

Федеральное агентство по образованию  
Государственное образовательное учреждение  
высшего профессионального образования  
Московский государственный университет леса

---

А. В. Чернышов

## **РАБОТА ЗА X-ТЕРМИНАЛОМ**

Учебное пособие  
Для студентов специальностей  
552800 (230100), 220100 (230101)

Издательство Московского государственного университета леса  
Москва — 2005

УДК [004.451.9Linux+004.51](075.8)  
Ч-49

**Ч-49 Чернышов А. В.** Работа за X-терминалом: Учебное пособие для студентов специальности 552800 (230100), 220100 (230101). — М.: МГУЛ, 2005. — 87 с.

*В настоящем учебном пособии приводится общая информация, необходимая студентам для выполнения лабораторных работ в специализированном учебном классе X-терминалов кафедры Вычислительной техники независимо от специальности и изучаемого курса.*

Разработано в соответствии с Государственным образовательным стандартом ВПО 2000 г. для подготовки специалистов по направлению 654600 на основе примерных программ дисциплин «Операционные системы» и «Системное программное обеспечение» для специальностей 552800 (230100) «Информатика и вычислительная техника», 220100 (230101) «Вычислительные машины, комплексы, системы и сети».

Одобрено и рекомендовано к изданию в качестве учебного пособия редакционно-издательским советом университета

Рецензент — доцент кафедры ПМ Д. В. Виноградов

Кафедра вычислительной техники

Автор — Александр Викторович Чернышов, доцент

© Чернышов А. В., 2005

© Московский государственный университет леса, 2005

## ПРЕДИСЛОВИЕ

Специализированный класс X-терминалов создан кафедрой Вычислительной техники для обеспечения проведения лабораторных работ студентами разных специальностей по широкому кругу читаемых курсов, среди которых «Операционные системы», «Системное программное обеспечение», «Применение сетей ЭВМ» и некоторые другие. Кроме этого, вычислительная установка класса может использоваться студентами для научно-исследовательской и учебно-исследовательской работы.

Особенностью класса является применение на вычислительной установке операционной системы Linux, что, кроме всего прочего, позволяет студентам расширить свой кругозор в области применения вычислительной техники и получить необходимые для практической деятельности навыки.

При выполнении лабораторных работ в специализированном классе X-терминалов студенты разных специальностей и курсов сталкиваются с необходимостью в короткий срок познакомиться с основными понятиями и приёмами работы в ОС Linux.

На сегодняшний день на рынке имеется большое число книг, посвящённых работе в этой ОС, но очень сложно найти книгу, в которой материал был бы изложен компактно и с прицелом на быстрое практическое получение минимально необходимых для работы навыков.

Кроме того, выполнение лабораторных работ в классе X-терминалов имеет свою специфику, которую студентам также необходимо знать для эффективной работы, но узнать о которой из имеющихся на рынке книг, естественно, невозможно.

Данное учебное пособие призвано решить названные проблемы. В нём приводятся общие сведения, необходимые для выполнения лабораторных работ, а также использования системы X-терминалов в

учебных и научных целях. Описания конкретных лабораторных работ в данной книге не приводятся — студенты должны получать их из соответствующих лабораторных практикумов.

В первой главе учебного пособия описаны принципы построения вычислительной системы класса X-терминалов и её состав.

Во второй главе описано рабочее окружение пользователя — домашний каталог и его наполнение, — а также возможности пользователя по подстройке внешнего вида графической оболочки и порядок действий для этого.

Третья глава посвящена описанию команд и правил их применения, необходимых для выполнения стандартных действий пользователя в системе X-терминалов.

В приложении дана неполная сводка основных команд, позволяющая пользователям быстро найти название команды, выполняющей то или иное действие в системе.

## Глава 1

### СИСТЕМА X-ТЕРМИНАЛОВ

X-терминалом называется графический терминал, реализующий аппаратным, либо программным способом интерфейс с системой XWindow (X-протокол). При этом X-терминал не является самостоятельным вычислительным средством, а лишь выполняет следующие функции:

- выводит на экран графические окна и обновляет в них информацию в соответствии с запросами системы XWindow;
- управляет работой клавиатуры и мыши, подключёнными к нему (в частности, перемещает по экрану курсор мыши в соответствии с движениями мыши);
- формирует запросы к системе XWindow при нажатии кнопки клавиатуры, либо кнопки (кнопок) мыши.

