.

## 5.4.2. Стартовый виртуальный диск

*Стартовый виртуальный диск* для загрузки Linux (initrd) - это временная корневая файловая система, которая монтируется в процессе загрузки системы в оперативную память для поддержки 2-х уровневой модели загрузки. **Initrd** состоит из различных исполняемых файлов и драйверов, которые позволяют смонтировать настоящую корневую файловую систему, после чего initrd размонтируется и освобождается память. Настройка LILO хранится в конфигурационном файле **/etc/lilo.conf**.

Простейший файл **/etc/lilo.conf**:

```
boot=/dev/hda
image=/boot/vmlinuz-up
    root=/dev/hda1
```

Первичный загрузчик записывается в начало первого жёсткого диска (строка `boot=/dev/hda`), ядро читается из файла `/boot/vmlinuz-up`, а запись `root=/dev/hda1` указывает ядру, что корневая файловая система находится на первом разделе первого диска.

*Например*, если на машине установлены MS-DOS и Linux:

```
[root@localhost root]# cat /etc/lilo.conf
boot=/dev/hda
default=linux-up
prompt
timeout=50
image=/boot/vmlinuz-up
    label=linux-up
    root=/dev/hda5
    initrd=/boot/initrd-up.img
    read-only
image=/boot/vmlinuz-up
    label=failsafe
    root=/dev/hda5
    initrd=/boot/initrd-up.img
    vga=normal
    append=" failsafe noapic nolapic acpi=off"
    read-only
other=/dev/hda1
    label=dos
other=/dev/fd0
    label=floppy
    unsafe
```

Здесь Linux был установлен на пятый раздел диска (о нумерации разделов в IBM-совместимых компьютерах рассказано в лекции Работа с внешними устройствами), а на первом находится MS-DOS. Кроме загрузки MS-DOS предусмотрено два варианта загрузки Linux и ещё один — любой операционной системы с дискеты. Каждый вариант загрузки помечен строкой `label=вариант`. При старте LILO выводит простейшее окошко, в котором перечислены все метки (в данном случае — `linux-up`, `failsafe`, `dos` и `floppy`). Пользователь с помощью стрелочек выбирает нужный ему вариант и нажимает **Enter**. При необходимости пользователь может вручную дописать несколько параметров, они

передадутся ядру системы. Если пользователь ничего не трогает, по истечении тайм-аута выбирается метка, указанная в поле **default**.

Метки **linux-up** и **failsafe** в примере используют одно и то же ядро (vmlinuz-up), но во втором случае перенастраивается режим графической карты и добавляются параметры, отключающие поддержку необязательных для загрузки аппаратных расширений (многопроцессорность, автоматическое управление электропитанием и т. п.). Строчку, стоящую после `append=` пользователь мог бы ввести и самостоятельно, это и есть параметры ядра. Поле `initrd=` указывает, в каком файле находится стартовый виртуальный диск, **unsafe** (для метки `floppy`) означает, что дискета — съёмное устройство, поэтому `lilo` не проверяет правильность её загрузочного сектора и не составляет карту. `Other=` устройство это неизвестен тип операционной системы, находящейся на этом устройстве, а значит, загрузить ядро невозможно. В первом секторе устройства будет обнаружен ещё один первичный загрузчик, `LILO` загрузит его и передаст управление по цепочке. Так и загружается MS-DOS на этой машине: первичный загрузчик берётся (по метке `dos`) из начала первого раздела первого диска.

Установка загрузчика `LILO` происходит во время начальной установки. При изменении конфигурационного файла и любых файлов участвующих в загрузке (ядро, стартовый виртуальный диск и т.д.) необходимо запустить утилиту `/sbin/lilo` чтобы перестроить карту размещения.

Все файлы необходимые для загрузки находятся в директории `/boot`.

Для использования графической оболочки необходимо запустить программу **Центр управления системой** (программа `acc`) и перейти на закладку **Настройка загрузчика**. Изменение полей в этой программе так же изменит конфигурационный файл `/etc/lilo.conf`. После выхода из программы необходимо запустить `/sbin/lilo` для перестройки карты размещения.

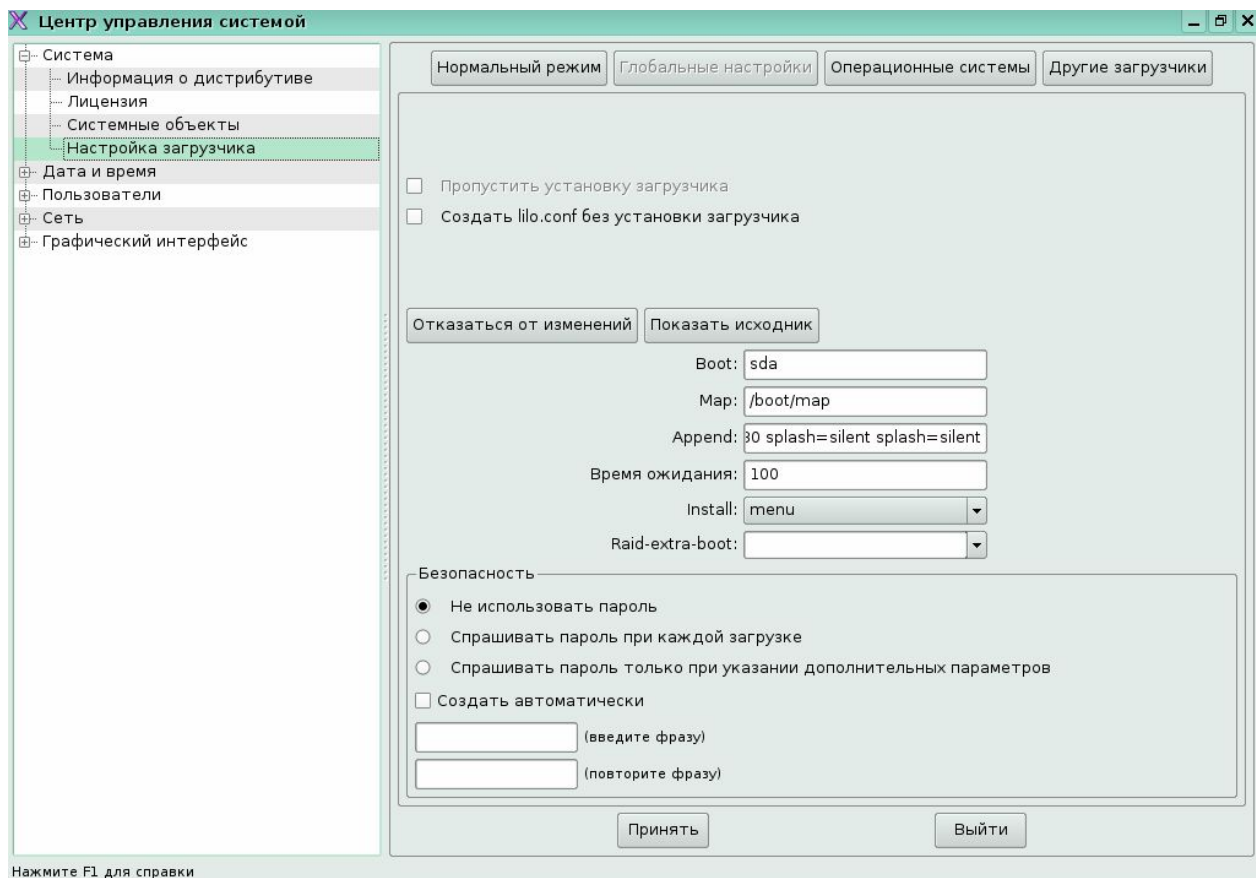


Рис 10. Настройка загрузчика в программе Центр управления системой

## Запуск системных служб

Набор утилит и системных программ, предназначенных для предоставления таких услуг как Web, FTP, почтовый сервер, отсылка заданий на печать и обеспечения их очереди, запуск заданий по расписанию, проверка целостности, принято называть подсистемами или службами. Службы запускаются во время загрузки ОС. Администратор должен настроить запуск и работу всех необходимых служб.

### Уровни выполнения

В Linux предусмотрено несколько вариантов начальной загрузки, называемых *уровни выполнения* (run levels). Уровни выполнения нумеруются с 0 до 6:

- Уровень 1 соответствует однопользовательскому режиму загрузки системы. Это сделано для того, чтобы администратор мог исправить серьёзные неполадки системы, изменить ключевые настройки.
- Уровень 2 соответствует многопользовательскому режиму загрузки системы с отключённой сетью. В этом режиме не запускаются сетевые службы, что, с одной стороны, соответствует строгим требованиям безопасности, а с другой стороны, позволяет запускать службы и настраивать сеть вручную.
- Уровень 3 соответствует многопользовательскому сетевому режиму загрузки системы без графической оболочки. Сеть при загрузке на этот уровень настроена,

и все необходимые сетевые службы запущены. На этом уровне обычно работают компьютеры-серверы. Графическую оболочку администратор может запустить вручную.

- Уровень 5 соответствует многопользовательскому графическому режиму загрузки системы. На этом уровне обычно работают рабочие станции, предоставляя пользователям возможность работать с графической подсистемой X11. Сеть на этом уровне настроена, а вот список запущенных сетевых служб может быть меньше, так как рабочая станция не всегда выполняет серверные функции.

- Уровни 0 и 6 — специальные. Они соответствуют останову и перезагрузке системы.

- Остальные уровни никак специально в Linux не описаны, однако администратор может использовать и их, определяя особый профиль работы системы. Переход с уровня на уровень происходит очень просто: по команде **init номер\_уровня**. На какой уровень загружаться при старте системы, указано в **/etc/inittab** (в поле действие должно быть написано `initdefault`, а в поле уровни — только одна цифра). Узнать текущий уровень выполнения можно с помощью команды **runlevel**:

```
[root@localhost root]# grep initdefault /etc/inittab
id:3:initdefault:
[root@localhost root]# runlevel
N 3
```

*Уровень выполнения* - сохранённый профиль загрузки системы. Старт/останов сервисов в ОС Linux реализован через выполнение всех сценариев из подкаталога **rc.уровеньd** каталога **/etc** или **/etc/rc.d**

Схема **.d** легко учитывает уровни выполнения. В каталоге **/etc/rc.d** заводится несколько **.d**-подкаталогов, соответствующих каждому уровню выполнения: **/etc/rc.d/rcуровень.d**. Именно оттуда их запускает стартовый сценарий **/etc/rc.d/rc**.

```
[root@localhost root]# ls -F /etc/rc.d
init.d/ rc.powerfail* rc0.d/ rc2.d/ rc4.d/ rc6.d/
rc*      rc.sysinit*   rc1.d/ rc3.d/ rc5.d/ scripts/
[root@localhost root]# ls /etc/rc2.d
K10power      K75netfs      S15random      S31klogd      S37gpm        S54sshd
K44rawdevices K95kudzu      S30sound       S32hotplug   S40cron      S98splash
K50xinetd     S10network   S30syslogd     S35keytable  S41anacron   S99local
[root@localhost root]# ls -l /etc/rc2.d/K75netfs
lrwxrwxrwx 1 root root 15 Nov  9 01:16 /etc/rc2.d/K75netfs -> ../init.d/netfs
```

Переход с уровня на уровень должен сопровождаться не только запуском, но и остановкой служб. Это касается не только уровней 0 и 6, но и любых других. *Например*, при переходе с уровня 3 на уровень 2 необходимо остановить все сетевые службы. Поэтому схема **.d** была расширена: сначала с параметром **stop** запускаются сценарии, имена которых начинаются на **K** (Kill), а затем, с параметром **start** — те, имена которых начинаются на **S** (Start). В приведенном примере при переходе на уровень 2 останавливаются несколько служб. Если при переходе с уровня на уровень некой службе

не требуется менять своего состояния, сценарий не запускается вовсе. Так, при переходе с уровня 3 на уровень 2 сетевые настройки остаются активными, поэтому соответствующий сценарий (S10network), скорее всего, запущен не будет.

Продолжим рассматривать процесс загрузки ОС. Поле загрузки ядра в оперативную память происходит следующее:

1. Инициализация ядра. Ядро - это основная программа, обеспечивающая работу аппаратуры. Система уже вполне работоспособна. Проверить это можно, если при загрузке указать **lilo** параметр **init=/bin/bash**.

2. Запуск процесса `init`, который родоначальником всех других процессов ОС Linux.

Основная задача `init` это запуск и останов в определенном порядке других процессов. `Init` читает конфигурационный файл **/etc/inittab**:

```
# grep rc /etc/inittab
si::sysinit:/etc/rc.d/rc.sysinit
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
# grep initdefault /etc/inittab
id:5:initdefault:
```

Метод **sysinit** в **inittab** означает, что процесс запускается во время начальной загрузки системы, до перехода на какой-нибудь уровень выполнения. Следовательно, первым запускается сценарий **/etc/rc.d/rc.sysinit**. Он настраивает аппаратуру дисковых массивов, проверяет и монтирует дисковые файловые системы, инициализирует область подкачки, межсетевой экран — словом, делает всё, без чего дальнейшая полноценная загрузка системы невозможна. Далее из строчки с `initdefault` `init` узнаёт, что уровень выполнения по умолчанию — пятый (многопользовательский графический), и выполняет все строки из `inittab`, в поле уровни которых есть 5. В частности, запускается сценарий `rc` с параметром 5 (`15:5:wait:/etc/rc.d/rc 5`), который и выполняет необходимые действия из `d`-каталога **/etc/rc.d/rc5.d**. Метод запуска `rc` — `wait`, так что `init` ждёт, пока не выполнятся все стартовые сценарии, а потом продолжает разбор `inittab`.

```
[root@localhost root]# ls /etc/rc.d/rc5.d/
K10acpid      S10network  S30syslogd  S37gpm       S50xinetd
K20nfs        S13portmap  S31klogd    S40crond     S54sshd
K65apmd       S15random   S32hotplug  S41anacron  S56rawdevices
K86nfslock    S25netfs    S35keytable S44xfs       S64power
S05kudzu      S30sound    S36update_wms S45dm        S98splash
```

Сценарий **K20nfs** (с параметром `stop`) не выполнялся: соответствующего сообщения на системной консоли не появилось. Беглый просмотр `/etc/rc.d/init.d/nfs` показал, что этот сценарий предназначен для запуска и остановки сервера сетевой файловой системы (NFS). Сервер используется на уровне 3, а на уровне 5 — нет, поэтому при переходе с 3 на 5 его следует останавливать. Поскольку во время начальной загрузки останавливать нечего, сценарий не выполнялся.

После запуска всех сценариев одного из уровней `init` запускает скрипт `/etc/rc.local`.

На этом загрузка ОС Linux завершается, пользователь видит приглашение войти в систему.

## Останов системы

Как уже говорилось, операция, обратная загрузке системы — *останов* — устроена в Linux как специальный уровень выполнения: 0 — если требуется выключить систему, и 6 — если требуется перезагрузка. Соответствующие каталоги `rc0.d` и `rc6.d` будут состоять почти сплошь из ссылок вида `K*`, но как минимум один сценарий, `killall`, будет запущен с параметром `start`. Этот сценарий остановит все процессы, которые не были остановлены `K`-сценариями: программы пользователей, демоны, запущенные администратором вручную, и т. п.

Для останова или перезагрузки системы можно выполнять команды `init 0` и `init 6`. Они оповещают и останавливают активные программы. Администраторы пользуются утилитой **shutdown**, которая запускается за несколько минут до времени перезагрузки, каждую минуту предупреждая пользователей о завершении работы системы, после чего уже запускает `init`:

```
# shutdown -r +3 "Sorry, we need to reboot"
```

### 5.4.3. Запуск и остановка системных служб

Чаще всего во время загрузки система использует уровни загрузки 3 или 5.

Например, `/etc/rc.d/rc0.d/K60crond` и `/etc/rc.d/rc3.d/S40crond`, указывающие на один скрипт `/etc/init.d/crond` службы системного журнала. Скрипт, начинающийся с "K" соответствует останову службы, а "S" - запуску. Числа, следующие перед именем службы, задают порядок запуска скриптов в директории.

Например, скрипт `/etc/rc.d/rc3.d/S34syslogd` будет запущен до скрипта `/etc/rc.d/rc3.d/S40crond`, тогда как `/etc/rc.d/rc3.d/K60crond` до `/etc/rc.d/rc3.d/K66syslogd`.



Можно заметить, что сумма чисел для одной службы равна 100 - это позволяет упорядочить все скрипты в порядке старта, обратном порядку завершения.

Где имя сервиса - это имя исполняемого скрипта находящегося в /etc/rc.d/init.d

Для управления списком служб, которые должны запускаться при старте системы, используется команда **chkconfig**.

Например, командой

```
#chkconfig sendmail on
```

запускается почтовый сервер, осуществляющий отправку и доставку почты.

Для запуска, перезапуска или останова служб в процессе работы применяется утилита **service**. Её вид следующий:

```
service имя_службы{start|stop|reload|restart|status}.
```

Например, если понадобилось перезапустить службу системного журнала, это можно сделать следующим образом:

```
# service syslogd restart
Stopping system logger service:           [ DONE ]
Starting system logger service:           [ DONE ]
```

Для управления службами необходимы права суперпользователя.

Для того чтобы скрипт автоматически загружался во время старта ОС Linux надо произвести следующую последовательность действий:

1. Создать исполняемый скрипт.
2. Разместить исполняемый скрипт в /etc/rc.d/init
3. Выполнить команду

```
#chkconfig --add <исполняемый_скрипт>
```

Для использования графической оболочки необходимо запустить программу **Центр управления системой (/usr/sbin/acc)** и перейти на закладку **Системные объекты**.

#### 5.4.4. Основные характеристики процессов и управление процессами

Выполняющаяся программа называется в Linux *процессом*. Каждый процесс имеет уникальный номер — идентификатор процесса (process identifier, *PID*). Для просмотра списка процессов можно воспользоваться утилитами **top** или **ps** (process status):

```
$ ps -f
  UID          PID  PPID  C  STIME TTY          TIME CMD
  admin    3590   1850  0  13:58 tty3          00:00:00 -bash
  admin    3624   3590  0  14:01 tty3          00:00:00 ps -f
```

Ключ **-f** (full), для получения подробной информации.

*Процесс* — это программа, выполняющаяся в оперативной памяти компьютера. Каждый процесс порожден каким-то другим процессом, то есть для каждого процесса однозначно определен его "родитель", для которого данный процесс является "дочерним". Каждый процесс имеет уникальный идентификатор процесса, *PID*. Процессы получают

доступ к ресурсам системы (оперативной памяти, файлам, внешним устройствам и т. п.) и могут изменять их содержимое. Доступ регулируется с помощью идентификатора пользователя и идентификатора группы, которые система присваивает каждому процессу.

Демоны или сервисы отличаются от обычных процессов только тем, что они работают в фоновом режиме. Если с обычным процессом всегда ассоциирован какой-то терминал или псевдотерминал, через который осуществляется взаимодействие процесса с пользователем, то демон такого терминала не имеет. Демоны обычно используются для выполнения сервисных функций, обслуживания запросов от других процессов, причем не обязательно выполняющихся на данном компьютере. Пользователь не может непосредственно управлять демонами, он может влиять на их работу, только посылая им какие-то задания, например, отправляя документ на печать.

Для того, чтобы запустить процесс в фоновом режиме из shell нужно добавить **&** в конец командной строки:

```
$ sh loop&
[1] 3634
$ ps -f
  UID          PID   PPID  C  STIME TTY          TIME CMD
  admin      3590    1850  0  13:58 tty3          00:00:00 -bash
  admin      3634    3590  99  14:03 tty3          00:00:02 sh loop
  admin      3635    3590  0  14:03 tty3          00:00:00 ps -f
```

В результате стартовый командный интерпретатор (PID 3590) оказался родителем сразу двух процессов: sh, выполняющего сценарий loop и ps.

*Активный процесс* - процесс, имеющий возможность вводить данные с терминала.

В каждый момент у каждого терминала может быть не более одного активного процесса.

*Фоновый процесс* - процесс, не имеющий возможность вводить данные с терминала. Пользователь может запустить любое, не превосходящее заранее заданного в системе, число фоновых процессов.

Чтобы посмотреть, как потребляются ресурсы системы, можно использовать утилиту top

*Сигнал* - короткое сообщение, посылаемое системой или процессом другому процессу. Обработывается асинхронно специальной подпрограммой-обработчиком. Если процесс не обрабатывает сигнал самостоятельно, это делает система.

Два сигнала — 9 (KILL) и 19 (STOP) — всегда обрабатывает система. Первый из них нужен для того, чтобы убить процесс наверняка (отсюда и название). Сигнал STOP приостанавливает процесс: в таком состоянии процесс не удаляется из таблицы процессов, но и не выполняется до тех пор, пока не получит сигнал 18 (CONT) — после чего продолжит работу.

Передавать сигналы из командной строки можно любым процессам с помощью команды **kill** - сигнал PID или просто kill PID, которая передаёт сигнал 15 (TERM).

#### 5.4.5. Регистрация событий в системе

Один из ключей к успеху в управлении любой системой - контроль за системными событиями. Linux предлагает необычный метод журналирования, а также позволяет конфигурировать составные части журналов.

В Linux журналы представляют собой обычный текст, так что вы можете исследовать и читать их, не применяя специальных средств. Вы также можете написать скрипт для просмотра журналов и автоматического выполнения каких-либо функций на основе их содержимого.

Linux-журналы хранятся в каталоге **/var/log**. Здесь содержится несколько файлов, которые поддерживаются системой. Другие сервисы и программы также могут размещать здесь свои log-файлы. Большинство журналов доступны для чтения только root.

**/var/log/dmesg** - журнал сообщений ядра;

**/var/log/messages** - основной системный журнальный файл. Он содержит сообщения о ходе загрузки системы, а также другие сообщения о статусе работающей системы. С изучения файла **/var/log/messages** обычно начинают выявление и устранение неполадок.

##### *Другие журналы*

В каталоге **/var/log** могут быть и другие log-файлы в зависимости от того, какой дистрибутив Linux вы используете, и какие сервисы и приложения запущены. *Например*, они могут быть связаны с запуском почтового сервера, распределением ресурсов, автоматическим выполнением задач и т.д.

Для работы с журнальными файлами можно использовать любой инструмент работы с текстом.

**Dmesg** - для беглого обзора журнала загрузки с целью просмотра сообщений о последней загрузке системы воспользуйтесь командой **dmesg**. Как правило, она выдает большое количество текста.

```
#dmesg
```

Команда, представленная выше, будет осуществлять постраничный вывод сообщений о ходе загрузки.

Команда **tail** предназначена для просмотра последних нескольких строк текстового файла. Если добавить опцию **-f**, tail продолжит показывать новые строки файла по мере их формирования.

```
#tail -f /var/log/messages
```

Чтобы остановить команду `tail -f`, нажмите **Ctrl + C** для прерывания процесса.

Команда **logger** позволяет посылать ваши собственные сообщения в службу журналирования.

#### 5.4.6. Обзор планировщика выполнения заданий `cron`

`Cron` - это программное обеспечение для автоматизации запуска заданий по расписанию. Конфигурирование осуществляется через команду

```
$crontab -e.
```

Вы окажетесь в текстовом редакторе **vi**, где сможете вводить текст сценария для `cron`. Краткая справка по редактору **vi**:

- для вставки текста нажмите **i**, затем вводите текст
- для удаления символов нажмите **ESC**, а затем наберите **x**
- для выхода из **vi** без сохранения изменений нажмите **ESC**, а затем наберите

**:q!**

- для сохранения и выхода нажмите **ESC**, а затем наберите **:wq**

Задания для **cron** пишутся по одному в строке. После каждой строки обязательно нужно нажать клавишу **<Enter>** — иначе задания работать не будут.

Задание для **cron** выглядит как строка, в начале которой находятся пять полей для указания периодичности задания, а далее следует команда, которую нужно запускать:

```
поле1 поле2 поле3 поле4 поле5 команда
```

Значения первых пяти полей:

минуты — число от 0 до 59

часы — число от 0 до 23

день месяца — число от 1 до 31

номер месяца в году — число от 1 до 12

день недели — число от 0 до 7 (0-Вс,1-Пн,2-Вт,3-Ср,4-Чт,5-Пт,6-Сб,7-Вс)

Для каждого конкретного параметра можно задать несколько значений через запятую. Например, если в поле "часы" написать `1,4,22`, то задание будет запущено в 1 час ночи, в 4 часа утра и в 22 часа. Можно задать интервал — `4-9` будет означать, что программу нужно запускать каждый час в период с 4 до 9 часов включительно. Символ `*` означает "все возможные значения". Например, указание `*` в поле "часы" будет означать "запускать каждый час". Символ `/` служит для указания дополнительной периодичности задания. Например, `*/3` в поле "часы" означает "каждые три часа".