Программы, ориентированные на работу в системе XWindow, осуществляют весь свой графический ввод/вывод исключительно через программный интерфейс системы XWindow и не имеют ни малейшего понятия не только об аппаратных особенностях устройства вывода, но даже о том, где физически находится это устройство.

Использование системы XWindow позволяет, в частности, работая на одном компьютере сети, запустить на другом компьютере (например, более мощном) X-приложение, вывод окон которого будет направлен на терминал третьего компьютера (например, обладающий лучшими графическими возможностями).

#### Построение системы

Система X-терминалов в учебном классе построена на базе устаревших ПЭВМ, в основном типа IBM PC/486. Терминальные ПЭВМ посредством локальной сети 10 Мбит соединены с сервером средней мощности, на котором установлена ОС Linux.

На терминальных ПЭВМ установлена ОС MS DOS. В качестве одной из программ на каждую ПЭВМ установлен начальный загрузчик ОС Linux и образ её ядра.

Для превращения ПЭВМ в X-терминал на ней достаточно запустить X-сервер, взаимодействующий с сервером Linux по X-протоколу. Теоретически такой X-сервер может выполняться под управлением любой ОС. В данном случае X-сервер запускается под управлением ОС Linux.

Все файлы, необходимые для загрузки ОС Linux и X-сервера (кроме образа ядра ОС), находятся на диске сервера в специальном подкаталоге, который доступен для всех ПЭВМ для монтирования по протоколу nfs. В результате каждая ПЭВМ при загрузке в режиме X-терминала получает в своё распоряжение сетевой диск, на котором находятся все необходимые для загрузки файлы.

Для нормального функционирования ядра ОС Linux на X-терминальной ПЭВМ необходимо наличие файла свопинга. Такой файл создаётся в процессе загрузки системы на локальном диске ПЭВМ.

После окончания загрузки X-терминала все ресурсы терминальной ПЭВМ (процессор, оперативная память и т. п.) используются для выполнения программы X-сервера. Эта программа не требует сверхбольшой производительности процессора и вполне удовлетворительно работает на 4 Мб оперативной памяти. Именно поэтому в качестве X-терминалов могут использоваться устаревшие ПЭВМ.

Все прикладные программы, запускаемые пользователем, работающим на X-терминале, на самом деле запускаются и выполняются на сервере.

Все X-терминалы, подключённые к системе, с точки зрения возможностей работы конечного пользователя равнозначны (с поправкой на индивидуальные характеристики графических адаптеров, установленных в ПЭВМ).

К серверу системы подключён матричный принтер, доступный всем пользователям для печати текстовой и графической информации. В составе сервера имеются также накопители для носителей информации нескольких типов, позволяющие пользователям вводить в систему собственные файлы с информацией для обработки и получать копии файлов с результатами работы.

Дополнительно сервер системы посредством отдельного компьютера-шлюза подключён к локальной сети МГУЛ с возможностью доступа к её открытым ресурсам и выхода в сеть Internet.

### **Возможности, предоставляемые системой**

Система X-терминалов учебного класса представляет собой однопроцессорную многопользовательскую вычислительную установку, функционирующую в режиме разделения времени под управлением ОС Linux.

Каждый пользователь системы X-терминалов имеет свой домашний каталог, в котором находятся все его файлы. Каталог находится

на диске сервера системы и доступен пользователю при входе с любого X-терминала.

В своём домашнем каталоге пользователь может создавать, копировать, переименовывать (перемещать), удалять файлы по своему желанию. Он может сделать файлы доступными или недоступными для других пользователей системы. Он может также пользоваться доступными файлами из каталогов других пользователей.

**ВНИМАНИЕ!!!** Обслуживающий персонал сервера не несёт ответственности за сохранность файлов пользователей! По окончании каждого сеанса работы пользователям рекомендуется создавать резервные копии своих важных файлов на дискетах 3,5” или других доступных в системе внешних носителях данных.

На диске также специально выделен каталог `/home/public`, доступный всем студентам на чтение (в домашнем каталоге каждого студента на этот каталог имеется ссылка `public`). В этом каталоге хранятся файлы, необходимые всем студентам для выполнения лабораторных работ.