*Итак*, как выглядит простейший сценарий `cron`:

```
0 */3 * * 2,5 /home/admin/script.sh
```

Скрипт `/home/admin/script.sh` будет автоматически запускаться каждые три часа во вторник и в пятницу. Введя такой сценарий в редакторе `vi` выйдите с сохранением результатов редактирования и, если вы не допустили ошибок, задание будет поставлено на выполнение с указанной периодичностью.

Посмотреть список уже установленных в `crontab` сценариев можно командой

```
$ crontab -l
```

Если файл `/etc/cron.allow` существует, любой пользователь, не являющийся суперпользователем `root`, должен быть указан в нем, чтобы иметь возможность использовать `cron`. Если такого файла нет, но есть `/etc/cron.deny`, пользователь, не являющийся суперпользователем `root` и попавший в список из этого файла, не сможет пользоваться `cron`. Если нет ни того, ни другого файла, то только суперпользователю будет позволено использовать эту команду. Пустой файл `/etc/cron.deny` (а по умолчанию он пустой) дает возможность всем пользователям работать с `cron`.

## 5.5. Структура ядра. Добавление/удаление устройств

### 5.5.1 Внутреннее устройство ядра Linux

Ядро *Linux* - это центральная часть большой и сложной операционной системы. Ядро имеет четкую структурную организацию в виде подсистем и уровней.

Операционную систему можно условно разделить на два уровня: пространство пользователя и пространство ядра.

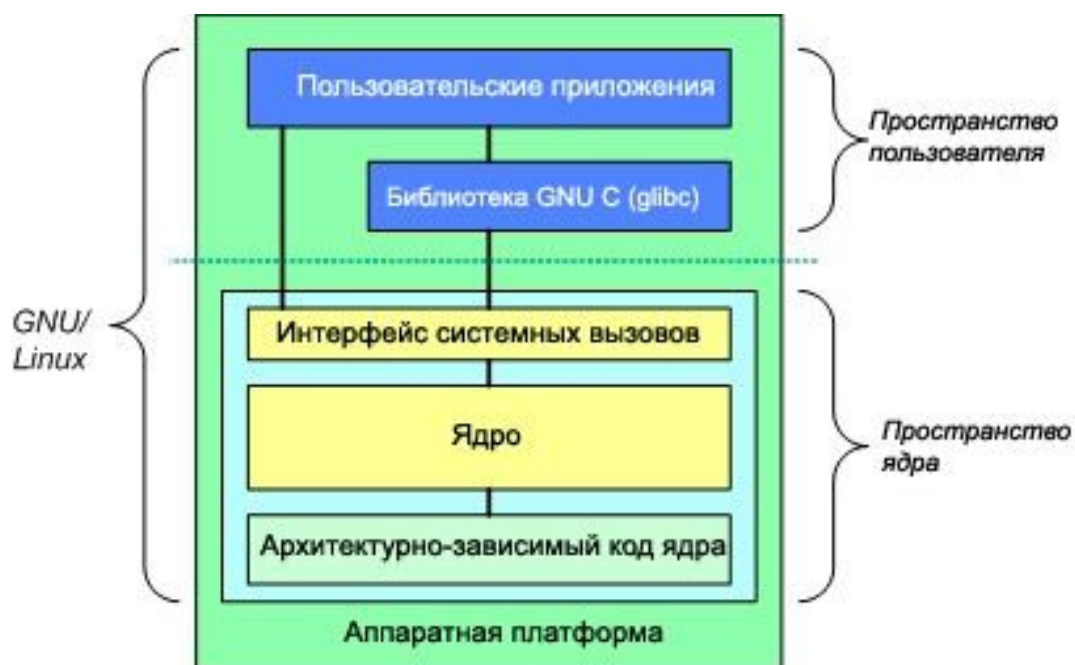


Рис 11. Уровни операционной системы.

На верхнем уровне находится пользовательское пространство (пространство приложений). Здесь исполняются приложения пользователя. Под пользовательским пространством располагается пространство ядра. Здесь функционирует ядро Linux.

Имеется также библиотека GNU C (glibc). Она предоставляет интерфейс системных вызовов, который обеспечивает связь с ядром и дает механизм для перехода от приложения, работающего в пространстве пользователя, к ядру. Это важно, поскольку ядро и пользовательское приложение располагаются в разных защищенных адресных пространствах. При этом, в то время как каждый процесс в пространстве пользователя имеет свое собственное виртуальное адресное пространство, ядро занимает одно общее адресное пространство.



Рис. 12. Архитектура ядра Linux

## Интерфейс системных вызовов

*Интерфейс системных вызовов. (SCI)* - это уровень, предоставляющий средства для вызова функций ядра из пространства пользователя.

### *Управление процессами.*

*Ядро представляет собой диспетчер ресурсов.* Независимо от того, что представляет собой управляемый ресурс - процесс, память или аппаратное устройство, - ядро организует и упорядочивает доступ к ресурсу множества конкурирующих пользователей (как в пространстве ядра, так и в пространстве пользователя).

Управление процессами сконцентрировано на исполнении процессов. В ядре эти процессы называются *потоками* (threads); они соответствуют отдельным виртуализованным объектам процессора (код потока, данные, стек, процессорные регистры). В пространстве пользователя обычно используется термин процесс, хотя в реализации Linux эти две концепции (процессы и потоки) не различают. Ядро предоставляет интерфейс программирования приложений (API) через SCI для создания нового процесса (порождения копии, запуска на исполнение, вызова функций, остановки процесса (kill, exit), взаимодействия и синхронизации между процессами).

## Управление памятью

Другой важный ресурс, которым управляет ядро - это *память*. Для повышения эффективности память организуется в виде страниц (в большинстве архитектур размером 4 КБ). В ОС Linux имеются средства для управления имеющейся памятью, а также аппаратными механизмами для установления соответствия между физической и виртуальной памятью.

При нехватке памяти страницы можно удалять и переносить на диск. Этот процесс обмена страниц между оперативной памятью и жестким диском называется подкачкой.

## Виртуальная файловая система.

*Виртуальная файловая система (VFS)* является общим интерфейсом ко всем стандартным для ОС Linux файловым системам.

Файловые системы, поддерживаемые VFS делятся на три группы:

1. Дисковые файловые системы (ext2, ext3, ext4, reiserfs, minix и т.д.).
2. Сетевые файловые системы (nfs, cifs и т.д.)
3. Специальные файловые системы (/proc).

На верхнем уровне VFS располагается единый API-интерфейс таких функций, как открытие, закрытие, чтение и запись файлов. На нижнем уровне VFS находятся API-вызовы стандартных файловых систем, которые определяют, как реализуются функции верхнего уровня. Они представляют собой подключаемые модули для конкретных файловых систем.



Рис. 13. Виртуальная файловая система

### Сетевой стек.

Сетевой стек по своей конструкции имеет многоуровневую архитектуру, повторяющую структуру протоколов. *Протокол Internet Protocol (IP)* - это базовый протокол сетевого уровня, располагающийся ниже транспортного протокола Transmission Control Protocol, TCP). Выше TCP находится уровень сокетов, вызываемый через SCI.

Уровень сокетов представляет собой стандартный API к сетевой подсистеме. Он предоставляет пользовательский интерфейс к различным сетевым протоколам. Уровень сокетов реализует стандартизованный способ управления соединениями и передачи данных между конечными точками, от доступа к "чистым" кадрам данных и блокам данных протокола IP (PDU) и до протоколов TCP и User Datagram Protocol (UDP).

### Драйверы устройств

Ядро взаимодействует с устройствами ввода/вывода посредством драйверов. Драйверы содержат структуры данных и функции, управляющие одним или несколькими устройствами, такими как жесткие диски, клавиатуры, мыши, сетевые карты, мониторы и т.д.

### Архитектурно-зависимый код.

Хотя основная часть Linux независима от архитектуры, на которой работает операционная система, в некоторых элементах для обеспечения нормальной работы и повышения эффективности необходимо учитывать архитектуру.

### 5.5.2. Создание файлов устройств. Установка параметров ядра и модулей при загрузке

*Модули ядра* - это другое название драйверов устройств, которые могут быть вставлены в запущенное ядро. Они позволяют расширить список аппаратных устройств, поддерживаемых ядром, без установки другого ядра, или перекомпиляции существующего ядра.

Модули ядра могут быть загружены или выгружены в любое время, даже во время работы системы. Это позволяет системным администраторам обновлять драйвера устройств, не прерывая работу сервера.

Модули хранятся в каталоге `/lib/modules/<kernel version>` вашей системы. Они могут быть загружены во время загрузки компьютера из `rc.modules` файла. Версию ядра можно посмотреть командой:

```
#uname -r
```

На экран выведется:

```
2.6.18-std-smp-alt12.M40.1
```

Для просмотра загруженных модулей воспользуйтесь командой **lsmod**:



```

#lsmod
Module Size      Used by
parport_pc 7220          0
parport     7844          0 [parport_pc]
scsi_mod    137096       5 sg,mptspi,mptscsih,scsi_transport_spi,sd_mod
ide_disk    20352         0

```

Для выгрузки модуля используйте команду **rmmmod**.

Пример выгрузки модуля **vmhgfs**:

```
#rmmode vmhgfs
```

Модули могут быть загружены командами **modprobe** или **insmod**.

Утилита **Modprobe** безопаснее, так как автоматически загрузит еще и все зависимые модули.

Команда **modinfo** показывает информацию о модуле:

```

#modinfo parport
filename:      /lib/modules/2.6.18-std-smp-alt12.M40.1/kernel/drivers/parport/parport.ko
license:      GPL
srcversion:    FDA3D21E571EC3B4AE6FB8A
depends:
vermagic:     2.6.18-std-smp-alt12.M40.1 SMP mod_unload 586 REGPARM gcc-4.1

```

## Виртуальная файловая система /proc.

*Файловая система /proc* является механизмом для ядра и его модулей, позволяющим посылать информацию процессам. С помощью этой виртуальной файловой системы можно работать с внутренними структурами ядра, получать полезную информацию о процессах и изменять установки (меняя параметры ядра) на лету. Файловая система /proc располагается в памяти в отличие от других файловых систем, которые располагаются на диске или в сети.

Перечислим некоторые ключевые файлы директории **/proc**:

- **/proc/cpuinfo** - информация о процессоре ( модель, семейство, размер кэша и т.д.)
- **/proc/meminfo** - информация о RAM, размере свопа и т.д.
- **/proc/mounts** - список подмонтированных файловых систем.
- **/proc/devices** - список устройств.
- **/proc/filesystems** - поддерживаемые файловые системы.
- **/proc/modules** - список загружаемых модулей.
- **/proc/version** - версия ядра.
- **/proc/cmdline** - список параметров, передаваемых ядру при загрузке.

Изменяя содержимое некоторых файлов в директории /proc меняем настройки или параметры ядра.

Команда

```
#echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

запрещает системе отвечать на команду ping и тем самым как бы спрячет компьютер в сети. Команда **ping** посылает ICMP (Internet Control Message Protocol) эхо-запрос на указанный интерфейс в сети и в ответ ожидает получения ICMP эхо-отклика.

Во время загрузки ОС Linux происходит обращение к файлу `/etc/sysctl.conf`, который содержит настройки некоторых параметров ядра. При изменении параметров ядра нужно пользоваться этим конфигурационным файлом.

## **5.6. Система печати.**

### **5.6.1. Организация системы печати ОС Linux. Возможности систем печати**

В ОС Linux у пользователя есть несколько возможностей, чтобы настроить систему печати. Проще всего запустить программу **kdeprint**, входящую в состав графической среды KDE. Она обладает дружелюбным графическим интерфейсом и в большинстве случаев правильно определяет параметры принтера. Другой путь — настроить систему при помощи **CUPS** или **foomatic**.

Каждый принтер известен в системе под определённым именем. Перед отправкой документа на печать пользователь может выбрать имя принтера, на котором следует распечатывать. Если специально ничего не указывать, задание на печать будет отправлено на принтер по умолчанию.

Настроенный принтер может находиться в состоянии отключён — в этом случае все отправленные на печать задания так и останутся в очереди, пока принтер не будет включён снова. Проверьте, отмечен ли пункт принтер включён для тех принтеров, которые вы собираетесь использовать.

Для установки и удаления принтеров предназначен **alterator-printers** — модуль настройки, который можно вызвать командой **config-printers** или из меню **Настройка—ALT Linux—Принтеры**, он также доступен в качестве одного из модулей АСС (ALT Linux Control Center).

**Примечание.** Основное средство настройки ALT Linux — Центр управления ALT Linux (ALT Linux Control Center). Он позволяет в графическом интерфейсе управлять наиболее востребованными настройками системы: пользователями, сетевыми подключениями, периферийным оборудованием, правами доступа и т. п. ALT Linux Control Center состоит из нескольких независимых диалогов — модулей. Каждый модуль отвечает за настройку определённой функции или свойства системы и может быть запущен автономно.

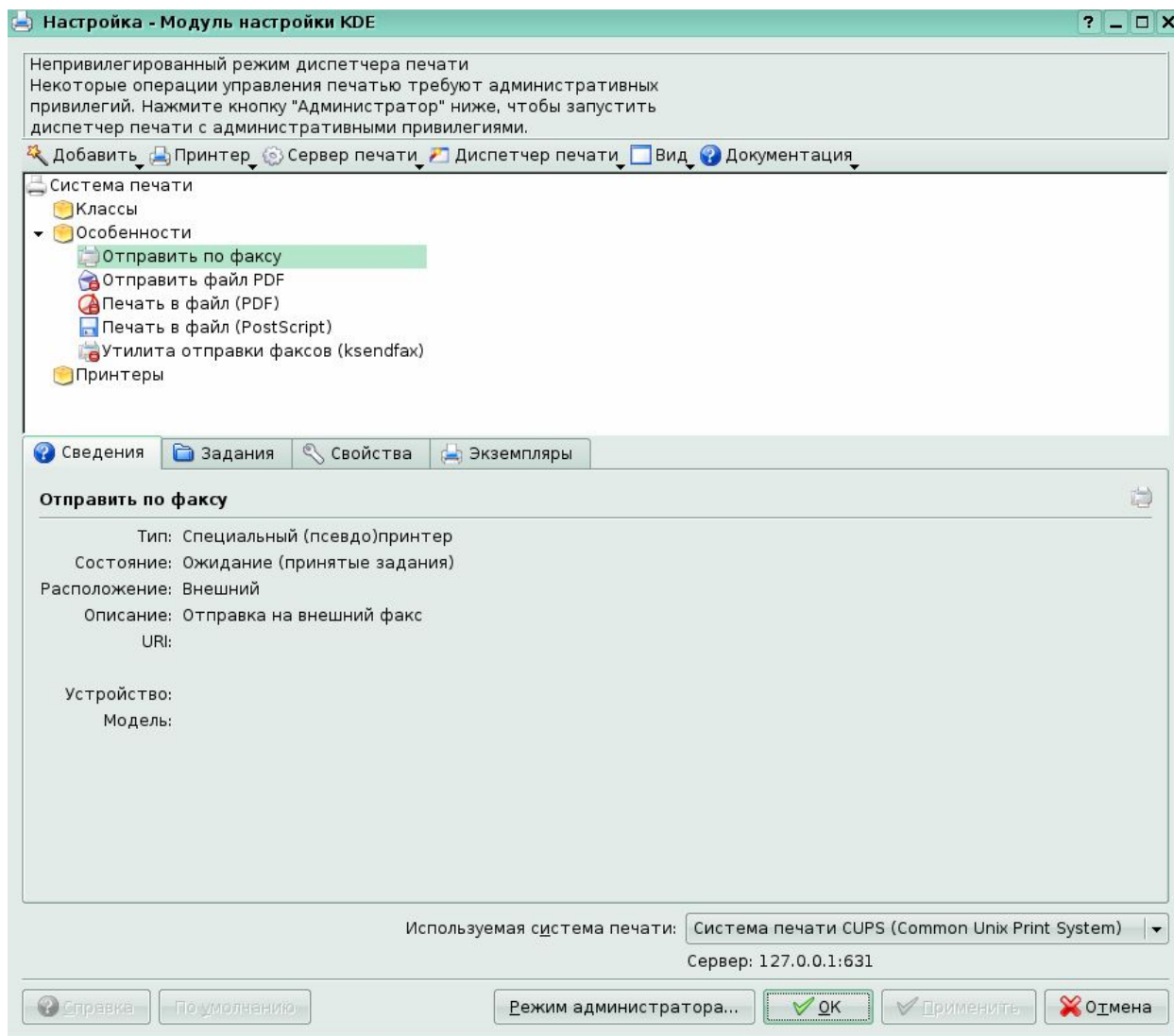


Рис. 14. Настройка KDE

Управлять свойствами принтеров (разрешение при печати, формат бумаги, цветовые гаммы и т. п.) можно двумя способами:

- через web-интерфейс — чтобы его открыть, наберите в браузере адрес <http://localhost:631>.
- изменяя конфигурационные файлы, подробнее см. раздел Подсистема печати (CUPS).

### Добавление принтера

Для каждого добавляемого принтера нужно выбрать имя, под которым он будет известен в системе. Имя представляет собой слово, состоящее только из латинских букв (цифры и знаки препинания недопустимы). Имя принтера — это условность, которая никак не связана с моделью принтера, выбирайте любые имена, которые будут для вас удобны.

Один и тот же принтер можно добавить под разными именами. Этот трюк обычно используется для печати на одном и том же принтере с разными настройками. Например,

можно отправлять на принтер **fine** документы, которые требуют качественной печати, а на принтер **draft** — черновики.

### Способ соединения принтера

Документы можно распечатывать как на принтере, подключённом непосредственно к компьютеру, так и на принтере, подключённом к серверу, доступному по локальной сети. Есть две технологии доступа к принтеру на удалённом сервере: *SMB* (*Samba*) и *CUPS*. Если в локальной сети имеются узлы с операционными системами семейства Windows, то для доступа к принтеру используется SMB. CUPS используется в операционных системах семейства UNIX/Linux.

Для добавления локального принтера, то нужно сначала проверить, что он подключён к компьютеру и включён — в этом случае будет возможность автоматически определить модель принтера.

### Добавление локального принтера

Если подключённый принтер удалось определить автоматически, название модели появится в списке. Если принтер определился неправильно — модель можно изменить вручную. Принтеры, подключённые к параллельному порту (LPT) не всегда поддаются автоматическому определению, в этом случае среди доступных устройств будут перечислены названия портов, к которым может быть подключён принтер (parport0).

### Принтер на удалённом сервере SMB

Сведения, которые здесь нужно ввести, относятся к свойствам самого сетевого принтера, их можно выяснить у администратора локальной сети. SMB-имена следует вводить без начальных символов /. Пользователя, пароль и рабочую группу нужно указывать только в том случае, если доступ к принтеру ограничен и требует указания этих параметров. Если принтер доступен для всех, эти поля следует оставить пустыми.

### Выбор модели принтера

Следующий шаг при добавлении принтера — выбор производителя и модели. Если в списке не нашлось модели принтера, можно выбрать универсальный драйвер, в качестве производителя выберите **Generic**.

### 5.6.2. Настройка системы печати CUPS

В ОС Linux для печати существует *система CUPS (Common UNIX Printing System)*. Принтер должен быть подключен к системе CUPS, и она должна быть запущена. Сделать это можно с помощью программы **printerdrake** или программы конфигурации принтера из KDE (Меню **KDE - Прочие - Настройка - Принтеры**). После того как принтер настроен печать можно производить обычным образом. Например, в OpenWriter: **Файл - Печать**

Нужно запустить web-браузер с URL-адресом <http://localhost:631>.

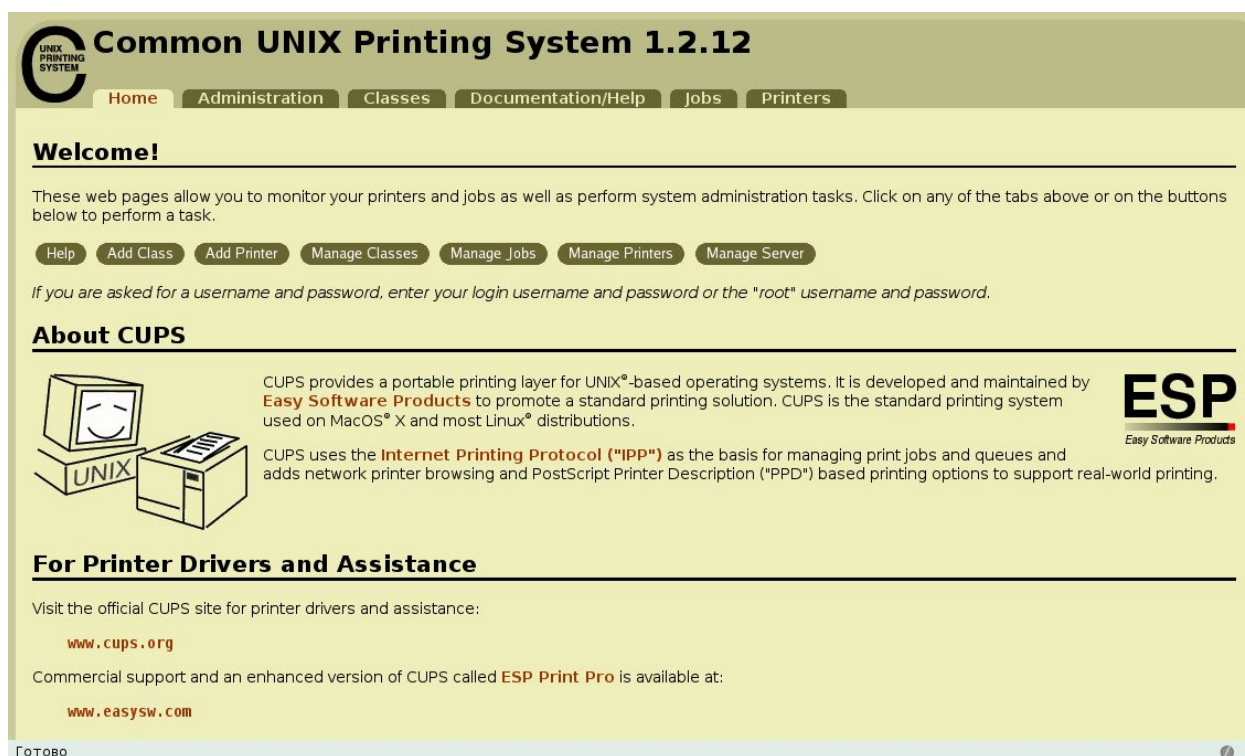


Рис. 15 Web-страница настройки CUPS

Все CUPS системы общаются между собой; сервер узнает, что уже есть сосед с настроенным принтером и посылает задание сразу на него.

После открытия web-страницы CUPS выбрать раздел **Manage Printers**, далее раздел **Add Printer**. Затем вводите имя и пароль администратора CUPS; по умолчанию это администратор системы (root).

Появится первый диалог, в котором нужно указать имя (Name) принтера и описание (Description). В следующем диалоге выбираем устройство, к которому подключён принтер. Затем – устанавливаем модель и производителя принтера.

Если производителя или модели принтера нет в списке CUPS, нужно его завести..

*Описание принтеров для CUPS* — это PPD-файлы в каталоге `/usr/share/cups/model`. Нужно найти PPD-файл для нужной модели принтера и поместить в `/usr/share/cups/model`. База принтеров есть также в **foomatic**, если нужной модели не нашлось и там, стоит поискать PPD-файл на сайте производителя принтера.

*Настройка CUPS через foomatic.*

Пример. Для настройки принтера HP LaserJet 1022, которого не нашлось в базе принтеров CUPS. Нужно найти описание для данной модели принтера в базе данных **foomatic**. Это делается с помощью команды:

```
$ foomatic-ppdfile -A|grep 1022
HP LaserJet 1022 Id='HP-LaserJet_1022' Driver='hpijs'
```

```
Ricoh Aficio 1022 PS Id='Ricoh-Aficio_1022_PS'  
Driver='Postscript'
```

Чтобы не просматривать огромный список принтеров, лучше всего профильтровать вывод **foomatic-ppdfile -A** утилитой **grep**, задав в качестве поискового выражения название/номер модели принтера.

Нас прежде всего интересует идентификатор-описание в поле Id. Кроме того, мы получили имя ghostscript-драйвера, с которым будет работать данный принтер.

Сгенерируем ppd-файл:

```
$ foomatic-ppdfile -p 'HP-LaserJet_1022' > HP-LaserJet_1022.ppd
```

Поместим этот файл в каталог **/usr/share/cups/model** и перезапустим службу CUPS.

```
#service cups restart
```

После этого возвращаться к web-интерфейсу CUPS, где появится нужная модель, и настраивать принтер привычным способом.

*Как удалить лишние принтеры?* Удаление можно производить вручную, удалив описания из конфигурационных файлов спулера **/etc/cups/printers.conf**. Для CUPS можно вновь воспользоваться web-интерфейсом. Если использовать foomatic, нужно запустить команду

```
# foomatic-configure -s тип_спулера -n имя_принтера -R
```

где тип\_спулера — CUPS.

Конфигурационные файлы CUPS находятся в каталоге **/etc/cups**. Файл **/etc/cups/cupsd.conf** содержит описание главных параметров сервера. Наиболее часто используемые параметры:

**LogLevel.** Уровень подробности протоколирования. По умолчанию значение равно info.

**Port.** TCP-порт, по которому сервер будет ожидать обращений клиентов. По умолчанию это 631 (зарезервированное для протокола IPP).

**Location, Order, Allow, Deny.** Серия директив, аналогичных имеющимся в web-сервере Apache. Order — порядок просмотра значений Allow и Deny, Allow — адреса, с которых разрешён доступ, Deny — адреса, с которых доступ запрещён. Пример:

```
<Location />  
Order Deny,Allow  
Deny From All  
Allow From 127.0.0.1  
</Location>
```

Это означает, что сначала сервер будет смотреть, запрещён ли доступ с машины (здесь запрещено для всех), а потом — разрешён ли (здесь разрешён доступ только с

самого сервера). Последний параметр понадобится для организации одного CUPS-сервера на всю локальную сеть.

Файл `/etc/cups/client.conf` содержит настройки для клиентской части. В нём указываются два параметра — местоположение сервера и защищённость соединения.

Файл `/etc/cups/printers.conf` содержит описание принтеров. Пример настройки локального принтера.

```
<DefaultPrinter printer>
DeviceURI parallel:/dev/lp0
State Idle
Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
```

Параметр **Accepting** отвечает за то, принимает ли принтер задания. Параметр **DeviceURI** – это описание местоположения принтера. Описание каждого установленного принтера (соответствующий PPD-файл) находится в `/etc/cups/ppd/имя_принтера.ppd`. *Имя\_принтера* — это имя, под которым установлен принтер. PPD-файл это конфигурационный файл, в котором описаны свойства принтера.

При каждом изменении в конфигурационных файлах CUPS следует перезапускать системную службу **cups** командой

```
#service cups restart
```

## 5.7. Файловая система Linux - возможности и использование.

### 5.7.1. Типы файловых систем. Особенности и преимущества

#### использования файловых систем. Структура файловой системы

Стандартные файловые системы для ОС Linux: ext2, ext3, ext4, ReiserFS, Minix JFS, XFS и т.д. (более 50). Благодаря виртуальной файловой системе VFS перенос “чужих” файловых систем (NTFS, FAT16, FAT32 и т.д.) на Linux осуществляется проще, чем в других ОС.

#### Файловая система ext2

Вторая расширенная файловая система EXT2 является родной для Linux. Основным недостатком этой файловой системы является отсутствие журналирования, необходимое для всех высоко загруженных серверов. Развитием ext2 стала журналируемая файловая система ext3, полностью совместимая с ext2.

### *Свойства ext2:*

- при создании файловой системы можно выбрать размера блока (1024 – 4096 байт)
- выбор количества индексных дескрипторов
- объединение блоков в группы
- предварительное выделение блоков данных под обычные файлы.
- поддержка быстрых символьных ссылок
- аккуратная реализация обновления файла
- автоматическая проверка согласованности файловой системы
- поддержка неизменяемых файлов и дополняемых файлов
- совместимость с семантикой SVR4.

Все типы файлов имеют символьные имена. Ограничения на простое имя состоят в том что, его длина не должна превышать 255 символов, в имени не должны присутствовать символ NULL и '/'. Полное имя представляет собой цепочку простых символьных имен всех каталогов, через которые проходит путь от корня до файла. В файловой системе ext2 файл может входить в несколько каталогов, а значит, иметь несколько полных имен.

### *Атрибутами файла в системе ext2 являются:*

- тип и права доступа к файлу;
- владелец, группа;
- информация о разрешённых операциях доступа к файлу;
- время создания, последнего доступа, последнего изменения и время последнего удаления;
- текущий размер файла;
- тип файла;
- число блоков, занимаемых файлом;
- ACL
- другие

Атрибуты файлов хранятся не в каталогах, а в специальных таблицах. Каталог имеет простую структуру, состоящую из двух частей: номера индексного дескриптора и имени файла.

### **Физическая организация файловой системы ext2.**

*В составе ext2 можно выделить следующие составляющие:*

- блоки и группы блоков;
- индексный дескриптор;



- суперблок;

Всё пространство раздела диска разбивается на блоки фиксированного размера, Размер блока указывается при создании файловой системы. Все блоки имеют порядковые номера. Блоки объединяются в группы блоков.

Индексный дескриптор **inode** (information node) - это таблица, которая содержит информацию об атрибутах и физическом расположении файла.

*Суперблок содержит общую информацию о файловой системе:*

- общее число блоков и индексных дескрипторов в файловой системе;
- статус файловой системы;
- число свободных блоков и индексных дескрипторов в файловой системе;
- размер блока файловой системы;
- количество блоков и индексных дескрипторов в группе;
- размер индексного дескриптора;
- идентификатор файловой системы.



Рис. 16. Обобщенная структурная схема файловой системы ext2

От целостности суперблока напрямую зависит работоспособность файловой системы. ОС Linux создаёт несколько резервных копий суперблока для возможности его восстановления в случае повреждения. Описание группы блоков, представляет собой массив, содержащий общую информацию обо всех блоках раздела. Битовая карта блоков — это структура, каждый бит которой показывает, отведён ли соответствующий ему блок какому-либо файлу. Если бит равен 1, то блок занят. Аналогичную функцию выполняет битовая карта индексных дескрипторов, показывая какие именно индексные дескрипторы заняты.

Место, обозначенное в таблице, как данные, отводится для хранения файлов.

### **Файловая система ext3**

*Ext3 или 3-я расширенная файловая система* - это журналируемая файловая система. Она является развитием ext2 и совместима с ext2. Ее можно рассматривать как файловую систему ext2 с журналированием. Возможность журналирования лишает необходимости дожидаться работы fsck и беспокоиться о повреждении метаданных. Можно переключаться с ext2 на ext3 и обратно на выбранном разделе диска - достаточно лишь задать правильный тип файловой системы для команды монтирования.

Основное отличие ext3 от ext2 состоит в том, что у нее предусмотрена запись некоторых данных, позволяющих восстановить файловую систему при сбоях в работе компьютера.

### **Файловая система ReiserFS**

*ReiserFS* — журналируемая файловая система, разработанная специально для Linux. В настоящее время ReiserFS поддерживается только под GNU/Linux.

Особенности:

1. Возможность упаковки нескольких небольших файлов в один блок во избежание фрагментации и потери дискового пространства.
2. Журналирование только метаданных.

При работе с файлами меньше 4 КБ с включённой функцией tail packing превосходит по производительности ext2 и ext3 в 10—15 раз.

### **Файловая система NFS**

*Сетевая файловая система NFS (Network File System)* делает возможным обращаться к файлам и каталогам, расположенным на удалённых компьютерах, как если бы эти файлы и каталоги были локальными. NFS - это клиент-серверное приложение. Должен быть установлен NFS-клиент, а на компьютерах, предоставляющих свое дисковое пространство для общего доступа - NFS-сервер.

#### **5.7.2. Типы файлов в файловой системе**

*Обычные файлы* представляют собой набор блоков (возможно, пустой) на устройстве внешней памяти, на котором поддерживается файловая система. Такие файлы могут содержать как текстовую информацию, так и произвольную двоичную информацию.

*Каталоги* – это тоже файлы, содержащие списки ссылок на файлы с данными или другие каталоги, которые пользователь видит как содержимое данного каталога.

*Файлы физических устройств*

Все подключаемые к компьютеру устройства (жесткие и съемные диски, терминал, принтер, модем и т. д.), представляются файлами. Если надо вывести на экран какую-то информацию, то система как бы производит запись в файл `/dev/tty01`.

*Физические устройства* бывают двух типов: символьными и блочными. Различие между ними состоит в том, как производится считывание и запись информации в эти устройства. Взаимодействие с символьными устройствами производится посимвольно, в режиме потока байтов. К таким устройствам относятся, например, терминалы. На блок-ориентированных устройствах информация записывается и считывается блоками. Примером устройств этого типа являются жесткие диски.

Взаимодействием с физическими устройствами в Linux управляют драйверы устройств. Для взаимодействия с остальными частями ОС каждый драйвер образует коммуникационный интерфейс, который выглядит как файл. Большинство таких файлов для различных устройств располагаются в каталоге `/dev`.

Таблица 1. Основные файлы

Имя	Значение
<code>/dev/console</code>	Системная консоль, т. е. монитор и клавиатура, физически подключенные к компьютеру
<code>/dev/hd</code>	Жесткие диски с IDE-интерфейсом.
<code>/dev/sd</code>	Жесткие диски SATA, SCSI, USB накопители
<code>/dev/fd</code>	Файлы дисководов для гибких дисков.
<code>/dev/tty</code>	Файлы поддержки пользовательских консолей. В Linux эти файлы устройств обеспечивают работу виртуальных консолей (переключаться между которыми можно с помощью Alt-F1— Alt-F6)
<code>/dev/pty</code>	Файлы поддержки псевдо-терминалов. Применяются для удаленных рабочих сессий.
<code>/dev/ttS</code>	Файлы, обеспечивающие работу с последовательными портами. Если мышь подключается через последовательный порт, то <code>/dev/mouse</code> является символической ссылкой на соответствующий <code>/dev/ttSN</code>
<code>/dev/cua</code>	Специальные устройства для работы с модемами.
<code>/dev/null</code>	Это устройство — черная дыра. Все, что записывается в <code>/dev/null</code> , навсегда потеряно. На это устройство можно перенаправить вывод ненужных сообщений. Если <code>/dev/null</code> используется как устройство

*Именованные каналы (pipes)*

Еще один тип специальных файлов — *именованные каналы, или буферы FIFO (First In — First Out)*. Файлы этого типа служат в основном для организации обмена данными между разными приложениями (pipe - труба).

*Канал* — это удобное и широко применяемое средство обмена информацией между процессами. Все, что один процесс помещает в канал, другой может оттуда прочитать. Если два процесса, обменивающиеся информацией, порождены одним и тем же родительским процессом.

*Доменные гнезда (sockets)*

*Гнезда* — это соединения между процессами, которые позволяют им взаимодействовать, не подвергаясь влиянию других процессов. После того как связь установлена, общение программ происходит без участия файла гнезда: данные передаются ядром ОС непосредственно от одной программы к другой.

Несмотря на то, что другие процессы могут видеть файлы гнезд как элементы каталога, процессы, не участвующие в данном конкретном соединении, не могут осуществлять над файлами гнезд операции чтения/записи. Среди стандартных средств, использующих гнезда — система X Window, система печати и система syslog.

**Жесткие и Символические ссылки**

Файл в Linux может иметь несколько имен или "ссылок".

*Жесткая ссылка* является просто еще одним именем для файла. Она прописывается в индексном дескрипторе. Если вы удаляете одну из ссылок, то файл еще сохраняется на диске. Жесткие ссылки прямо указывают на номер индексного дескриптора и поэтому они могут быть только внутри одной файловой системы.

Для создания жесткой ссылки используется команда **ln**. Пример:

```
$ln /home/file1 ~/file2
```

*Символические ссылки* это дополнительные имена файлов. В отличие от жестких ссылок символические ссылки могут указывать на файлы, расположенные в другой файловой системе. Если исходный файл удален, символическая ссылка не удаляется, но становится бесполезной.

Для создания символической ссылки используется команда **ln** с опцией **-s**:

```
$ln -s имя_файла_или_каталога имя_ссылки
```

Пример:

```
$ln -s /home/test/file1 ~/file2
```

### 5.7.3. Монтирование файловых систем различных типов. Создание и проверка файловых систем различных типов

Программа *fdisk*.

Работа с программой **fdisk** осуществляется через командный интерфейс.

```
#!/sbin/fdisk /dev/hda
Command (m for help):
Список доступных команд мы получаем по команде <m>:
Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)
Command (m for help):
```

Где:

Команда <p> выводит полный перечень имеющихся на диске разделов.

Команда <a> - снять/установить флаг активности раздела;

Команда <d> - удалить раздел;

Команда <l> - вывести список известных программе типов разделов;

Команда <n> - создать новый раздел;

Команда <o> - создать новую пустую таблицу разделов в стиле DOS;

Команда <q> - выйти из программы, не записывая на диск сделанные изменения;

Команда <t> - изменить тип раздела;

Команда <v> - проверить таблицу разделов;

Команда <w> - записать исправленную таблицу разделов на диск и выйти из программы;

Команда <x> - дополнительные возможности (только для экспертов).

Жесткий диск разбивается на разделы. Создание разделов облегчает выполнение резервного копирования, решает задачу разграничения полномочий, повышает производительность и ограничивает потенциальный ущерб, наносимый вышедшими из-под контроля программами. ОС Linux может работать с разными типами файловых систем.

Создание файловой системы семейства ext2 (ext2,ext3,ext4) подразумевает создание в данном разделе диска суперблока, таблицы индексных дескрипторов и совокупности блоков данных. Делается все это все с помощью команды **mkfs**:

```
#mkfs.ext3 /dev/hda5
```

Создать файловую систему на дискете:

```
#mkfs.ext2 /dev/fd0
```

В новой файловой системе автоматически создается один каталог с именем `lost+found`. Он используется программой `fsck`. Для того, чтобы начать работать с новой файловой системой необходимо подключить (монтировать) ее в дерево каталогов с помощью команды **mount**.

Команде `mount` надо указать устройство и "точку монтирования". Точка монтирования это каталог в уже существующем дереве каталогов, который будет корневым каталогом для подключаемой файловой системы.

```
#mount -t <type> <device> <path>
```

где *type* задает тип файловой системы, *device* указывает, на каком устройстве (в каком разделе) она находится, а *path* задает точку монтирования. Пример:

```
#mount -t ext3/dev/hdb1 /mnt/disk2
```

подсоединяет файловую систему первого раздела на втором жестком диске к каталогу **/mnt/disk2**.

Отметим, что после монтирования файловой системы в каталог `/mnt/disk2` прежнее содержимое этого каталога станет недоступно до тех пор, пока не размонтировать вновь подключенную файловую систему. Прежнее содержимое не уничтожается, а становится временно недоступным.

Конфигурационный файл **/etc/fstab** необходим для обеспечения автоматического монтирования файловых систем при загрузке ОС Linux. Каждая строка этого файла содержит описание подключения файловой системы и состоит из 6 полей разделенных пробелами:

- имя устройства. Может использоваться имя локального устройства (`/dev/hda5`), имя сетевой файловой системы NFS (`pc21:/home/jim`, что указывает на каталог `/home/jim` на машине с именем `pc21`) или метку (`label`) имени локального устройства;
- точка монтирования;
- тип файловой системы;
- опции монтирования;
- уровень дампа. Это поле используется программой `dump`, предназначенной для создания резервных копий. Если файловая система должна участвовать в процессе резервного копирования, то здесь должно стоять число 1, если нет — 0. Возможны и другие значения;

- порядок проверки файловых систем программой `fsck`. Системы с меньшими значениями этого поля проверяются раньше. Системы с одинаковыми номерами проверяются, если это возможно, параллельно.

Если пропустить тип файловой системы команда `mount` попытается самостоятельно определить тип монтируемой файловой системы.

```
#mount /dev/hdb1 /mnt/disk2
```

Когда монтируемая файловая система указана в файле `/etc/fstab`, то в команде `mount` достаточно указать только один аргумент — либо имя устройства, либо точку монтирования. Все остальные параметры команда `mount` возьмет из файла `/etc/fstab`.

Обычно монтировать файловые системы может только `root`, но если в поле опций монтирования файла `/etc/fstab` указать `user`, то соответствующую файловую систему смогут смонтировать все пользователи. Так, если в `/etc/fstab` имеется строка

```
/dev/cdrom /cd iso9660 ro,user,noauto,unhide,
```

то любой пользователь сможет смонтировать файловую систему на своем CDROM, используя команду

```
$mount /dev/cdrom или $mount /cd
```

Команды **mount** и **umount** поддерживают в актуальном состоянии таблицу смонтированных файловых систем – файл `/etc/mtab`.

Если вы хотите монтировать файловую систему только для чтения, то в `/etc/fstab` надо либо указать опцию `r` (read only, по умолчанию подразумевается `rw`), либо использовать команду `mount` с параметром `-r`.

Размонтирование файловой системы выполняется с помощью команды **umount**. В качестве аргумента команде `umount` надо дать либо имя устройства, либо точку монтирования

```
#umount /mnt/disk2
```

Демонтировать файловую систему может только тот пользователь, который ее смонтировал или `root`. Для того чтобы операцию демонтажа мог выполнить любой пользователь, в поле параметров соответствующей строки файла `/etc/fstab` надо указать опцию **users**.

Демонтирование файловой системы возможно только тогда, когда в ней нет открытых файлов и в системе нет процессов, использующих эту файловую систему.

*Метка (label) раздела* – альтернативное независимое имя устройства. Возможная проблема монтирования устройства заключается в том, что физическое подключение устройства может измениться или система даст устройству другое имя. Это особенно характерно для `scsi`-устройств. Использование меток при монтировании устройств поможет избежать подобные проблемы. Метка задается командой

```
$e2label <устройство> <метка>
```

Пример:

```
$e2label /dev/hdb1 data
$mount LABEL=data /mnt/disk2
```

*Команда fsck.* Основная функция программы **fsck** заключается в восстановлении логической непротиворечивости файловой системы. При выполнении этой команды производится поиск следующих ошибок:

- сектора, которые используются одновременно двумя файлами;
- сектора, которые включены в список свободных секторов, хотя они содержат часть какого-то файла;
- сектора, которые не содержат информации, но не включены в список свободных секторов;
- индексные дескрипторы файлов (inodes), не указанные ни в одном каталоге;
- неверная общая информация в суперблоке и т. д.

Формат запуска команды следующий:

```
# fsck [опции] [-t fstype] [--fs-options] filesystem
```

где *fstype* — тип проверяемой файловой системы, а в качестве *filesystem* можно указать либо имя устройства, либо точку монтирования.

При загрузке ОС Linux в некоторых случаях происходит автоматический запуск команды **fsck**. Основных причин для выполнения **fsck** на этапе загрузки системы две: некорректный выход из системы в предыдущий раз (например, резкое отключение питания или неисправность аппаратуры) и достижение заданного порога для количества выполнений операции размонтирования файловой системы. При каждом выполнении операций монтирования-размонтирования файловой системы в ее суперблоке делаются специальные отметки. Если последнее размонтирование завершилось корректно, то файловая система помечается как "чистая" (clean) и число операций монтирования-размонтирования увеличивается на единицу. Если корректного размонтирования не было, то файловая система помечена как "грязная" (dirty). На этапе загрузки ОС проверяется, являются ли все файловые системы "чистыми", и если нет, то для "грязных" систем выполняется команда **fsck**.

Иногда требуется запустить **fsck** и вручную. При этом лучше всего предварительно перевести систему в однопользовательский режим и размонтировать проверяемые файловые системы.

Если **fsck** обнаруживает "потерянные файлы", т. е. такие файлы, которые не указаны ни в одном из каталогов, она помещает их в каталог `lost+found` на верхнем уровне



проверяемой файловой системы. Поскольку имена файлов регистрируются только в родительском каталоге, то в данном случае их "истинные" имена неизвестны, и команда присваивает им имена, совпадающие с номерами их индексных дескрипторов.

#### 5.7.4. Настройка квот

*Ограничение дискового пространства* (квотирование) является очень полезной возможностью.

Команда **quota** позволяет просмотреть ограничения дискового пространства пользователя. Квоты устанавливает root. Существуют два типа квот - на количество файлов (inodes) и на использования дискового пространства (blocks). Квоты могут быть установлены как для пользователя, так и на группу. Если вы входите в группу, которая превысила наложенное на ее квоты, то вы не сможете использовать дисковое пространство, даже если вы не превысили квоту как пользователь.

*Для каждой квоты характерны четыре параметра:*

- квота, которая используется в данный момент
- "мягкая" квота (softlimit)
- "жесткая" квота (hardlimit)
- время, по истечении которого мягкая квота будет интерпретироваться как жесткое

Мягкая квота определяет число блоков, которые пользователь все еще может превысить, жесткую квоту превысить нельзя. При попытке сделать это, пользователь получит сообщение об ошибке.

При истечении определенного времени (7 дней) мягкая квота переходит в жесткую. За это время пользователь должен удалить ненужные ему файлы и после этого он может использовать это же время мягкую квоту.

*Настройка квот.* Необходимо определить на какую файловую систему необходимо наложить квоты.

Отредактируйте файл **/etc/fstab**

```
/dev/hda1      /          ext2    defaults
/dev/hda4      /home      ext2    defaults,usrquota
/dev/hda5      /usr       ext2    defaults,usrquota,grpquota
none          /proc      proc    defaults
```

Слово **usrquota** означает ограничение пространства для пользователей на данном устройстве, а **grpquota** - для групп.

*Проверка квот.* Периодически нужно проверять квоты для пользователя. Для этого используется команда **quotacheck**. Для проверки файловой системы на число блоков, которые используются пользователем, выполните команду

```
#quotacheck -avug
```

Проверку нужно выполнять при некорректной заверении работы ОС Linux. Вы можете включить эту команду в один из rc-сценариев и выполнять ее также, как и fsck.

Только root имеет право использовать команды, связанные с квотированием. Команда quota используется для проверки используемых ограничений любого пользователя, а команду repquota - для проверки используемого пространства и ограничений для всех пользователей на данной файловой системе. При запуске программы quotacheck -avug будут автоматически изменены файлы квот, а также будет изменена таблица ядра. Файлы квот имеют размер 2Мб, даже если ни один из пользователей не использует квот.

Для того, чтобы получить информацию о квотах наложенных на пользователя или группу выполните команду

```
#repquota -ua или #repquota -ga
```

User		used	Block limits			File limits			
			soft	hard	grace	used	soft	hard	grace
root	--	487082	0	0		12147	0	0	
daemon	--	399	0	0		2	0	0	
news	--	626	0	0		46	0	0	
www	--	156	0	0		1	0	0	
user	+-	1024	1	0	none	1	0	0	
den	+-	1024	1	0	none	94	0	0	
operator	+-	1024	1	0	none	94	0	0	
ifmail	+-	3030	1	0	none	794	0	0	

Перед определением квот необходимо сгенерировать необходимые файлы для поддержания баз данных квот. Это делается с помощью команды **quotacheck**.

Пример:

```
#quotacheck -u /u01
```

И включить подсчет квот с помощью команды **quotaon**.

Пример:

```
#quotaon -f /u01
```

*Определение квот.* Для этого существует команда **edquota**. Для ограничения дискового пространства пользователя нужно использовать команду **quota** с параметром **-u**, а для определения квот для группы - с параметром **-g**. После запуска этой команды будет запущен редактор, который указан в переменной окружения \$EDITOR, для редактирования квот. Изменять нужно только число, которое следует за словом **hard** или **soft**.

```
#equota -u admin
```

```
disk quotas for user admin (uid 500):
Filesystem          blocks      soft      hard    inodes      soft      hard
/dev/sdb1            5141         0        200         1           0         0
```

Жесткое (hard) ограничение является максимальным значением, которое может иметь пользователь или группа на данной файловой системе.

Строка **/dev/sdb1: blocks: 5141, limits (soft = 0, hard = 200)** определяет количество блоков, которое может быть выделено для пользователей или группы. Обычно размер блока в ОС Linux составляет 1024 байт. В данном случае ограничение равно 200Мб.

Строка **inodes: 1, limits (soft = 0, hard = 0)** сообщает какое число inode (файлов, устройств, поименованных каналов(pipes)) может быть выделено для данного пользователя или группы.

Пользователь имеет некоторый интервал времени, по истечению которого "мягкое" ограничение перейдет в "жесткое". Данный интервал можно изменить с помощью команды **edquota -t**

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem          Block grace period   Inode grace period
/dev/sdb1            7days                7days
```

*Использование команды quota. Quota* используется для проверки ограничений дискового пространства пользователей и групп.

```
$quota [-guqv]
```

```
#quota [-qv] -u username
```

```
#quota [-qv] -g groupname
```

Параметр **-v** используется для вывода информации о файловых системах, которые не имеют активных ограничений, и файловых системах, на которых квоты уже активны, но не занят еще ни один блок.

Параметр **-q** используется для получения сведений о файловых системах, на которых превышено значение "мягкого" ограничения.

Параметр **-g** предоставляет информацию о квотах группы.

Например, можно просмотреть ограничения для пользователя **admin**

```
Disk quotas for user admin (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/sdb1  5141*  0  200          1  0  0
```

Если ограничения для данного пользователя не заданы, появится сообщение:

```
Disk quotas for user root (uid 0): none
```

*Запрет квоты для пользователя или группы*

Для этого нужно использовать программу **edquota**. Установите значение **soft** и **hard**, равное 0. Данный пользователь или группа сможет использовать дисковое пространство без ограничений.

## 5.8. Сетевые ресурсы. Клиентская часть.

### 5.8.1. Подключение рабочей станции в существующую сеть

ОС Linux создавалась с учетом необходимости работы в сети и поэтому сетевые функции поддерживаются на уровне ядра. Для настройки конкретной рабочей станции понадобится только сетевой инструментарий. Это графическая утилита Центр Управления Системой и текстовые утилиты из пакета net-tools, net-scripts и утилита iproute2.

*Настройка сетевого интерфейса.*

Интерфейсом в ОС Linux является устройство, через которое система получает и передает IP-пакеты. Роль интерфейса локальной сети может выполнять одно или несколько устройств: Ethernet-карта, ISDN-адаптер или модем. Каждое устройство имеет свой IP-адрес. Для выхода в локальные сети используется, как правило, Ethernet карта.

Может быть три варианта настройки IP адреса: постоянного IP-адрес, настройка получения адреса от DHCP сервера и метод IPv4LL (из сети 169.254.0.0/16 будет подобран ещё не использованный адрес и назначен на интерфейс).

Приступим сетевой настройке на примере графической утилиты **Центр управления системой**. При заполнении полей все значения попадут в соответствующие конфигурационные файлы. Месторасположение и назначение этих файлов:

**/etc/sysconfig/network** - файл содержит общие настройки сети

**/etc/net** - система конфигурации сети

**/etc/resolv.conf** - файл конфигурации для процедур сервера имен

**/etc/hosts** — файл, содержащий локальную база данных доменных имен

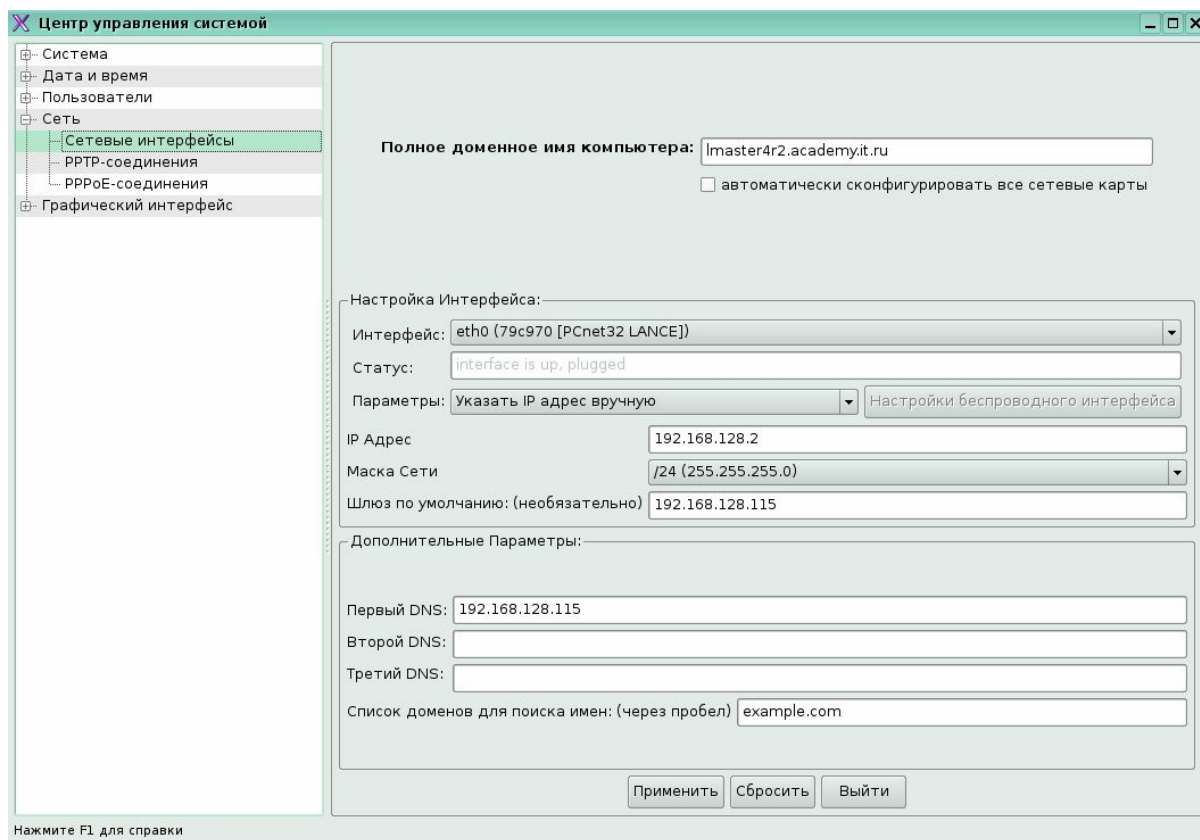


Рис. 17. Настройка сетевых ресурсов в программе Центр управления системой

Для настройки имени рабочей станции заполните поле **Полное доменное имя компьютера** (Конфигурационный файл `/etc/sysconfig/network`, параметр `hostname`).

Сетевой интерфейс должен быть определен и активирован ОС Linux. Поле **Интерфейс** на рисунке должно отображать название вашей настраиваемой сетевой карты или драйвера. Если сетевых карт несколько, выберите из выпадающего списка необходимый сетевой интерфейс.

Поле **Параметры** является выпадающим списком режимов настройки сетевого интерфейса. При выборе значения **Указать IP адрес вручную** необходимо ввести обязательный параметр **IP Адрес** (Конфигурационный файл `/etc/net/ifaces/eth0/ipv4address`, параметр `BOOTPROTO=static`). При необходимости работать с компьютерами из других подсетей нужно заполнить поле **Шлюз по умолчанию** (Конфигурационный файл `/etc/net/ifaces/eth0/ipv4route`).

При выборе значения **Получить настройки IP адреса с помощью DHCP** на экране появится возможность настройки еще нескольких параметров специфичных для DHCP (Конфигурационный файл `/etc/net/ifaces/eth0/ipv4address`, параметр `BOOTPROTO=dhcp` и т.д.).

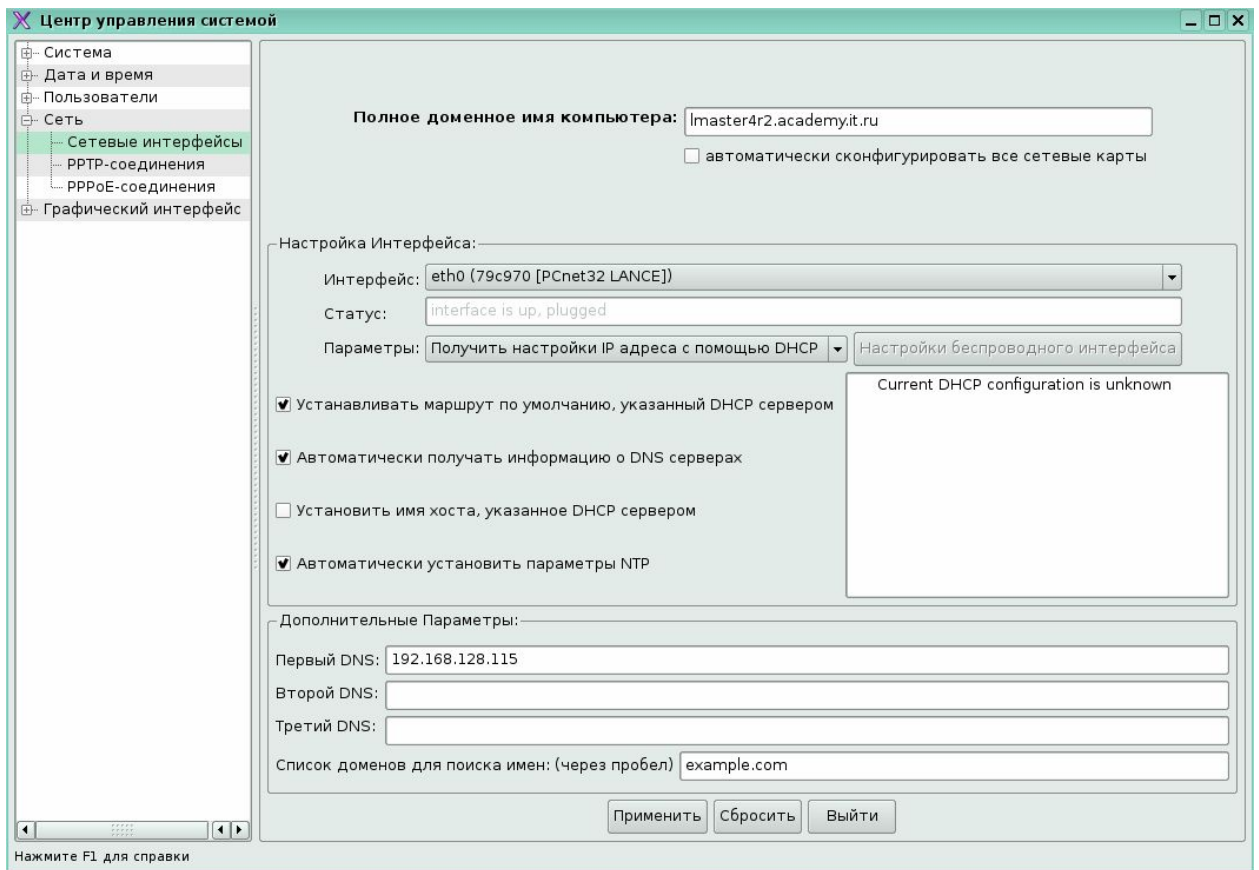


Рис. 18 ..Настройка DHCP-клиента в программе Центр управления системой

После заполнения полей в блоке **Настройка интерфейса** приступим к **Дополнительным параметрам**. Эти поля относящиеся к службе имён DNS (Файл/etc/resolv.conf).

После окончания всех изменений нужно нажать на клавишу ”Применить” и все значения сохранятся в своих конфигурационных файлах. Автоматически перезапустится сетевая служба.

### Утилиты командной строки

Для сложных конфигураций необходимо пользоваться утилитами командной строки или напрямую редактировать конфигурационные файлы.

#### Тестирование сетевого соединения

Чтобы проверить, соединен ли компьютер с сетью, попробуйте дать команду ping, указав ей в качестве параметра IP-адрес одного из компьютеров сети.

```
$ping 192.168.0.2
```

или через службу DNS

```
$ping pc1
```

Если соединение с сетью установлено, должны появиться и периодически обновляться строчки примерно такого вида:

```
64 bytes from 192.168.0.2: icmp_seq=0 ttl=32 time=1.2 ms
```

64 bytes from 192.168.0.2: icmp\_seq=4 ttl=32 time=1.1 ms

Это означает, что сетевое соединение работает. Для того чтобы прервать тестирование сети, нажмите комбинацию клавиш **<Ctrl><C>**.

Примеры использования команд для настройки сети:

Команда вывода настроек сетевого интерфейса

```
$ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:E0:F1:9B
          inet addr:192.168.128.2  Bcast:192.168.128.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4964 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2589 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:3693468 (3.5 MiB)  TX bytes:388519 (379.4 KiB)
          Interrupt:177 Base address:0x1400
```

Команда динамических изменений настроек (IP адрес) сетевого интерфейса.

```
#ifconfig eth0 inet 192.168.128.3 netmask 255.255.255.0 up
```

Команда добавления виртуальной сетевой карты:

```
#ifconfig eth0:1 192.168.1.100 up
```

Команда определения мас-адресов сетевых интерфейсов, с которыми работает наша рабочая станция.

```
#arp
```

Продвинутая команда конфигурирования сетевого интерфейса

```
#ip link show
```

```
#ip route list
```

Команда выполнения запросов к DNS-серверам.

```
#host www.google.com
```

Команда остановки сетевого интерфейса

```
#ifdown eth0
```

Команда запуска сетевого интерфейса

```
#ifup eth0
```

Команда просмотра содержимого различных структур данных, связанных с сетью

```
#netstat -nr
```

```
Kernel IP routing table
Destination      Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.128.0    0.0.0.0         255.255.255.0   U           0  0          0 eth0
```

## 5.8.2. Удалённый доступ к компьютеру. Работа с программами удалённого доступа (telnet, ssh, ftp)

*Программы telnet и rlogin*

Для того чтобы воспользоваться программой **telnet**, необходимо знать имя или IP-адрес удаленного компьютера, работающего под управлением ОС Linux. Предположим, что на компьютере linux2 имеется пользователь user5. В таком случае вы можно дать команду

```
$ telnet linux2
```

Появится сообщение **Connected to server linux2** и приглашение к входу в систему linux2. После ввода имени (user5) и пароля, можно работать компьютере linux2. Команда **rlogin** так же может быть использована для выхода на удаленный компьютер:

```
$ rlogin -l user5 linux2
```

Закреть сессию нужно командой **exit** и происходит возврат к командной строке локальной оболочки.

При работе с telnet или rlogin команды выполняются в оперативной памяти удаленной системы (можно просматривать (по команде ls) каталоги и файлы на дисках удаленного компьютера. Вывод результатов осуществляется на ваш монитор. Внутри telnet невозможно открыть для просмотра файл, расположенный на локальном диске. Локальный компьютер выполняет роль удаленного терминала. Для организации обмена файлами между вашим компьютером и удаленным, можно воспользоваться программой **ftp**.

Программа *ftp* — это пользовательский интерфейс к стандартному протоколу передачи файлов по Интернету — File Transfer Protocol. Программа позволяет передавать файлы на удаленный компьютер и получать файлы с удаленного компьютера. Однако, введя команду ftp, вы запускаете только клиентскую программу. Для того чтобы получить доступ к файлам удаленного компьютера, на нем должен быть запущен ftp-сервер. Кроме того, необходимо знать либо имя и пароль пользователя, либо ftp-сервер должен разрешать анонимный доступ. После запуска **ftp** появится приглашение:

```
ftp >
```

Если ввести знак вопроса, программа выдаст перечень возможных команд. Первая команда, которую нужно в этом случае ввести, — команда **open**, после которой надо указать сетевое имя компьютера, на котором запущен ftp-сервер. Если анонимный доступ к этому серверу разрешен, то вы получите запрос на ввод имени и пароля пользователя. По команде **pwd** можно узнать имя текущего каталога на удаленном компьютере, а по команде **dir** — вывести список файлов и подкаталогов этого каталога. Команда **cd имя\_каталога** используется для смены текущего каталога на удаленном компьютере.

Для пересылки файла на удаленный компьютер используется команда

```
ftp > put имя_файла
```

Копирования файла с удаленного компьютера в текущий каталог на локальном диске — команда

```
ftp > get имя_файла
```

Пересылка всех файлов:

```
ftp > mget *
```



Пересылку можно прервать с помощью комбинации клавиш **<Ctrl>+<C>**.

Команда **close** завершает сеанс работы с удаленным компьютером, не выходя из программы **ftp**. Выход из программы – команда **bye**.

*SSH (Secure Shell — безопасная оболочка)* — сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и передачу файлов. SSH шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. SSH-клиенты и SSH-серверы имеются для большинства ОС. Для того чтобы получить доступ к удаленной системе, на ней должна быть запущена служба **sshd**.

SSH позволяет безопасно передавать в незащищенной среде практически любой другой сетевой протокол, можно не только удаленно работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео, например, с веб-камеры. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удаленного запуска клиентов X Window System.

*Примеры использования SSH*

Команда подключения к локальному SSH-серверу из командной строки для пользователя **pacify**:

```
$ssh pacify@127.0.0.1
```

### 5.8.3. Настройка клиента NFS

*Network File System (NFS)* — протокол сетевого доступа к файловым системам, NFS абстрагирована от типов файловых систем, как сервера, так и клиента.

*Как настроить NFS клиента*

На клиентской системе должна быть запущена служба **portmap**.

```
#service portmap start
```

Используйте команду **mount** для подключения файловой системы NFS с другого компьютера:

```
#mount shadowman:/mnt/export /mnt/local
```

*Каталог* - точка подключения на локальном компьютере (**/mnt/local** в приведённом выше примере) должен существовать.

В этой команде, *shadowman* - это имя файлового сервера NFS, */mnt/export* - файловая система, экспортируемая сервером shadowman, а */mnt/local* - каталог на локальном компьютере, в который будет подключена файловая система.

## 5.9. Настройка серверной части Linux.

### 5.9.1. Сетевые службы Linux – установка, настройка и администрирование стандартных сетевых служб DNS, Apache, SMB, DHCP.

#### Настройка DNS (Bind)

*Система доменных имён* — распределённая база данных. С помощью пакета Bind можно организовать мощный полнофункциональный DNS-сервер. Любой DNS сервер решает две основные задачи: именованье узлов сети и выполнение этой операции быстро. Как во всякой сложной программе, в Bind обнаруживаются уязвимости. В Линукс Мастере используется Bind версии 9, который запускается с использованием механизма “песочницы” chroot. **Chroot** — это операция изменения корневого каталога в ОС Linux. Любая программа, запущенная с изменённым корневым каталогом, будет иметь доступ только к файлам, содержащимся в данном каталоге.

В каталоге `/etc` от пакета Bind осталась только символическая ссылка на главный файл настроек `/etc/named.conf`. Служба Named – это сервер доменных имен пакета BIND. Named может реализовывать следующие функции: мастер (primary) сервер, подчиненный (slave) или вторичный сервер и кэширующий (cache) сервер. Программа named в момент запуска или перезапуска считывает данные из своего файла конфигурации и файлов описания зон, если они существуют, и таким образом настраивается.

Мастер сервер имен читает файл с данными о зоне и отвечает за эту зону.

Вторичный сервер имен разделяет нагрузку с основным сервером, и обрабатывает запросы, если основной сервер не работает.

Кэширующий сервер инициирует поиск имен в пределах вашей зоны, спрашивая один из первичных или подчиненных серверов. Второй основной задачей такого сервера является хранение найденной информации.

Named проводит свою работу с файлами в каталоге `/var/lib/bind`.

Использование пакета Bind для именованья компьютеров в сети требует создания двух зон (т. н. прямой и обратной), содержащих в виде записей определённого формата информацию о доменных именах компьютеров и об их роли в этих доменах. Каждая зона должна включать запись типа SOA (State Of Authority, сведения об ответственности). В этой записи определяются основные временные и административные параметры домена, в том числе электронный адрес лица, ответственного за домен (администратора) и серийный номер зоны. *Серийный номер* — число в диапазоне от 0 до 4294967295 ( $2^{32}$ ); каждое изменение, вносимое в зону, должно сопровождаться увеличением этого номера.

Обнаружив увеличение серийного номера, кэширующие и вторичные серверы признают все кэшированные записи из этой зоны устаревшими. Структура номера может быть любой, лишь бы он постоянно увеличивался. Удобно использовать формат годмесяцчисловерсия, где все числа, кроме года, двузначные, а версия может обнуляться раз в день, соответствовать времени (например, по формуле  $100*(\text{часы}*60+\text{минуты})/(60*24)$ ) или иметь сквозную нумерацию (в этом случае появляется сложность с переходом от версии 99 к версии 100, то есть 0). Даже если серийный номер генерируется автоматически, рекомендуется пользоваться этим форматом, наглядно отражающим время создания зоны. Пример зоны, не содержащей ничего, кроме записи SOA и обязательной записи типа NS, находится в файле `/var/lib/bind/zone/empty`.

Кроме записи типа **SOA**, в каждой зоне должна быть хотя бы одна запись типа **NS** (Name Server), указывающая адрес DNS-сервера, авторитетного в этом домене (как минимум — адрес сервера, на котором запущен named). Несколько зон включаются в настройку Bind автоматически (файл `/var/lib/bind/etc/rfc1912.conf`). Они нужны для обслуживания сети, привязанной к сетевой заглушке (127.0.0.1/8). Стоит обратить внимание на то, что имя домена, который обслуживается зоной, задаётся в файле настроек, а в самом файле зоны можно использовать относительную адресацию (без точки в конце имени), так что операция переименования домена делается редактированием одной строки. В ПСПО рекомендуется добавлять описания зон в конфигурационный файл `/var/lib/bind/etc/local.conf`.

Прямая зона нужна для преобразования доменного имени в IP-адрес — операции, необходимой многим программам постоянно. Большинство записей в прямой зоне — типа **A** (Address) — предназначены именно для этого. Другие часто встречающиеся типы записей — это **CNAME** (Canonical Name, настоящее имя), позволяющий привязать несколько дополнительных имён к одному, и **MX** (Mail eXchange, обмен почтой), указывающий, куда пересылать почтовые сообщения, в поле адресат которых встречается определённое доменное имя. Вот пример прямой зоны для воображаемого домена **internal.domain.net**:

```
# cat /var/lib/bind/etc/local.conf
```

```

zone "internal.domain.net" {
    type master;
    file "internal.domain.net";
};

# cat /var/lib/bind/zone/internal.domain.net
$TTL      1D
@         IN      SOA      server root.server (
                                2004082202      ; serial
                                12H               ; refresh
                                1H               ; retry
                                1W               ; expire
                                1H               ; ncache
                                )
          IN      NS       server
          IN      MX      10  server
server   IN      A        10.10.10.1
www      IN      CNAME    server
mail     IN      CNAME    server
jack     IN      A        10.10.10.100
jill     IN      A        10.10.10.101

```

Используются правила умолчания: если в записи некоторое поле опущено, оно наследуется от предыдущей. Так, вместо А можно было бы всюду написать IN А, а вместо MX — @ IN MX (@ означает имя домена, указанное в конфигурационном файле). Как видно из примера, всю работу в сети делает компьютер с адресом 10.10.10.1, он же server.internal.domain.net, он же www.internal.domain.net и mail.internal.domain.net. Несмотря на наличие среди CNAME этого сервера имени mail, MX-запись указывает всё же не на него, а на действительный адрес — так рекомендовано RFC.

Для того чтобы преобразовывать IP-адреса в доменные имена, у каждой сети должна быть обратная зона. Если такой зоны нет, и в файле /etc/hosts тоже ничего не написано, операция не выполнится. Такое преобразование нужно гораздо реже и в основном по соображениям административным: для того, чтобы выяснить принадлежность компьютера по его IP-адресу. В случае внутренней сети обратная зона необязательна.

Обратная зона состоит почти целиком из записей типа PTR (Pointer, указатель). Чтобы не умножать сущностей, решено было не вводить новый способ работы сервера имён и представить обратное преобразование IP-адреса как прямое преобразование доменного имени специального вида. Например, чтобы выяснить доменное имя компьютера с адресом 1.2.3.4, необходимо запросить информацию о доменном имени 4.3.2.1.in-addr.arpa. Таким образом, каждой подсети класса С (или выше) соответствует определённый домен, в котором можно найти ответ. Вот как выглядит обратная зона для нашего воображаемого домена:

```
# cat /var/lib/bind/etc/local.conf
```

```

zone "12.11.10.in-addr.arpa" {
    type master;
    file "12.11.10.in-addr.arpa";
};

# cat /var/lib/bind/zone/12.11.10.in-addr.arpa
$TTL      1D
@         IN      SOA      server.internal.domain.net. root.server.internal.domain.net (
                                2004082201      ; serial
                                12H              ; refresh
                                1H              ; retry
                                1W              ; expire
                                1H              ; ncache
                                )
0         IN      NS       server.internal.domain.net.
1         IN      PTR      internal.domain.net.
100      IN      PTR      server.internal.domain.net.
101      IN      PTR      jack.internal.domain.net.
101      IN      PTR      jill.internal.domain.net.

```

Относительные адреса, использованные в левой части записей PTR, раскрываются в полные вида адрес.12.11.10.in-addr.arpa, а в правой части используются полные (которые вполне могут указывать на имена в разных доменах).

Проверить синтаксическую правильность конфигурационного файла и файла зоны можно с помощью утилит `named-checkconf` и `named-checkzone`, входящих в пакет `bind`. Они же используются при запуске службы командой **service bind start**.

Проверку работоспособности, доступности DNS-сервера нужно делать утилитой **dig**:

```

# dig jack.internal.domain.net

; <<>> DiG 9.2.4rc5 <<>> jack.internal.domain.net
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32751
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; QUESTION SECTION:
;jack.internal.domain.net.      IN      A
;; ANSWER SECTION:
jack.internal.domain.net. 86400 IN      A      10.11.12.100
;; AUTHORITY SECTION:
internal.domain.net.      86400  IN      NS     server.internal.domain.net.
;; ADDITIONAL SECTION:
server.internal.domain.net. 86400 IN      A      10.11.12.1
;; Query time: 37 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Aug 22 16:07:17 2004
;; MSG SIZE  rcvd: 95

```

## Настройка DHCP

DHCP в ОС Linux реализован в виде сервиса сервера (`dhcpcd`) и сервиса клиента (`dhcpcd`). Сервис сервера непосредственно отвечает за назначение IP-адресов клиентам, при входе и выходе их из сети. Клиентский сервис запускается на стороне клиента.

Конфигурационным файлом для **dhcpcd** является `/etc/dhcp.conf`. При запуске DHCP-сервера происходит выделение IP-адресов согласно содержащимся в файле `/etc/dhcp.conf` установкам. Выделенные адреса **dhcpcd** регистрирует в файле **dhcpcd.leases**, который обычно находится в каталоге `/var/dhcpd`.

После изменения конфигурационного файла, сервис `dhcpcd` необходимо перезапустить. При этом используйте команду

```
#service dhcpcd stop
```

 для остановки службы, и команду

```
#service dhcpcd start
```

 для её запуска.

Каждая сетевая карточка имеет уникальный собственный MAC-адрес. Допустим, вам нужно связать какой-то MAC-адрес с определенным IP-адресом. Для этого воспользуйтесь конструкцией **host**:

```
host myhost {
hardware ethernet xx:xx:xx:xx:xx:xx;
fixed-address 192.168.0.9;
}
```

Ее нужно вставить в ту конструкцию подсети **subnet**, которой принадлежит назначаемый IP-адрес. Это означает, что компьютеру с аппаратным адресом `xx:xx:xx:xx:xx:xx` будет назначен IP-адрес `192.168.1.9`.

Полный пример:

```
# Подсеть 192.168.0.0, маска сети 255.255.255.0
subnet 192.168.0.0 netmask 255.255.255.0 {
# маршрутизатор по умолчанию
option routers 192.168.0.1;
# маска подсети 255.255.255.0
option subnet-mask 255.255.255.0;
option domain-name "domain.ru";
# задание широковещательного адреса
option broadcast-address 192.168.0.255;
# включение IP-Forwarding
option ip-forwarding on;
# адрес DNS сервера, который будут использовать клиенты
option domain-name-servers 192.168.0.1;
# диапазон адресов для клиентов
# 192.168.0.10-192.168.0.254
range 192.168.0.10 192.168.0.254;
# сказать клиентам, чтобы отдали адрес через 21600 секунд (6 часов)
# после получения адреса
default-lease-time 21600;
# забрать адрес самому через 28800 секунд (8 часов)
max-lease-time 28800;
# описание трех клиентов (dhcp50, dhcp51, dhcp52) и их аппаратных адресов
    host dhcp50 {
        hardware ethernet 00:40:C7:34:90:1E;
# обратите внимание на то, что вы должны использовать IP-адрес из указанного
# ранее диапазона адресов 192.168.0.10-254.
        fixed-address 192.168.0.50;
    }
    host dhcp51 {
        hardware ethernet 00:40:C7:34:90:1F;
        fixed-address 192.168.0.51;
    }
    host dhcp52 {
        hardware ethernet 00:40:C7:34:90:2A;
        fixed-address 192.168.0.52;
    }
}
```

## Настройка SMB

Samba представляет собой комплект серверного и клиентского программного обеспечения для осуществления связи UNIX-машин с сетями Microsoft.

Samba также использует протокол NetBIOS over TCP/IP.

*Для работы в сетях SMB необходимы:*

- клиент;
- сервер;
- средства администрирования.

Всё это есть в пакетах `samba-client`, `samba-client-cups`, `samba-common`, `samba`, `samba-swat`, входящих в состав дистрибутива.

*При использовании SMB доступны следующие ресурсы:*

- сетевые диски;
- прямые пути к дискам;
- принтеры;
- доменная авторизация и управление.

*Краткий обзор каталогов и файлов*

Все файлы конфигурации и авторизации Samba расположены в каталоге `/etc/samba` и его подкаталогах.

**smb.conf** - основной конфигурационный файл Samba. Он нужен не только серверной части, но и всем остальным компонентам этой системы;

**smbpasswd** - аналог `/etc/passwd` и `/etc/tcb/*/shadow` — файл пользователей сервера Samba с паролями.

**smbusers** - файл соответствий имён сетевых и локальных пользователей SMB; это удобный метод для организации административных и гостевых входов на сервер. Соответствие пользователей Samba и системных производится на основе символьных имён;

**/var/log/samba/\*** - файлы журнала серверной части Samba.

**/var/spool/samba** - каталог динамического спулинга печати сервера Samba. На несильно загруженных серверах печати он обычно пуст; наличие там множества файлов в то время, когда ни один из клиентов не печатает — явный признак сбоя сервера печати;

**/var/cache/samba/\*** - файлы (как правило, двоичные базы данных), формируемые в процессе работы различных компонентов SambaSamba;. Наиболее примечательны:

**browse.dat** и **wins.dat** - текстовые файлы, их названия говорят сами за себя;

**winbind\*.tdb** - базы данных доменных пользователей, формируемых winbind

**/var/lib/samba/\*** - служебные каталоги для администратора сервера.

*Серверные компоненты:*

**/usr/sbin/nmbd** - сервер преобразования имён и адресов;

**/usr/sbin/smbd** - файловый сервер;

**/usr/sbin/winbindd** - сервер импорта пользователей и групп с PDC;

**/usr/sbin/swat** - средство настройки Samba с web-интерфейсом;

**/etc/init.d/smb** и **/etc/init.d/winbind** - управляющие скрипты инициализации сервисов.

*Клиентские компоненты:*

**/usr/bin/smbclient** - интерактивное приложение для просмотра сетевых ресурсов;  
**/sbin/mount.smb**, **/sbin/mount.smbfs**, **/usr/bin/smbumount**, **/usr/sbin/smbmnt**,  
**/usr/bin/smbmount** - средства монтирования/размонтирования сетевых файловых систем.

*Утилиты*

**/usr/bin/smbpasswd** - управление пользователями и подключением к домену;  
**/usr/bin/wbinfo** - отображение списка пользователей, импортированных winbindd;  
**/usr/bin/testparm** - проверка синтаксиса конфигурационных файлов;  
**/usr/bin/smbstatus** - отображение статуса процессов smbd и nmbd;  
**/usr/bin/nmblookup** - программа разрешения имён WINS

*Настройка сервера*

В большинстве случаев настройка Samba заключается в редактировании основного конфигурационного файла **/etc/samba/smb.conf** и управлении пользователями с помощью **smbpasswd**.

*Обычный сервер*

Это компьютер, предоставляющий в сеть файловые ресурсы. Фактически, это файловый сервер, имеющий собственную базу авторизации пользователей.

Для создания сервера, необходимо подправить стандартный конфигурационный файл **/etc/smb.conf**, подставив требуемые имя рабочей группы и имена ресурсов и создать учётные записи пользователей.

Вот основные записи в **/etc/smb.conf**, которые создадут нам обычный сервер.

```
[global]
# Секция [global] определяет общие настройки серверной части Samba
# в целом для всех ресурсов.
# Имя рабочей группы OFFICE
workgroup = OFFICE
# Уровень определения прав доступа на уровне пользователей
security = user
# Приоритет данного сервера среди других компьютеров рабочей группы:
# определяем, кто именно будет главной машиной, отвечающей за
# отображение ресурсов сети. Для сравнения, у Win9X os level = 34, а
# у NT4 os level = 64.
os level = 65
# Очевидно, что раз нет домена - нет и мастера.
domain master = no
# Не стоит становиться сервером паролей для окрестных машин. Так что
# если к вам прибежал разъярённый администратор соседнего NT-сервера с
# жалобами на то, что его не пускают на его собственный сервер -
# поставьте
# domain logons = no
domain logons = no
# Как правило, в простейшей сети WINS не нужен, поэтому мы его отключаем также и у себя.
wins support = no
|
```

Теперь надо определить, какие каталоги предоставить в сеть. Для каждого ресурса существует отдельная секция. Самый простейший вариант — каталог с именем **public**:



```

# имя ресурса, видимое в сети
[public]
# комментарий, видимый в сети как комментарий к ресурсу
comment = Public Stuff
# путь к каталогу ресурса
path = /home/samba/public
# опметка о доступе на чтение всем авторизованным пользователям (в том
# числе и гостевым, если они определены)
public = yes
# запрещение на запись всем пользователям
writable = no
# разрешение на запись всем пользователям, входящим в системную
# группу staff
write list = @staff

```

Также создаются сетевые ресурсы сервера с различными правами доступа.

Samba исполняется не в “песочнице” **chroot**, внутри ресурсов можно использовать любые символические ссылки на расположенные локально и в сети (NFS, SMB) файловые объекты, что удобно при администрировании системы.

```

# Особые ресурсы - например, домашние каталоги пользователей:
# имя ресурса, которое автоматически будет заменено именем
# домашнего каталога пользователя, под которым подключился клиент
# и именно название его домашнего каталога будет отображено в сети
# как имя ресурса.
# Для получения доступа к этому ресурсу клиент должен предоставить
# серверу соответствующие имя и пароль, все прочие пользователи к
# этому ресурсу доступа не имеют вовсе.
[homes]
# комментарий, видимый в сети как комментарий к ресурсу
comment = Home Directories
# признак невидимости - данный ресурс виден в сети только тому
# пользователю, который является его владельцем. К этому
# ресурсу можно обратиться непосредственно задав его имя, но при
# просмотре ресурсов сети он будет виден только владельцу.
browseable = no
# Разрешение на запись.
writable = yes
Принтеры:
# имя ресурса, которое будет видно в сети. Кроме него, в сети будут
# также видны и локальные принтеры под теми же именами, что и в
# системе по команде lpr.
[printers]
# комментарий, который игнорируется.
comment = All Printers
# Путь к каталогу, в котором располагается спул принтеров,
# предоставляемых в сеть через Samba
path = /var/spool/samba
# невидимость ресурса в браузинге, он подменяется системным
# ресурсом.
browseable = no
# разрешение на печать для гостевого захода.
guest ok = yes
# запрещение на запись, поскольку в спул пишет сама Samba, а не
# пользователь.
writable = no

```

```

# признак того, что это именно принтер, а не файловый ресурс
printable = yes
# маска для создания файлов заданий на печать
create mode = 0700
# Команды, выполняемые Samba для того, чтобы напечатать документ.
# Использование драйвера клиента, применяется для не-UNIX клиентов.
print command = lpr-cups -P %p -o raw %s -r
# Использование драйвера CUPS на стороне сервера (на стороне
# клиентов используется generic PostScript драйвер).
# Следующие команды являются стандартными при установке printing=cups,
# их можно изменить в случае необходимости.
lpr command = lpr -P %p
lprm command = cancel %p-%j

```

### Учётные записи пользователей

Учётные записи хранятся в файле `/etc/samba/smbpasswd`. Учётные записи пользователей делятся на две категории:

- записи о компьютерах, входящих в домен;
- записи о пользователях, зарегистрированных на данном сервере.

Для создания и использования учётной записи в `/etc/samba/smbpasswd`, предварительно необходимо создать соответствующую запись в `/etc/passwd`. Общее правило — для каждого пользователя в `/etc/samba/smbpasswd` обязательно должен существовать пользователь в `/etc/passwd`.

Для управления учётными записями предназначена утилита **smbpasswd**.

Команда создания нового пользователя:

```
#smbpasswd -a <User_name>
```

Смена пароля у существующего пользователя:

```
#smbpasswd <User_name>
```

Удаление существующего пользователя:

```
# smbpasswd -x <User_name>
```

Приостановление учётной записи без удаления:

```
# smbpasswd -d <User_name>
```

Подключение данного компьютера к существующему домену:

```
# smbpasswd -j <Domain_name> -U <Administrator_name>
```

## NFS сервер

Для запуска службы NFS необходимо запустить сервис **portmap**. Portmap является сервером, который преобразует программные номера **RPC** (Remote Procedure Call — интерфейс распределённых вычислений) в номера портов протокола **DARPA** (протокол взаимодействия через сеть). **Portmap** обслуживает все обращения к RPC.

Сервер RPC при запуске сообщает процессу portmap номер прослушиваемого порта, а также номера тех программ, которые он готовится обслужить. Если клиенту нужно обратиться к RPC с конкретным номером программы, он сначала должен войти в

контакт с процессом portmap на серверной машине и определить номер порта связи с пакетами RPC.

Запущена ли служба можно проверить командой:

```
#!/usr/sbin/rpcinfo -p
rpcinfo: can't contact portmapper: RPC: Remote system error - Connection refused
#service portmap start
#!/usr/sbin/rpcinfo -p
  program vers proto  port
  100000    2    tcp   111  portmapper
  100000    2    udp   111  portmapper
```

Также перед запуском NFS необходимо провести конфигурацию. Файл конфигурации `/etc/exports` состоит из строк, включающих в себя ресурс и список клиентов, которым разрешен доступ к нему.

- имя компьютера;
- wildcards в названии домена (например : `linux-*.mondomaine.fr`) ;
- netgroup, в виде `@group`, если используется NIS и определены какие - либо;
- IP - адрес...

Также необходимо запустить службы **rpc.mountd** и **rpc.nfs** для доступа к работающему NFS - серверу.

При попытке клиента получить доступ к файловой системе сначала запрос направляется к **mountd**. Далее проверяется возможность запроса через etab. Затем проверка идет на уровне ядра - файлы **hosts.allow** и **hosts.deny**. Ядро использует для проверки **exportfs**, допуская обновление `/var/lib/nfs/etab`. При возникновении ситуации невозможности доступа - запрос отклоняется, и ядро обновляет xtab для нового клиента.

#### *Клиент NFS*

С помощью команды **mount** можно получить доступ к разным файловым системам. Команда сообщает ядру тип файловой системы, устройство и точку монтирования. Опция **-t** для обозначения типа файловой системы - например для NFS выглядит следующим образом :

**-t nfs.**

*Пример:* предположим, что на компьютере scool работает NFS - сервер и предоставляет в пользование свой `/usr/local`, для доступа к которому с компьютера с именем jill необходимо выполнить следующую команду:

```
#mount -t nfs scool:/usr/local /usr/local
```

## 5.9.2. FTP

### *FTP-сервер из пакета vsftpd*

В состав Линукс Мастер входит пакет **vsftpd (Very Secure FTP Daemon)** — FTP-сервер, позволяющий просматривать содержимое каталогов сервера, загружать файлы с

сервера или на сервер для анонимных пользователей и пользователей, зарегистрированных на сервере и имеющих доступ к его ресурсам. Он включает в себя множество полезных функций, которых нет в других FTP-серверах, такие как:

- высокие требования к безопасности;
- контроль над полосой пропускания канала;
- хорошая расширяемость;
- возможность создавать виртуальных пользователей;
- поддержка IPv6;
- высокая производительность;
- возможность устанавливать виртуальные IP-адреса;
- высокая скорость работы.

Настройки осуществляются через конфигурационный файл `/etc/vsftpd.conf`.

В целях безопасности сервер по умолчанию сконфигурирован именно для предоставления анонимного доступа. Запрещены любые команды записи, а также доступ локально зарегистрированных пользователей.

Область для анонимного доступа начинается с каталога `/var/ftp/pub`. Этот каталог традиционно используется для размещения общедоступных файлов.

Чтобы предоставить доступ к FTP-серверу для локально зарегистрированных пользователей, необходимо внести изменения в конфигурационный файл `/etc/vsftpd.conf`. Необходимо убрать комментарий перед директивой `local_enable=YES`, **перезапустить сервис vsftpd** и клиенты этого FTP-сервера получают доступ к любым каталогам файловой системы, для которых такой доступ разрешён исходя из прав локальных пользователей. Это могут быть как домашние каталоги пользователей, так и системные каталоги. Если в настройках vsftpd разрешена запись, клиенты получают и все права на запись, которыми располагают эти пользователи.

Сервер vsftpd может ограничить возможность пользователей, зарегистрированных локально, перемещаться по дереву каталогов.

Чтобы запретить анонимный доступ к FTP-серверу, прокомментируйте строки `anonymous_enable=YES` и перезапустите службу `vsftpd`.

## **5.10. Дополнительные возможности администрирования Linux.**

### **5.10.1. Резервное копирование и восстановление**

Регулярное резервное копирование необходимо по причине предотвращения потери данных. Принятие простых мер поможет избежать необратимых последствий.

Хотя своевременное резервное копирование и не спасет вас от всех возможных проблем с испорченными данными, оно сделает возможным восстановление информации.

### *Создание резервных копий*

Количество информации, которую вы сможете восстановить, во многом зависит от того, как часто вы ее сохраняете и насколько надежно храните.

Для начала определитесь с тем, где вы будете хранить резервные копии. Это может быть отдельный жесткий диск или CD/DVD диски.

В простейшем случае скрипт для сохранения backup-копии будет выглядеть следующим образом:

```
#!/bin/sh
tar -zcf /backup/home.tar.gz /home
tar -zcf /backup/etc.tar.gz /etc
```

Этот скрипт делает следующее:

- Сохранение и сжатие директории **/home** (домашние директории пользователей) в отдельный файл;
- Сохранение и сжатие директории **/etc** (системные настройки) в отдельный файл.

Рассмотрим использованные опции команды **tar**:

- z - сжать файл с использованием gzip
- c - создать новый архив
- f - использовать указанный файл
- v - выводить список обработанных файлов
- p - сохранять информацию о правах доступа
- directory - директория для сохранения в архив
- exclude - исключить директорию при сохранении в архив

### *Замечание*

Если вы решили использовать отдельный HDD для резервных копий, то вам подойдет следующее решение.

Для начала подключите предварительно отформатированный HDD. Далее подразумевается, что это **/dev/hdb1**, т. е. первый раздел на втором диске канала IDE1.

Создайте точку монтирования:

```
cd /
mkdir backup
```

Дополните ваши скрипты для резервного копирования следующими строками:

```
#!/bin/sh
mount /dev/hdb1 /backup
```

```
#####  
# ваш скрипт #  
#####  
umount /backup
```

Со временем вы сами разработаете наиболее удобный и эффективный для вас способ проведения регулярного резервного копирования. Например, вы можете исключить директории (опция `--exclude`) с музыкой и фильмами из списка резервируемых директории.

#### *Резервное копирование по расписанию*

Далее вам предстоит настроить запуск скрипта по расписанию. На самом деле нет ничего проще. Со времен ОС UNIX существует программа `cron`, предназначенная для выполнения действий по расписанию. Откройте файл `/etc/crontab` и запишите новое правило:

```
#мин      час      число      месяц      день недели      команда  
0         1        *          *          5 /usr/bin/full-backup
```

Это правило будет выполняться каждую пятницу в 1 час ночи. Для указания удобного для вас времени запишите нужные значения в соответствующих колонках. В примере предполагается, что командой резервного копирования является `/usr/bin/full-backup`. Замените эту команду на имя вашего скрипта.

Результат работы будет отправлен пользователю `root` по почте (при условии, что у вас настроен SMTP-сервер).

#### *Восстановление из резервных копий*

Ниже приведены основные команды для восстановления файлов из архива резервной копии. Перед замещением существующего файла убедитесь, что замена действительно необходима!

Перед извлечением файлов из резервной копии бывает необходимым просмотреть содержимое архива. Для этого укажите опцию `-t` команды `tar`.

Например, следующая команда позволит просмотреть содержимое архива `/backup/backup-07-March-2005.tar.gz`:

```
tar -ztvpf /backup/backup-07-March-2005.tar.gz
```

Для извлечения файлов из архива предназначена опция `-x` команды `tar`.

Например, следующая команда восстановит все файлы из архива `/backup/backup-12-March-2005.tar.gz`:

```
tar -zxvpf /backup/backup-12-March-2005.tar.gz
```

Для восстановления определенных файлов из архива укажите их имена после имени архива.

Например, следующая команда восстановит файлы `home/alenitchev/adt/backup.xml` и `etc/sendmail.cf` из архива `/backup/backup-17-March-2005.tar.gz`:

```
tar -zxvpf /backup/backup-17-March-2005.tar.gz home/alenitchev/adt/backup.xml etc/sendmail.cf
```

Перед восстановлением файла из резервной копии убедитесь, что восстанавливаемый файл не заменит более новый экземпляр.

### 5.10.2. Анализ ошибок и устранение неисправностей

ОС Linux это стабильная система, но только в том случае, если установить и эксплуатировать её правильно. Многие проблемы появляются после неправильной установки, ошибочных изменений в конфигурационных файлах или неправильной работе оборудования.

Время от времени могут возникнуть и проблемы даже в правильно настроенной системе. Все они благополучно могут быть решены, если вы заранее сделаете загрузочный CD или flash диск. Так же подойдет и загрузочный диск, с которого ставилась ваша ОС Linux. Установочный диск имеет режим восстановления (rescue mode).

*Общий подход в восстановлении работы системы в следующем:*

1. Рассматривать проблемы как симптомы.
2. Собирать данные уточняющие проблемы.
3. Определить, что еще работает.
4. Формировать гипотезы о том, что случилось.
5. Проверять журнальные файлы для подтверждения фактов.
6. Копировать конфигурационные файлы, прежде чем их изменять.

*Основные проблемы можно классифицировать следующим образом:*

#### 1. Проблемы с графической подсистемой.

- Не решайте проблемы отладки графической подсистемы на 5 runlevel;
- убедитесь, что диск не переполнен или квоты закончились;
- убедитесь, что запущен сервер фонтов `xfs`;
- узнайте входит ли ваше оборудование с список оборудования, которое поддерживаем ваша ОС.

#### 2. Проблемы с сетью

- *Проблемы с DNS.*

Для решения этих проблем применяйте утилиты `dig`, `host`, `nslookup`

- *Проблемы с конфигурацией IP.*

Для решения этой проблемы применяйте утилиту `ifconfig`

- *Проблемы со шлюзом по умолчанию (gateway).*

Для решения этой проблемы применяйте утилиту `route`, `netstat`

- *Есть ли проблемы с модулем ядра для вашего сетевого интерфейса.*

Для решения этой проблемы применяйте утилиту `insmod`, `rmmmod`, `modprobe`.

- *Активирован ли ваш сетевой интерфейс*

Для решения этой проблемы применяйте утилиту `ifup`, `ifdown`, `ifconfig`, Центр управления системой.

### 3. Проблемы с загрузкой

#### 3.1. Если загрузка не доходит до экрана:

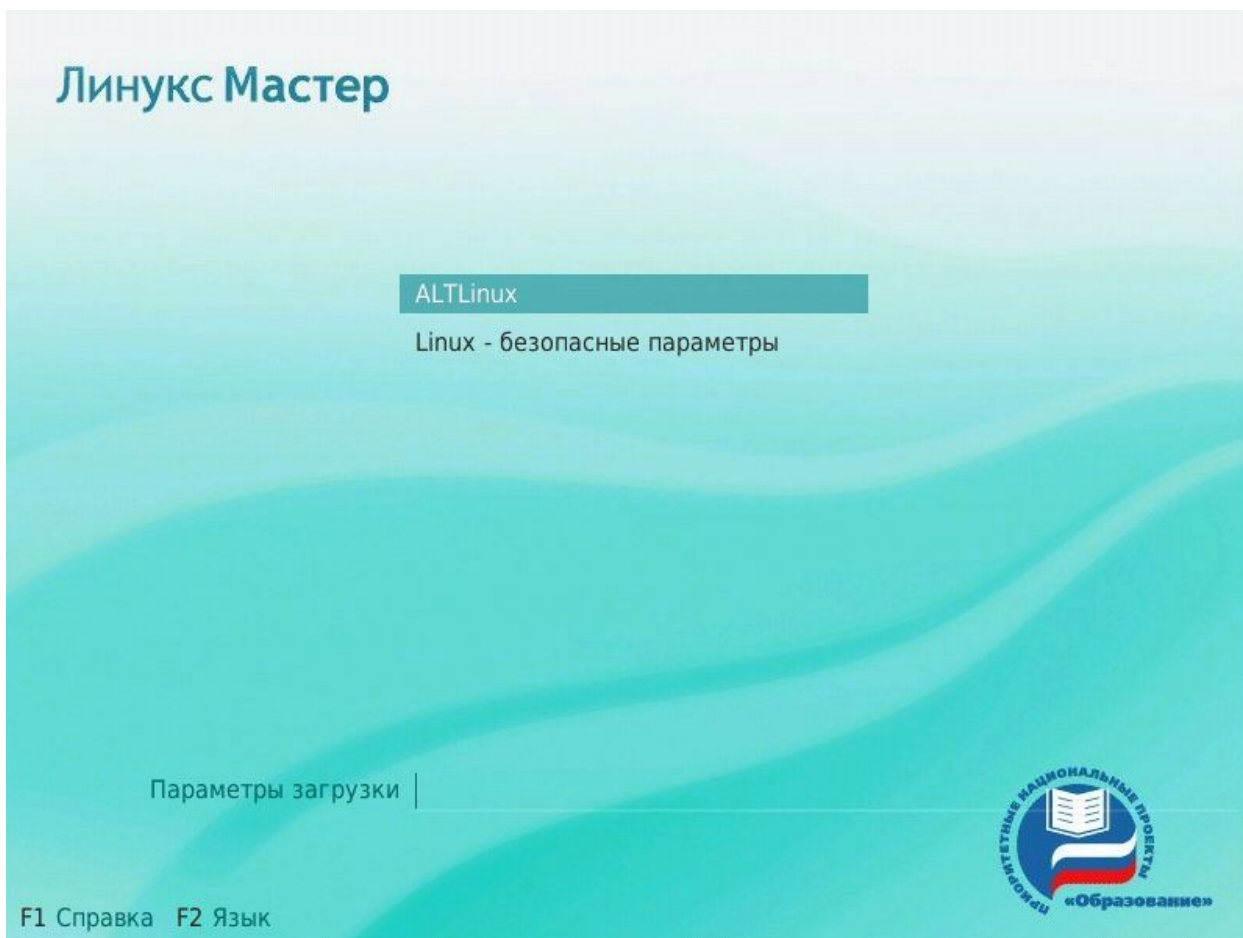


Рис. 18. Splash-экран ОС Линукс Мастер

Это может указывать на одну из следующих причин:

- *Нарушение конфигурация загрузчика `lilo`*

Загружаемся с установочного диска в режиме `rescue`. При необходимости исправить конфигурационный файл `/etc/lilo.conf` и запустить утилиту `lilo` для перестройки карты размещения.

- *Разрушение загрузочного сектора MBR*

- Загружаемся с установочного диска в режиме `rescue`. Уточняем, на каком устройстве находится загрузочный сектор с помощью утилиты **`fdisk`**. Допустим это `/dev/sda`.
- Монтируем корневой раздел с жёсткого диска к файловой системе `RescueCD`:



## **mount /dev/sda5 /mnt**

- Если есть отдельный раздел с /boot, монтируем его в точку **/mnt/boot**:

## **mount /dev/sda1 /mnt/boot**

- Убеждаемся, что смонтировали действительно тот раздел

## **ls /mnt/**

- Монтируем виртуальные файловые системы для использования их при работе с жёстким диском:

```
mount --bind /proc /mnt/proc
```

```
mount --bind /dev /mnt/dev
```

```
mount --bind /sys /mnt/sys
```

- Выполняем команду chroot в систему на жёстком диске:

## **chroot /mnt**

- Восстанавливаем загрузчик, предварительно подкорректировав, если это необходимо, файл **/etc/lilo.conf**

lilo

*- Неправильные установки BIOS*

Зайти в BIOS и исправить установки. Перегрузиться.

### *3.2 Ядро не загружается или частично загрузилось с паникой*

*- Нарушена целостность файлов образа ядра и initrd*

- Загружаемся с установочного диска в режиме rescue.
- Уточняем, на каком устройстве находится загрузочный сектор с помощью утилиты fdisk.  
Допустим это /dev/sda.
- Монтируем корневой раздел с жёсткого диска к файловой системе RescueCD:

## **mount /dev/sda5 /mnt**

- Если есть отдельный раздел с /boot, монтируем его в точку /mnt/boot

## **mount /dev/sda1 /mnt/boot**

- Убеждаемся, что смонтировали действительно тот раздел

## **ls /mnt/**

- Восстанавливаем файлы ядра и initrd из архива или дистрибутива.
- Запускаем lilo для формирования новой карты размещения.  
*- Неправильные установки параметров ядра.*
- Загружаемся с локального устройства в монопольном режиме.
- Вводим пароль пользователя root.

- Перемонтировать корневой раздел из **readonly** в **read/write**

Пример:

```
mount -t ext2 -n -o remount,rw /dev/hd-root/
```

- В текстовом редакторе исправляем параметры ядра в файле **/etc/sysctl.conf**
- Перегружаемся командой **exit**

*3.3 Ядро загрузилось, но паника появилась при попытке монтировать корневую файловую систему или при запуске /sbin/init*

*- Проблемы с загрузчиком **lilo***

- Загружаемся в монопольный режим.
- Перемонтируем корневую файловую систему в read/write режим
- Запускаем программу lilo для перестройки карты размещения.

*- Проблемы с целостностью /sbin/init или /etc/inittab*

- Загружаемся в монопольный режим.
- Перемонтируем корневую файловую систему в read/write режим
- Или восстанавливаем из дистрибутива **/sbin/init**
- Или в текстовом редакторе исправляем **/etc/inittab**
- Корневая файловая система разрушена и немонтируется
- Загружаемся с установочного диска в режиме rescue.
- Уточняем, на каком устройстве находится загрузочный сектор с помощью утилиты fdisk. Допустим это /dev/sda.
- Восстанавливаем корневую файловую систему. Эта операция очень индивидуальна для каждого сбоя файловой системы.

*3.4. Ядро загрузилось правильно, скрипт /etc/rc.d/sysinit запустился, но прервал запуск по ошибке*

*- Проблемы с файлом /sbin/bash*

Действия:

- Загружаемся с установочного диска в режиме rescue.
- Уточняем, на каком устройстве находится загрузочный сектор с помощью утилиты fdisk. Допустим это /dev/sda.
- Монтируем корневой раздел с жёсткого диска к файловой системе RescueCD:

```
mount /dev/sda5 /mnt
```

- Убеждаемся, что смонтировали действительно тот раздел

```
ls /mnt/
```

- Выполняем команду **chroot** в систему на жёстком диске:

```
chroot /mnt
```

- Восстанавливаем файл **/sbin/bash** из пакета bash-3.rpm с использованием утилиты **rpm**
  - Проблемы с конфигурационным файлом **/etc/fstab**
- Загружаемся в монопольный режим.
- Перемонтируем корневую файловую систему в read/write режим
- В текстовом редакторе исправляем **/etc/fstab**
  - *Ошибки при настройке квот или в программном RAID'e*
    - Загружаемся в монопольный режим.
    - Перемонтируем корневую файловую систему в read/write режим
    - В текстовом редакторе исправляем **/etc/fstab** (параметры, связанные с квотами или монтированием RAID)
    - Перегружаемся.
    - При необходимости восстанавливаем RAID.
  - *Проблемы с некорневыми файловыми системами*
    - Загружаемся в монопольный режим.
    - Перемонтируем корневую файловую систему в read/write режим
    - В текстовом редакторе исправляем **/etc/fstab** (убираем автоматическое монтирование сбойных файловых систем)
    - Перегружаемся
    - Восстанавливаем сбойные файловые системы

### 3.5. При запуске сервисов появляются ошибки

- Сбойные сервисы неправильно сконфигурированы для этого **runlevel'a**

Настраиваем работу сервисов

- Ошибки характерные для работы сервисов

Настраиваем работу сервисов

- Ошибки связанные неправильной конфигурацией графической подсистемы на 5 runlevel

Настраиваем работу сервисов

## 4 Проблемы, связанные с работой файловых систем

Сбой в работе файловых систем это один из самых распространенных сбоев связанных с загрузкой ОС. Он может быть вызван крахом во время работы или некорректным завершением работы ОС, т.е. завершением работы системы без размонтирования файловой системы. Корректное завершение работы файловой системы контролируется специальным маркером. Во время загрузки маркер проверяется, и если предыдущая работа файловой системы завершилась некорректно, маркет окажется «грязным». При «грязном» маркере дальнейшее поведение зависит от типа файловой

системы. Если используется ext2, то утилита **fsck** последовательно проверит все блоки файловой системы. Это может занять продолжительное время. При использовании журналируемых файловых систем (ext3, ext4, reiserfs и т.д.) произойдет восстановление последней транзакции файловой системы и загрузка ОС будет завершится успешно.

## 6. Работа в сети

### 6.1. Локальные сети

*Вычислительные сети* – это совокупность связанных между собой нескольких компьютеров, разнесенных в пространстве. Связь между компьютерами осуществляется с помощью специальных периферийных устройств – сетевых адаптеров, соединенных относительно протяженными каналами связи. Взаимодействие между компьютерами сети происходит за счет передачи сообщений через сетевые адаптеры и каналы связи. С помощью этих сообщений один компьютер обычно запрашивает доступ к ресурсам другого компьютера. Такими ресурсами могут быть как данные, хранящиеся на диске, так и разнообразные периферийные устройства – принтеры, модемы и др. Разделение локальных ресурсов каждого компьютера между пользователями сети – основная цель создания вычислительной сети.

Основные характеристики вычислительной сети - операционные возможности, время доставки сообщений, производительность и стоимость обработки данных.

*Операционные возможности сети* - перечень основных действий по обработке данных. Главные ЭВМ (серверы), входящие в состав сети, обеспечивают пользователей всеми традиционными видами обслуживания: средствами автоматизации программирования, доступом к пакетам прикладных программ, доступом к базам данных и т.д. Дополнительно вычислительная сеть предоставляет следующие виды услуг:

- удаленный ввод заданий - выполнение заданий, поступающих с любых терминалов, на любой рабочей станции (сервере) в пакетном или диалоговом режиме;
- передачу файлов между абонентами сети;
- доступ к удаленным файлам - обработку файлов, хранимых в удаленных ЭВМ;
- защиту данных и ресурсов от несанкционированного доступа;
- передачу текстовых, речевых и видео сообщений между абонентами;
- выдачу справок об информационных и программных ресурсах;
- распределенную обработку - параллельное выполнение задачи несколькими ЭВМ.

*Время доставки сообщений* определяется как статистическое среднее времени от момента передачи сообщения в сеть до момента получения сообщения адресатом.

Цена обработки данных формируется с учетом стоимости средств, используемых для ввода/вывода, передачи, хранения и обработки данных. На основе цен рассчитывается стоимость обработки данных, которая зависит от объема используемых ресурсов вычислительной сети (количества передаваемых данных, процессорное время), а также режима передачи и обработки данных.

В общем случае под локальной вычислительной сетью (ЛВС) понимают такую сеть, в которой все оборудование узлов сети (рабочие станции и периферийные устройства), объединяемое физическими линиями, размещается в пределах небольшой территории, ограниченной одним или несколькими помещениями, на расстоянии не более чем 1 - 2 км друг от друга. Такая сеть является более специализированной по классу решаемых задач (банковская, государственного учреждения), однако часто требует обмена и протокольной совместимости с другими локальными или глобальными (корпоративными) сетями.

В ЛВС наиболее эффективное средство связи между системами, объединяемыми в сеть, - последовательный интерфейс. В последовательных интерфейсах в качестве передающей среды используются коаксиальные кабели, витые пары, волоконно-оптические кабели, которые обеспечивают высокую пропускную способность до 100 Мбит/сек и более. При использовании таких каналов связи строятся различные конфигурации вычислительных сетей (топологии ЛВС). Наиболее распространенные конфигурации - звездная, кольцевая, шинная и древовидная.

### 6.1.1. Топология типа «звезда»

Концепция топологии сети в виде звезды пришла из области больших ЭВМ, в которой головная машина получает и обрабатывает все данные с периферийных устройств как активный узел обработки данных. Вся информация между двумя рабочими станциями проходит через центральный узел сети.



Рис.1. Топология типа «звезда»

Сети с такой топологией строятся, как правило, на базе метода коммутации каналов. В этом случае перед началом передачи информации абонент-инициатор передачи (вызывающий узел) запрашивает у центрального узла установление физического или логического соединения с абонентом-партнером (вызываемом узлом). После установления соединения соответствующий физический или логический путь монополюбно используется абонентами-партнерами для обмена информацией. По окончании обмена один из абонентов запрашивает у центрального узла разъединения.

Кабельное соединение довольно простое, так как каждая рабочая станция связана с узлом. Затраты на прокладку кабелей высокие, особенно когда центральный узел географически расположен не в центре топологии

При расширении вычислительной сети подобной топологии к новому рабочему месту необходимо прокладывать отдельный кабель из центра сети.

Производительность вычислительной сети в первую очередь зависит от производительности центрального файлового сервера, а пропускная способность сети определяется вычислительной мощностью узла и гарантируется для каждой рабочей станции. Коллизий (столкновений) в процессе передачи данных не возникает.

Топология в виде звезды является наиболее быстродействующей из всех топологий вычислительных сетей, поскольку передача данных между рабочими станциями проходит через центральный узел (при его хорошей производительности) по отдельным линиям, используемым только этими станциями.

Однако при данной топологии он может быть узким местом вычислительной сети. В случае выхода из строя центрального узла нарушается работа всей сети.

В настоящее время файловым серверам уделяется особое внимание с точки зрения надежности его работы, а так как для данной топологии характерна простота управления обменом информацией и механизмом против несанкционированного доступа, топология типа «звезда» находит широкое распространение при проектировании ЛВС.

### **6.1.2 Кольцевая топология**

При кольцевой топологии сеть не имеет явно выраженного центрального узла (сервера), а рабочие станции связаны одна с другой по кругу. Сообщения в подобной топологии циркулируют по кругу в строго заданном направлении. Рабочая станция посылает сообщение заданному адресату, предварительно получив из кольца запрос (маркер). Пересылка сообщений в подобной топологии является эффективной, так как большинство сообщений можно отправлять «в дорогу» по каналу связи одно за другим.

Каждая рабочая станция может выполнять функции сервера, а банки данных могут быть распределены между станциями.

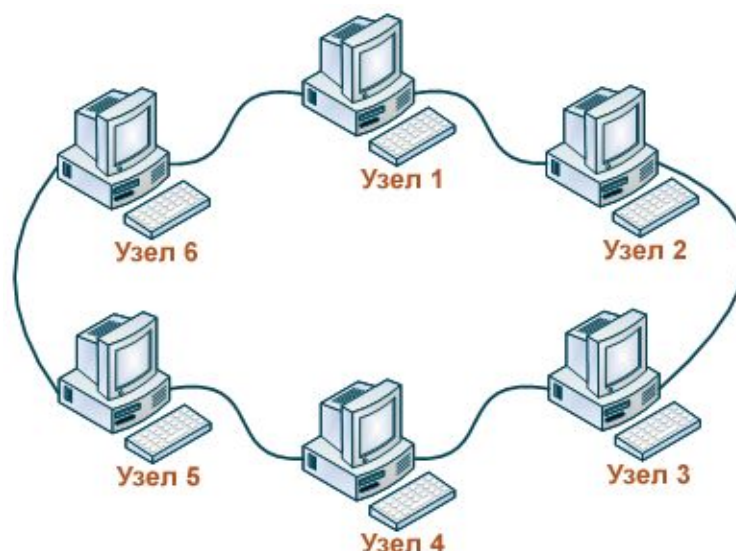


Рис.2. Кольцевая топология

Основная проблема при кольцевой топологии заключается в том, что каждая рабочая станция должна активно участвовать в пересылке информации, и в случае выхода из строя хотя бы одной из них вся сеть парализуется. Неисправности в кабельных соединениях локализуются легко. Подключение новой рабочей станции требует выключения сети, так как во время установки кольцо должно быть разомкнуто.

Специальной формой кольцевой топологии является логическая кольцевая сеть. Физически она монтируется как соединение звездных топологий. Отдельные звезды включаются с помощью специальных коммутаторов (англ. Hub - концентратор).

В зависимости от числа рабочих станций и длины кабеля между рабочими станциями применяют активные или пассивные концентраторы («хабы»). Активные концентраторы дополнительно содержат усилитель для подключения от 4 до 16 рабочих станций. Пассивный концентратор является исключительно разветвительным устройством (максимум на три рабочие станции). Каждой рабочей станции присваивается соответствующий ей адрес, по которому передается управление (от старшего к младшему и от самого младшего к самому старшему). Разрыв соединения происходит только для нижерасположенного (ближайшего) узла вычислительной сети, так что лишь в редких случаях может нарушаться работа всей сети.

### 6.1.3 Шинная топология

В сетях с шинной топологией все рабочие станции подключаются к одному каналу связи (коммуникационному пути) с помощью приемопередатчиков.

Рабочие станции могут непосредственно вступать в контакт с любой рабочей станцией, имеющейся в сети, и могут быть подключены или отключены без нарушения функционирования всей вычислительной сети. Канал оканчивается с двух сторон

пассивными терминаторами, которые поглощают передаваемые сигналы, поскольку по своей природе передача в такой сети является широковещательной (длинные линии – из курса теоретические основы электротехники).

Каждая рабочая станция подключается к шине непосредственно к соединителям кабельных секций (тройниковые соединители) либо с помощью специальной врезки, которая просто прокалывает коаксиальный кабель до контакта с центральным проводником.



Рис. 3. Шинная топология

Поскольку один общий канал связи (шина) разделяется между всеми абонентами сети, такие сети называются также моноканальными.

Пропускная способность и задержка в шинных сетях определяется большим числом параметров: методом доступа, полосой пропускания канала связи, числом узлов связи, длиной сообщений и др.

В данной топологии банки данных, также как и в сетях с кольцевой топологией, могут распределяться между рабочими станциями (станции ресурсов).

#### **6.1.4 Древоидная топология**

В локальной сети такого типа используется комбинация ранее рассмотренных типов топологий: “кольцо-звезда” либо “шина-звезда”. Сеть типа “кольцо-звезда” представлена на рисунке.

Сервера подключены к шине сети через контроллер, а к каждому серверу подсоединены рабочие станции по схеме «звезда». Сети такого типа применяются там, где невозможно непосредственное применение базовых сетевых структур в чистом виде.

Сети такой топологии обладают преимуществами каждого в отдельности типа сети («кольцо» и «звезда»), а также удобны при административном управлении сетью.

Недостаток такого типа в том, что выход из строя участка канала связи сети парализует работу группы абонентов этого участка.

Сравнительные характеристики различных топологий локальных сетей представлены в табл. 1.



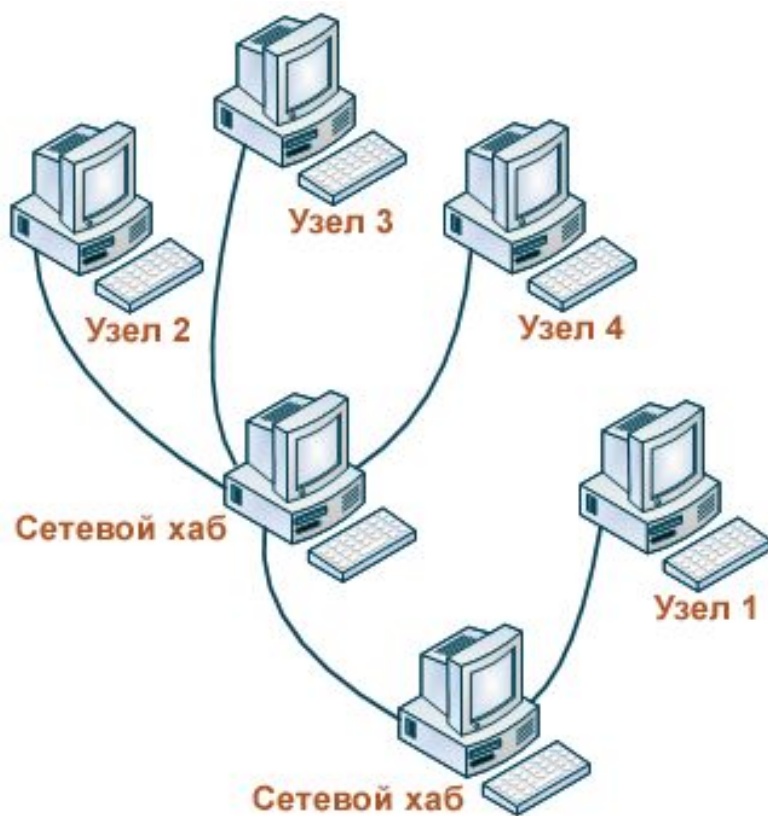


Рис.4. Древоподобная топология

Таблица 1. Сравнительные характеристики различных топологий локальных сетей

Характеристики	Топология			
	Звезда	Кольцо	Шина	Дерево
Сложность интерфейса	малая	малая	малая/средняя	средняя
Наращиваемость	малая	средняя	высокая	высокая
Надежность	средне/высокая	высокая	высокая	высокая
Стоимость подключения	высокая	средняя	низкая	низкая
Работа в реальном времени	очень хорошая	хорошая	плохая	средняя
Разводка кабеля	хорошая	удовлетворительная	хорошая	хорошая

## 6.2. Глобальная, мировая сеть

Если соединённые компьютеры находятся в разных частях города, а иногда и в разных городах и странах, то такие сети называют распределёнными. Иногда распределённые сети называют также территориальными. Часто к распределённой сети подключаются не отдельные компьютеры, а локальные сети. Таким образом, можно создавать корпоративные сети для предприятий, имеющих филиалы в других городах. Распределённые сети мирового масштаба также называют глобальными сетями. Интернет и является самой известной глобальной компьютерной сетью.

Отличительной особенностью Интернета является высокая надёжность. При выходе из строя части компьютеров и линий связи сеть будет продолжать функционировать. Такая надёжность обеспечивается тем, что в Интернете нет единого центра управления. Если выходят из строя некоторые линии связи или компьютеры, то сообщения могут быть переданы по другим линиям связи, т. к. имеются несколько путей передачи информации.

Локальные вычислительные сети позволили поднять на качественно новую ступень управление производственными объектами, повысить эффективность использования ЭВМ, поднять качество обрабатываемой информации, реализовать безбумажную технологию, создать новые технологии. Объединение ЛВС и глобальных сетей позволило получить доступ к мировым информационным ресурсам.

ПК, объединённые в сеть, делятся на абонентские – клиенты и вспомогательные – серверы. Клиенты выполняют все необходимые информационно-вычислительные работы и определяют ресурсы сети. Серверы – служат для преобразования и передачи информации от одной ЭВМ к другой по каналам связи и коммутационным устройствам ( host - ЭВМ ). К качеству и мощности серверов предъявляются повышенные требования.

*Клиент* – это приложение, посылающее запрос к серверу. Он отвечает за обработку и вывод информации, а также передачу запросов серверу.

*Сервер* – это персональная или виртуальная ЭВМ, выполняющая функции по обслуживанию клиента. Он распределяет ресурсы системы: принтеры, базы данных, программы и т.д. Существуют сетевые, файловые, терминальные серверы баз данных.

*Сетевой сервер* поддерживает выполнение следующих функций сетевой операционной системы: управление вычислительной сетью, планирование задач, распределение ресурсов, доступ к сетевой файловой системе, защиту информации.

*Терминальный сервер* поддерживает выполнение функций многопользовательской системы.

*Файл-сервер* обеспечивает доступ к центральной базе данных удалённым пользователем.

*Сервер баз данных* – многопользовательская система, обеспечивающая обработку запросов к базам данных. Он является средством решения сетевых задач, в которых локальные сети используются для совместной обработки данных.

*Коммутационная сеть* образуется множеством серверов и host -ЭВМ, соединённых физическими каналами связи, которые называют магистральными.

По способу передачи информации вычислительные сети делятся на:

- сети коммутации каналов,

- сети коммутации сообщений,
- сети коммутации пакетов,
- интегральные сети.

*При коммутации каналов* – сообщение между клиентами осуществляется по прямому каналу неизменному в течении всего сеанса. При лёгкости реализации такого способа реализации передачи информации его недостатки заключаются в низком коэффициенте использования каналов, высокой стоимости передачи данных, увеличения времени ожидания других клиентов.

*При коммутации сообщений* информация передаётся порциями, называемыми сообщениями. Прямое соединение обычно не устанавливается, а передача сообщения начинается после освобождения первого канала, второго и т. д., пока сообщение не дойдёт до адресата. Каждым сервером осуществляется приём информации, её сборка, проверка, маршрутизация и передача сообщения. Недостатком коммутации сообщений является низкая скорость передачи данных и невозможность проведения диалога между клиентами, хотя стоимость передачи уменьшается.

*При коммутации пакетов* обмен производится короткими пакетами фиксированной структуры. *Пакет* – часть сообщения, удовлетворяющая некоторому стандарту. Малая длина пакетов предотвращает блокировку линий связи, не даёт расти очереди в узлах коммутации. Она обеспечивает быстрое соединение, низкий уровень ошибок, надёжность и эффективность использования сети. Но при передаче пакета возникает проблема маршрутизации, которая решается программно-аппаратными методами. Наиболее распространённым способом являются:

- фиксированная маршрутизация,
- маршрутизация способом кратчайшей очереди.

*Фиксированная маршрутизация* предполагает наличие таблицы маршрутов, в которой закрепляется маршрут от одного клиента к другому, что обеспечивает простоту реализации, но одновременно - не равномерную загрузку сети.

*В методе кратчайшей очереди* используются несколько таблиц, в которых таблицы расставлены по приоритетам. Приоритет – функция обратная расстоянию до адресата. Передача начинается по первому свободному каналу с высшим приоритетом. При использовании этого метода задержка передачи пакета минимальна.

В настоящее время разработаны программно-аппаратные средства маршрутизации. *Повторитель* – самый простой тип устройства для соединения однотипных локальных ВС, он ретранслирует все принимаемые пакеты из одной ЛВС в другую.

Устройство связи, позволяющее соединять ЛВС с одинаковыми и разными системами сигналов, называется *маршрутизатор*. Он позволяет выполнять передачу пакетов в соответствии с определёнными протоколами, обеспечивает соединение ЛВС на сетевом уровне.

*Шлюз* – устройство соединения ЛВС с глобальной сетью.

Сети, обеспечивающие коммутацию каналов, сообщений и пакетов, называются *интегральными*. Они объединяют несколько коммутационных сетей. Часть интегральных каналов используется монопольно, т. е. для прямого соединения. Прямые каналы создаются на время проведения сеанса связи между различными коммутационными сетями. По окончании сеанса связи прямой канал распадается на независимые магистральные каналы.

### **6.3. Адрес компьютера. Имя компьютера**

Еще одной новой проблемой, которую нужно учитывать при объединении трех и более компьютеров, является проблема их адресации. К адресу узла сети и схеме его назначения можно предъявить несколько требований.

Адрес должен уникально идентифицировать компьютер в сети любого масштаба.

Схема назначения адресов должна сводить к минимуму ручной труд администратора и вероятность дублирования адресов.

Адрес должен иметь иерархическую структуру, удобную для построения больших сетей. Эту проблему хорошо иллюстрируют международные почтовые адреса, которые позволяют почтовой службе, организующей доставку писем между странами, пользоваться только названием страны адресата и не учитывать название его города, а тем более улицы. В больших сетях, состоящих из многих тысяч узлов, отсутствие иерархии адреса может привести к большим издержкам - конечным узлам и коммуникационному оборудованию придется оперировать с таблицами адресов, состоящими из тысяч записей.

Адрес должен быть удобен для пользователей сети, а это значит, что он должен иметь символьное представление например, Servers или dvoika.net.

Адрес должен иметь по возможности компактное представление, чтобы не перегружать память коммуникационной аппаратуры - сетевых адаптеров, маршрутизаторов и т. п.

Нетрудно заметить, что эти требования противоречивы - например, адрес, имеющий иерархическую структуру, скорее всего будет менее компактным, чем неиерархический (такой адрес часто называют «плоским», то есть не имеющим структуры). Символьный же адрес скорее всего потребует больше памяти, чем адрес-число.

Так как все перечисленные требования трудно совместить в рамках какой-либо одной схемы адресации, то на практике обычно используется сразу несколько схем, так что компьютер одновременно имеет несколько адресов-имен. Каждый адрес используется в той ситуации, когда соответствующий вид адресации наиболее удобен. А чтобы не возникало путаницы и компьютер всегда однозначно определялся своим адресом, используются специальные вспомогательные протоколы, которые по адресу одного типа могут определить адреса других типов.

Наибольшее распространение получили три схемы адресации узлов.

*Аппаратные (hardware) адреса.* Эти адреса предназначены для сети небольшого или среднего размера, поэтому они не имеют иерархической структуры. Типичным представителем адреса такого типа является адрес сетевого адаптера локальной сети. Такой адрес обычно используется только аппаратурой, поэтому его стараются сделать по возможности компактным и записывают в виде двоичного или шестнадцатеричного значения, например 0081005e24a8. При задании аппаратных адресов обычно не требуется выполнение ручной работы, так как они либо встраиваются в аппаратуру компанией-изготовителем, либо генерируются автоматически при каждом новом запуске оборудования, причем уникальность адреса в пределах сети обеспечивает оборудование. Помимо отсутствия иерархии, использование аппаратных адресов связано еще с одним недостатком - при замене аппаратуры, например, сетевого адаптера, изменяется и адрес компьютера. Более того, при установке нескольких сетевых адаптеров у компьютера появляется несколько адресов, что не очень удобно для пользователей сети.

*Символьные адреса или имена.* Эти адреса предназначены для запоминания людьми и поэтому обычно несут смысловую нагрузку. Символьные адреса легко использовать как в небольших, так и крупных сетях. Для работы в больших сетях символьное имя может иметь сложную иерархическую структуру, например ftp-arch1.ucl.ac.uk. Этот адрес говорит о том, что данный компьютер поддерживает ftp-архив в сети одного из колледжей Лондонского университета (University College London - ucl) и эта сеть относится к академической ветви (ac) Internet Великобритании (United Kingdom - uk). При работе в пределах сети Лондонского университета такое длинное символьное имя явно избыточно и вместо него удобно пользоваться кратким символьным именем, на роль которого хорошо подходит самая младшая составляющая полного имени, то есть имя ftp-arch1.

*Числовые составные адреса.* Символьные имена удобны для людей, но из-за переменного формата и потенциально большой длины их передача по сети не очень экономична. Поэтому, во многих случаях, для работы в больших сетях в качестве адресов узлов используют числовые составные адреса фиксированного и компактного форматов.

Типичным представителями адресов этого типа являются IP- и IPX-адреса. В них поддерживается двухуровневая иерархия, адрес делится на старшую часть - номер сети и младшую - номер узла. Такое деление позволяет передавать сообщения между сетями только на основании номера сети, а номер узла используется только после доставки сообщения в нужную сеть; точно так же, как название улицы используется почтальоном только после того, как письмо доставлено в нужный город. В последнее время, чтобы сделать маршрутизацию в крупных сетях более эффективной, предлагаются более сложные варианты числовой адресации, в соответствии с которыми адрес имеет три и более составляющих. Такой подход, в частности, реализован в новой версии протокола IPv6, предназначенного для работы в сети Internet. В современных сетях для адресации узлов применяются, как правило, одновременно все три приведенные выше схемы. Пользователи адресуют компьютеры символьными именами, которые автоматически заменяются в сообщениях, передаваемых по сети, на числовые номера. С помощью этих числовых номеров сообщения передаются из одной сети в другую, а после доставки сообщения в сеть назначения вместо числового номера используется аппаратный адрес компьютера. Сегодня такая схема характерна даже для небольших автономных сетей, где, казалось бы, она явно избыточна - это делается для того, чтобы при включении этой сети в большую сеть не нужно было менять состав операционной системы.

Проблема установления соответствия между адресами различных типов, которой занимается служба разрешения имен, может решаться как полностью централизованными, так и распределенными средствами. В случае централизованного подхода в сети выделяется один компьютер (сервер имен), в котором хранится таблица соответствия друг другу имен различных типов, например символьных имен и числовых номеров. Все остальные компьютеры обращаются к серверу имен, чтобы по символьному имени найти числовой номер компьютера, с которым необходимо обменяться данными.

При другом, распределенном подходе, каждый компьютер сам решает задачу установления соответствия между именами. Например, если пользователь указал для узла назначения числовой номер, то перед началом передачи данных компьютер-отправитель посылает всем компьютерам сети сообщение (такое сообщение называется широковещательным) с просьбой опознать это числовое имя. Все компьютеры, получив это сообщение, сравнивают заданный номер со своим собственным. Тот компьютер, у которого обнаружилось совпадение, посылает ответ, содержащий его аппаратный адрес, после чего становится возможным отправка сообщений по локальной сети.

Распределенный подход хорош тем, что не предполагает выделения специального компьютера, который к тому же часто требует ручного задания таблицы соответствия

имен. Недостатком распределенного подхода является необходимость широковещательных сообщений - такие сообщения перегружают сеть, так как они требуют обязательной обработки всеми узлами, а не только узлом назначения. Поэтому распределенный подход используется только в небольших локальных сетях. В крупных сетях распространение широковещательных сообщений по всем ее сегментам становится практически нереальным, поэтому для них характерен централизованный подход. Наиболее известной службой централизованного разрешения имен является служба Domain Name System (DNS) сети Internet.

#### **6.4. Процедура входа в сеть. Общий диск (дисковое пространство)**

Если рабочее место пользователя подключено к сети и предварительные настройки сделаны, то как правило, для того чтобы войти в сеть никаких дополнительных действий производить не нужно. Тем не менее часто возникает необходимость обмениваться файлами с компьютерами работающими под управлением ОС Windows. Если в сети есть папки, к которым открыт общий сетевой доступ, то это возможно.

В этом поможет пакет **Samba**. Название Samba происходит от названия протокола SMB (Server Message Block), он же NetBIOS, на котором основана работа Windows-сети. Пакет Samba содержит набор приложений, позволяющих Linux-машине включиться в работу такой сети — как в роли клиента, так и в роли сервера.

В роли обычного пользователя вам достаточно уметь подключаться к Windows-сети как клиенту. Для этого вам понадобятся пакеты **samba-common** и **samba-client**, входящие в большинство современных дистрибутивов. Если их у вас нет, то загляните на <http://www.Samba.org>.

Также может понадобиться дать доступ к своим локальным папкам. Для этого обычно используется служба NFS.

**NFS** (расшифровывается как Network File System) — это протокол, разработанный Sun Microsystems для разделения ресурсов (файлов и каталогов) локальной сети. NFS-клиент “монтирует” файловую систему, “экспортируемую” NFS-сервером. Смонтированная таким образом файловая система представляется на клиентском компьютере как часть локальной файловой системы.

Как правило, все необходимые пакеты уже установлены и остается только настроить данный вид доступа.

Для того чтобы предоставить доступ к директориям Вашего компьютера откройте **К - Настройка - Центр управления KDE - Сеть и интернет - Совместное использование файлов** и поставьте галочку напротив пункта **Разрешить общие файлы**

в локальной сети. Выберите **Упрощенный режим**, если Вы хотите разрешить доступ к Вашей домашней директории без ввода паролей.

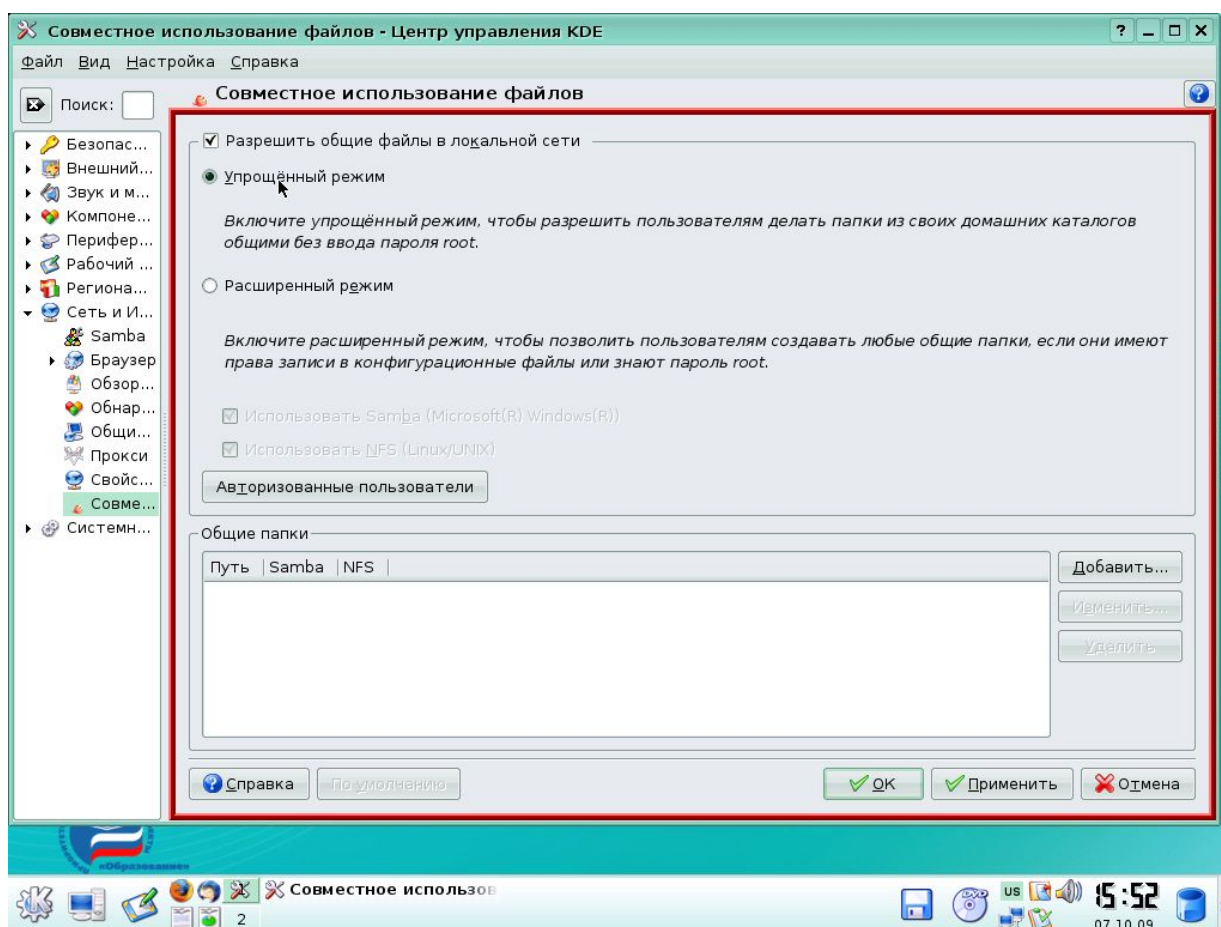


Рис.5. Упрощенный режим

Выберите **Расширенный режим**, чтобы ограничить доступ к ресурсам Вашего компьютера. После выбора Расширенного режима Вы можете добавить пользователей, которые получают доступ к Вашей машине, щелкнув по кнопке **Авторизованные пользователи**.



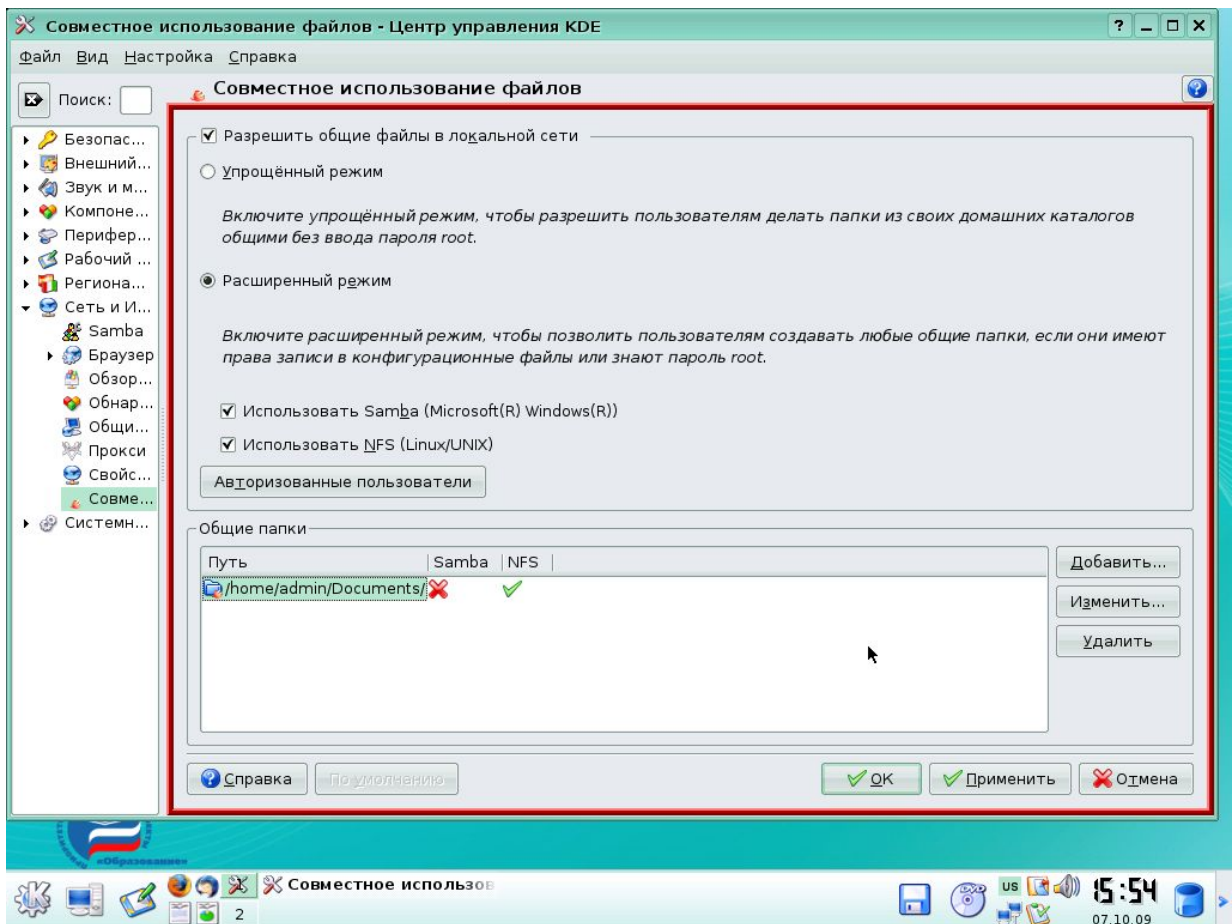


Рис.6. Расширенный режим

Также все это можно настроить следующим образом: откройте **К - Настройка - Центр управления KDE - Сеть и интернет - Samba**, выберите вкладку **Общие ресурсы** и в появившемся окне щелкните по кнопке **Добавить общий ресурс**. Затем выберите нужную директорию.

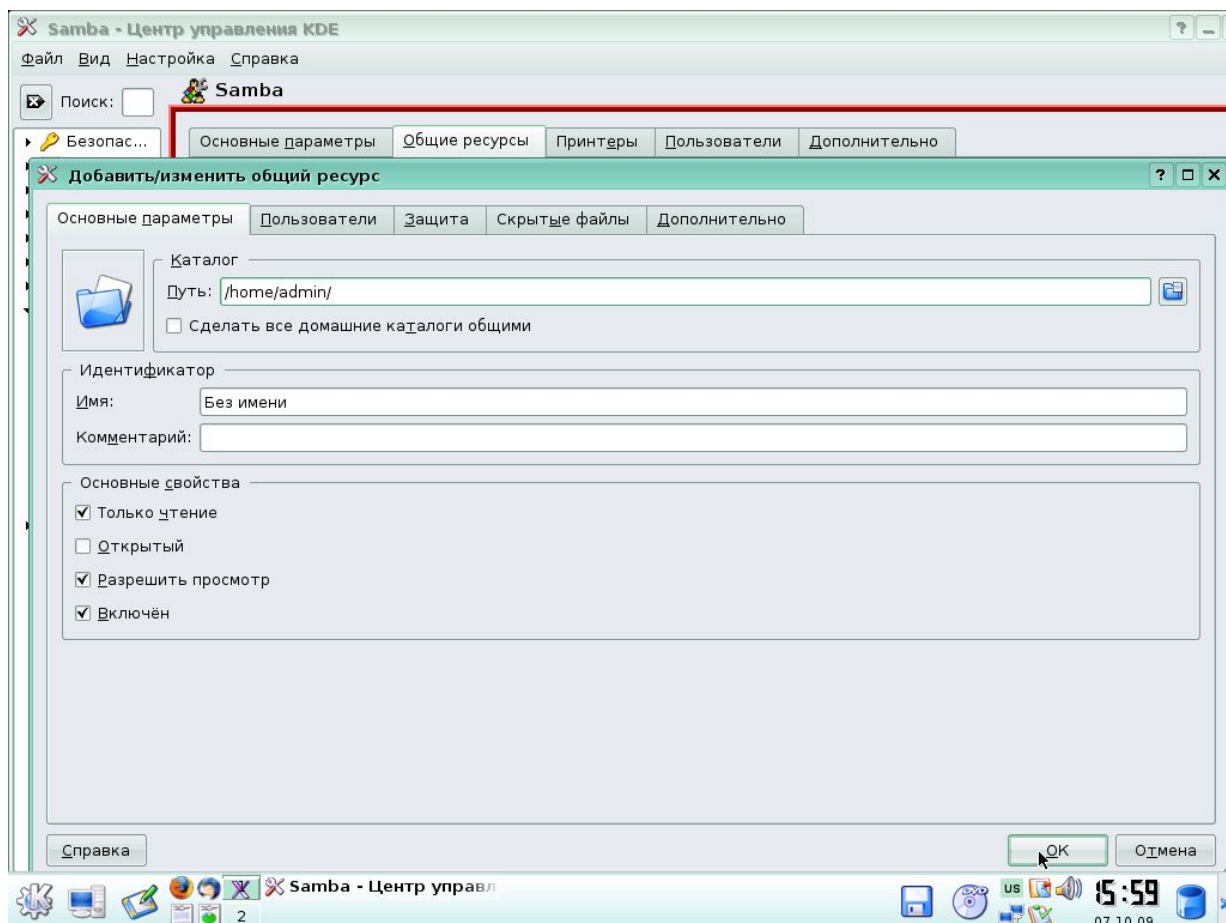


Рис.7. Общие ресурсы Samba

Для того получить доступ к разделяемым директориям удаленного компьютера можно воспользоваться утилитой **К - Настройка - Центр управления KDE - Сеть и интернет - Обзор локальной сети.**

## 6.5. Общий принтер (сетевой принтер)

Электронный документооборот на протяжении нескольких последних лет все больше и больше набирает силу. Однако не менее актуальным остаётся и бумажная, «твёрдая» копия.

Сначала обсудим несколько общих вопросов, которые прольют свет на проблемы печати в Linux.

Инженерная мысль не останавливалась ни на секунду; с момента появления первого печатающего устройства опробована и введена в эксплуатацию масса новых технологий. На данный момент наиболее распространены следующие технологии печати:

1. Матричные принтеры постепенно выводятся из эксплуатации, но, обладая удивительной простотой и надёжностью, продолжают использоваться. С настройкой этого типа принтеров, за редкими исключениями, проблем не должно возникнуть.

2. Лазерные принтеры обладают высокой скоростью печати, незаменимы в офисах организаций всех типов. Моделей существует множество, но, к счастью, для большинства можно найти «родной» драйвер или подобрать совместимый.
3. Главный лозунг струйных принтеров «дешево и в цвете». Так как в этой индустрии не появилось явного лидера, то процветает множество стандартов и существует вероятность иметь очень экзотический с точки зрения Linux принтер.
4. GDI- или Windows-принтеры. Используют компьютер вместо того, чтобы все делать самим. Проблемы очень вероятны, так как протокол работы зачастую закрыт. К счастью, существуют несколько моделей, имеющих интерфейс, напоминающий интерфейс полноценных принтеров, позволяющий все-таки производить печать с низким разрешением.

Где находится драйвер принтера? Если вам доводилось работать в Microsoft Windows, то вы прекрасно знаете, что в комплекте с каждым принтером поставляется дискета или компакт-диск с драйвером для настройки печати. К великому сожалению, в мире UNIX/Linux пока так и не договорились о едином стандарте драйверов. В настоящее время стандартом де-факто является пакет **ghostscript** фирмы Aladdin Software.

**Ghostscript** (далее GS) — это программа перевода из векторного формата в растровый (RIP). На вход программы подаётся документ в формате PostScript или PDF, а на выходе получается документ на языке, понятном конкретной модели принтера или графическое изображение страницы. Большинство программ Linux формируют документ в формате PostScript, поэтому оказывается, что применения GS вполне достаточно для обеспечения печати из приложений.

**PostScript** — это самый настоящий язык программирования, который используется для того, чтобы описать содержимое страницы. Например, пишутся такие серии команд: переместиться туда-то, напечатать слово такое-то, сменить шрифт на такой-то. Существуют модели принтеров, непосредственно понимающие PostScript, для всех остальных требуется посредник, и GS успешно с этим справляется. Дело в том, что количество языков принтеров намного меньше количества существующих моделей. Например, известные всем HP LaserJet 4, 4L, 5, 5L, 6, 6L, 1100, 2100 «понимают» один и тот же язык PCL5.

Итак, теперь понятно, как устроена печать в любом дистрибутиве ПСПО: программа формирует документ на языке PostScript, GS переводит его на язык принтера, установленного в вашей системе и передаёт все принтеру.

Если вам надо послать документ на принтер, установленный на другой машине, или напечатать сразу несколько документов, для этого существует так называемый сервер

печати или спулер (от англ. spooler), обслуживающий очереди печати. Последний термин более точно отражает суть, поэтому им мы и будем пользоваться и писать дальше просто «спулер».

В ПСПО у пользователя есть несколько возможностей, чтобы настроить систему печати. Проще всего запустить программу **kdeprint**, входящую в состав графической среды KDE. Она обладает дружелюбным графическим интерфейсом и в большинстве случаев правильно определяет параметры принтера. Другой путь — настроить систему при помощи CUPS или foomatic (подробно описаны далее). Такая настройка может показаться более сложной, однако вы в итоге с большей вероятностью получите работающий принтер и сможете тонко регулировать параметры печати.

## 6.6. Информационные источники (веб-сайты)

Если в процессе работы возникнут сложности и сбои, очень важно по возможности конкретно сформулировать суть проблемы (вопрос). Поиски ответа стоит начать с документации (локальной и в Интернете), также можно спросить опытных пользователей и обратиться в службу поддержки.

Если среди установленной документации не удалось найти ответа, стоит обратиться к поискам в сети Интернет. Самый первый адрес, куда следует отправиться — сборник наиболее Часто задаваемых ВОпросов (FAQ) по использованию программных продуктов ПСПО. Большинство затруднений при работе с Linux типичны, поэтому здесь с большой вероятностью найдётся ответ на Ваш вопрос.

Кроме того, любому пользователю Интернет доступен поисковый сайт <http://google.com>, наиболее подходящий для поиска чего бы то ни было. Если вы ищете причину конкретной ошибки и способ её устранить, стоит задать в качестве поискового выражения то сообщение об ошибке, которое было выдано системой. Программы с графическим интерфейсом обычно выводят такие сообщения в особых диалоговых окнах, которые появляются поверх основного окна программы и содержат текст сообщения об ошибке и как минимум одну кнопку — «ОК». Если программа была запущена из командной строки, то сообщения о ходе её работы и об ошибках появятся там же. Сведения о событиях, происходящих в системе, всегда можно найти на 12-ой виртуальной консоли (**Ctrl+Alt+F12**), многие сообщения об ошибках тоже туда попадают.

## 6.7. Хранилища файлов (FTP узлы)

Протокол **FTP** (file transfer protocol, протокол передачи файлов) широко используется для обмена файлами в Интернете и локальных сетях. Это специализированный протокол, который предназначен только для передачи файлов и хорошо приспособлен для выполнения этой задачи. К сожалению, изначально протокол

спроектирован таким образом, что пароли, данные и управляющие команды передаются открытым текстом и их можно легко перехватить. Однако это не является проблемой при работе с многочисленными серверами, которые предоставляют только анонимный доступ.

В состав дистрибутива ПСПО входит **vsftpd** (Very Secure FTP Daemon) — полнофункциональный FTP-сервер, позволяющий обслуживать как анонимные запросы, так и запросы от пользователей, зарегистрированных на сервере и имеющих полноценный доступ к его ресурсам. Именно **vsftpd** рекомендован разработчиками дистрибутива для использования в качестве FTP-сервера.

Для работы с FTP-ресурсами достаточно использовать практически любой из файл-менеджеров (**mc**, **Konqueror** и т.п.) в качестве клиента.

Как осуществлять неанонимные передачи по FTP через **Konqueror**? После запуска **Konqueror** (процесс запуска рассматривался в предыдущих главах) необходимо ввести адрес URL, например, `ftp://ftp.linux.armd.ru/master/`, **Konqueror** пытается соединиться с FTP-сервером как анонимный пользователь.

## **II. Программное обеспечение для сжатия и архивирования файлов (Ark)**

### **1. Введение**

Часто бывает удобно работать с группой файлов как с одним целым, например для того, чтобы переслать много файлов по электронной почте. Для этой цели используют специальные программы, которые называются архиваторы. Большинство архиваторов, при создании архивов, уменьшают размер итогового файла, что позволяет вас экономить место на диске.

**Ark** - программа для работы с архивами различных форматов. С помощью **Ark** архивы можно просматривать, извлекать, создавать и изменять. Программа может обрабатывать различные форматы, как-то:

- tar
- gzip
- bzip2
- rar
- zip
- lha

(если установлены соответствующие утилиты командной строки).

Эти способы компрессии отличаются алгоритмами сжатия, которые они используют, и могут на разных данных (текст, картинки, исполняемые файлы, архивы) давать разные результаты.

Особняком стоит утилита tar, ее разрабатывали для того, чтобы можно было данные с файловой системы удобно перенести на другой носитель информации, например на магнитную ленту, которая представляет собой устройство последовательной записи. Поэтому при создании архива tar не уменьшает размер итогового файла. Обычно tar используют совместно с другими архиваторами, например с zip. Полученный архив часто имеет расширение tar.gz или tgz.

При обработке архивов Ark работает совместно с файловым менеджером Konqueror в рабочей среде KDE.

## 2. Сжатие и архивирование файлов с использованием Ark

### 2.1. Открытие архивов

Запустите программу **Ark** из главного меню, пункт **Служебные**, программа **Архиватор (Ark)**.

Для открытия архива в Ark выберите в меню **Файл** пункт **Открыть** и выберите нужный архив.

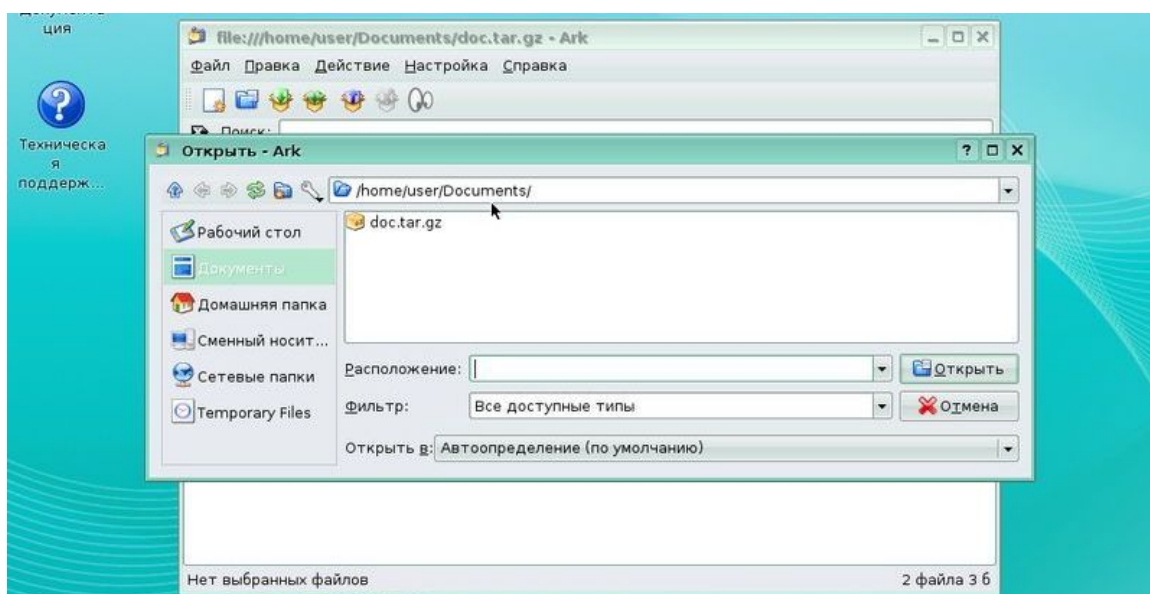


Рис. 1. Открытие архива в Ark

Вы можете открыть файл архива прямо из файлового менеджера, например из Konqueror.

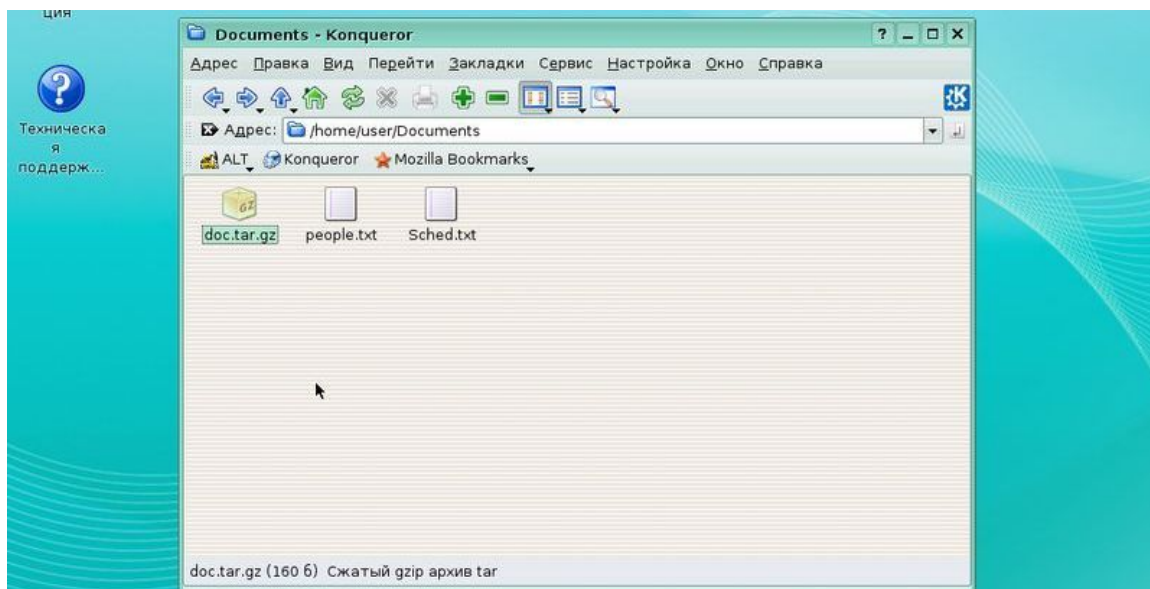


Рис. 2. Открытие архива в Ark из файлового менеджера

Файл архива отображается специальным значком, так что вы можете щёлкнуть правой кнопкой мыши на имени файла и выбрать в контекстном меню **Открыть в Ark**, чтобы открыть его. При этом откроется окно Ark, со списком файлов внутри архива.

## 2.2. Работа с файлами архива

Как только архив открыт, вы можете выполнять различные операции с файлами в нём. Щёлкая правой кнопкой мыши на имени файла в архиве, либо выбирая файл и используя меню **Действие**, вы можете выбрать необходимое действие над этим файлом.

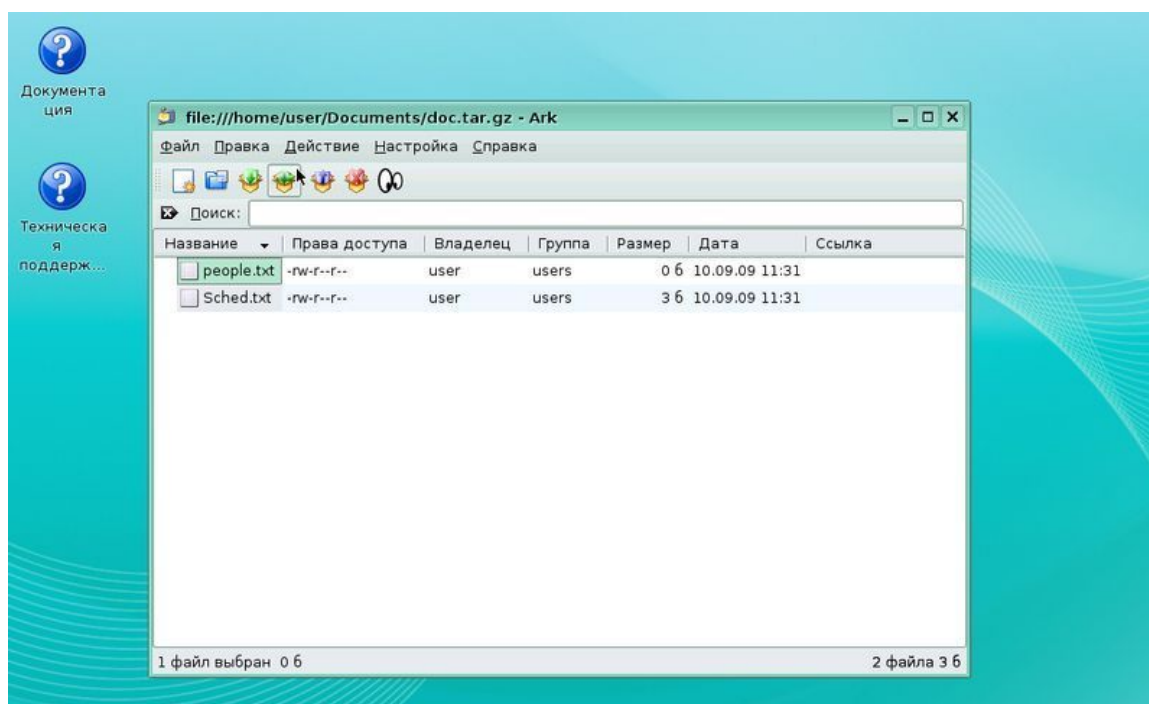


Рис. 3. Работа с файлами в Ark

**Извлечь** - извлекает всё содержимое архива в указанное расположение на диске.

**Удалить** - удаляет выбранные файлы из архива.

**Просмотреть** - открывает файл во встроенном просмотрщике, или во внешней программе, если встроенная компонент просмотра не может открыть файл или выключен.

**Открыть в...** - позволяет открывать файл в программе по вашему выбору.

**Изменить в...** - открывает файл в выбранной программе. Разница с действием Открыть в... состоит в том, что изменённый файл будет сохранён обратно в архиве.

### Извлечение из архива

Как только архив был открыт в Ark, его содержимое может быть извлечено. Для извлечения файлов из архива вы можете выбрать в меню Действие пункт **Распаковать...** либо щёлкнуть правой кнопкой мыши на файле. В диалоге распаковки можно выбрать, куда поместить извлекаемые файлы.

Также вы можете указать, какие именно файлы хотите извлечь:

**Все файлы** - Извлекает всё содержимое архива.

**Выбранные файлы** - извлекает все выделенные файлы.

В поле **Извлечь в:** можно задать папку для сохранения файлов. По умолчанию файл будет извлечён в ту же папку, в которой находится архив. Можно также указать, что по завершении извлечения файла папка должна быть открыта в Konqueror.

## 2.3. Создание архивов и добавление файлов

Для создания нового архива в Ark выберите Создать в меню Файл.

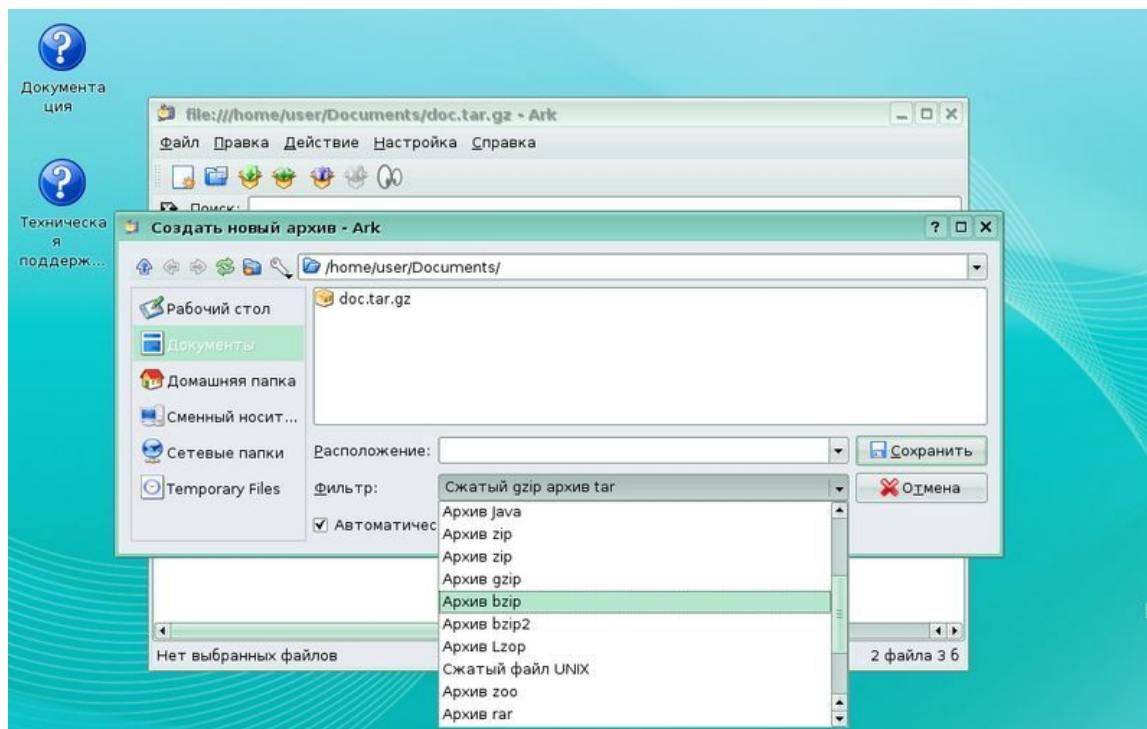


Рис. 4. Создание нового архива в Ark



Здесь вы можете указать имя архива и его тип (tar.gz, zip, bz2 и пр.). В системе должны быть установлены соответствующие архиваторы командной строки, иначе вы получите сообщение об ошибке.

Откроется окно Ark, куда теперь вы должны поместить файлы, которые хотели добавить в архив. Для этого есть несколько вариантов:

Для добавления файлов в архив выберите в меню **Действие** пункт **Добавить файл...**

Удерживая нажатой клавишу Ctrl вы можете выбирать несколько файлов.

Если вы хотите добавить в архив всю папку, выберите в меню **Действие** пункт **Добавить папку...**

Другой способ добавления файлов в архив состоит в переносе файлов мышью из Konqueror в главное окно Ark, при этом перенесённый файл будет добавлен в текущий архив.

В этом же окне можно удалять файлы и папки из архива. После выполнения всех действий закройте Ark либо кнопкой с крестиком в верхней панели программы либо через меню Выход. При этом будет задан вопрос, нужно ли сохранить все изменения, которые вы сделали. Ответьте да, если вы согласны.

## 2.4. Настройка Ark

В панели окна Ark, в меню **Настройка** выберите пункт **Настроить Ark**. Все настройки сгруппированы в 3 раздела.

### 1 - Общие параметры

*Встроенный компонент просмотра* - Использовать встроенный компонент просмотра, если это возможно.

*Включить интеграцию с Konqueror* - Этот модуль позволяет работать с архивами через контекстное меню Konqueror. Опция доступна только в том случае, если установлен модуль интеграции с Konqueror из пакета kdeaddons.

### 2 - Добавление

*Создавать архивы с паролем (Ask for password when create archive if possible)* – вам предложат ввести пароль на архив, если выбранный вами тип архива поддерживает эту возможность.

*Заменять старые файлы новыми* - Если в архиве уже содержится файл с именем добавляемого, заменить его только в том случае, если он более старый по сравнению с добавляемым файлом.

*Использовать короткие имена файлов MS-DOS (Zip)* - Принудительно именовать файлы в формате DOS 8.3. Параметр влияет только на архивы zip.

*Преобразовать LF в DOS CRLF (Zip)* - Конвертировать окончания строк текстовых файлов из формата UNIX в формат DOS. Влияет только на архивы zip

*Сохранять символические ссылки как есть (Zip, Rar)* - Сохранять символические ссылки, вместо помещения в архив соответствующих файлов. Только для архивов zip и rar.

*Рекурсивно добавлять подпапки (Zip, Rar)* - При добавлении в архив папки также будут добавлены и все её подпапки. Только для архивов zip и rar.

### 3 - Распаковка

Большинство из этих опций используются для особых форматов архивов. Другие форматы могут вести себя по-другому и не поддаваться настройке.

*Заменять файлы (Zip, Tar, Zoo, Rar)* – При распаковке архива заменять все существующие файлы, имена которых совпадают с извлекаемыми.

*Сохранять права (Tar)* - Сохранять владельца, группу и права доступа к файлам. Будьте внимательны, так как это может при извлечении привести к файлам, не принадлежащим никому в вашей системе, или к ситуации, когда после распаковки вы не сможете прочитать файлы.

*Игнорировать имена папок (Zip)* - Извлекать все файлы из архива в одну папку назначения, игнорируя структуру каталогов в архиве.

*Преобразовывать имена файлов в нижний регистр (Zip, Rar), Преобразовывать имена файлов в верхний регистр (Rar)* - Использовать для извлекаемых файлов нижний (или верхний) регистр в имени файла.

## 2.5. Работа с архивами из файлового менеджера

Есть возможность работать с архивом непосредственно в файловом менеджере Konqueror, правда при этом есть некоторые ограничения.

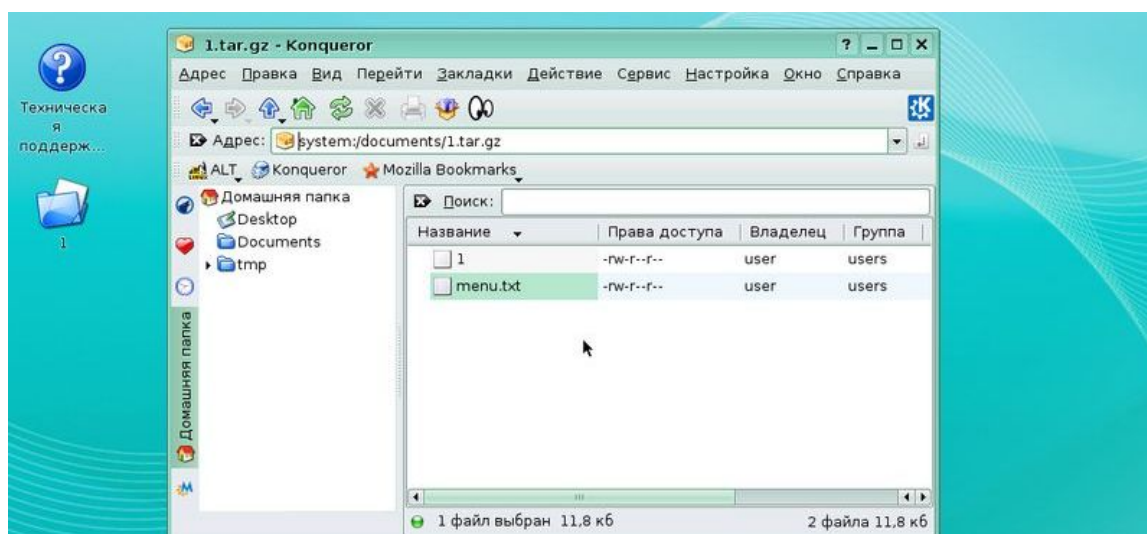


Рис. 5. Файловый менеджер Konqueror

Итак, щелкнув в файловом менеджере по архиву, вы увидите его содержимое. Любой файл из архива вы можете просмотреть, щелкнув по нему.

По правой кнопке мыши доступно две команды: Распаковать и Просмотреть.

Итак, из файлового менеджера вы можете только смотреть. Никакие действия по изменению архива вам недоступны. Вы не можете изменять содержимое файлов, добавлять или удалять файлы из архива, переименовывать их.

Для выполнения этих действий вам придется извлечь файлы из архива, проделать нужные действия и заново создать архив. Полноценно работать с архивом можно из программы Ark.