Установленное в системе X-терминалов программное обеспечение позволяет решать следующие задачи:

- ввод текста, его форматирование и различная обработка, а также печать;
- ввод (разработка), трансляция, запуск и отладка программ;
- обработка и анализ массивов информации;
- разработка и отладка Web-сайтов;
- разработка и отладка баз данных;
- разработка и отладка интерфейсов к базам данных по технологии Intranet;
- просмотр и преобразование (без редактирования) графических файлов различных форматов;
- посылка локальных почтовых сообщений между пользователями системы;
- групповая работа над проектами;
- сбор и обработка статистической информации о функционировании системы;
- обучение приёмам администрирования и оперативного управления многопользовательской вычислительной системой;
- групповые игры.

В качестве дополнительных (ограниченных) возможностей выступают:

- получение информации из локальной сети МГУЛ;

– поиск и получение информации из сети Internet при условии соответствующей авторизации пользователя в Интернет-Центре МГУЛ.

### **Ограничения системы**

Ограничения системы являются следствием ограниченности, в основном, аппаратных ресурсов сервера. Так как все прикладные процессы всех пользователей выполняются на одном сервере, их суммарные потребности в ресурсах (прежде всего в объёме оперативной памяти) не должны превышать возможности сервера.

Так в системе нежелательно запускать сложные графические редакторы (типа GIMP), современные офисные пакеты (типа OpenOffice), многофункциональные клиенты Web (типа Netscape или Mozilla) более, чем двумя пользователями одновременно.

В связи с ограниченностью ёмкости жёсткого диска сервера каждый пользователь должен стремиться к минимизации места на диске, занимаемого его личными файлами. При необходимости записи на диск сервера файла с массивом данных большого объёма (более 500 КБайт) необходимо согласовать этот вопрос с администратором (дежурным оператором) сервера. Обработку таких файлов желательно выполнить за минимальное количество сеансов работы, после чего эти файлы сразу удалить с диска.

К ограничениям, накладываемым ОС, относится невозможность для рядового пользователя самостоятельного взаимодействия с компьютерами локальной сети МГУЛ по протоколу SMB (стандартный сетевой протокол ОС Windows). При необходимости соединения по этому протоколу с каким-либо компьютером необходимо обратиться к администратору (дежурному инженеру) сервера.

### **Программное обеспечение, установленное в системе**

В системе X-терминалов пользователям доступно следующее программное обеспечение:

- широкий набор стандартных команд и фильтров ОС UNIX (ls, cat, sort, sed, awk, grep и т. п.);
- сервер Web Русский Apache;
- СУБД PostgreSQL;
- языки программирования Ассемблер (GNU), Си, Фортран, Perl и некоторые другие;
- система создания динамических страниц Web PHP4;



- система анализа данных и моделирования octave;
- система манипулирования графическими изображениями ImageMagick;
- система вёрстки сложных документов T<sub>E</sub>X;
- средства для просмотра и печати документов в форматах ps и pdf;
- некоторые специальные системы, фильтры и утилиты.

Имеющееся на сервере системы программного обеспечения позволяет решать широкий круг задач как учебного, так и научного плана.

## Глава 2

### СТАНДАРТНОЕ ОКРУЖЕНИЕ ПОЛЬЗОВАТЕЛЯ

Под стандартным окружением здесь понимаются возможности, предоставляемые пользователю по созданию и управлению своими файлами в создаваемом для каждого пользователя домашнем каталоге, доступу к файлам других пользователей и системы в целом, а также по выбору менеджера окон и его настройке под желания и нужды пользователя.

#### Домашний каталог пользователя

Домашний каталог создаётся для каждого пользователя при регистрации в системе. При каждом входе в систему пользователь всегда оказывается в своём домашнем каталоге. При этом полное имя домашнего каталога пользователя в файловой системе выглядит как `/home/имя/`, где `имя` — регистрационное имя пользователя. Для ссылки на домашний каталог из командной строки в системе предусмотрено сокращение — `~/`

При необходимости быстро вернуться в свой домашний каталог из любого другого каталога системы достаточно в командной строке дать команду `cd` (без аргументов).

В своём домашнем каталоге (и только в нём!) пользователь может создавать, изменять и удалять любые файлы и подкаталоги.

**ВНИМАНИЕ!** Во всех других каталогах системы пользователь может только просматривать файлы, для которых установлены соответствующие права доступа.

Подробнее о работе в системе можно прочитать во множестве книг, посвящённых UNIX и Linux. Для примера укажем на великолепно написанную книгу [1].

#### *Файлы в домашнем каталоге*

В начальный момент времени в домашнем каталоге пользователя присутствуют несколько предопределённых файлов и подкаталогов. Все предопределённые файлы и некоторые подкаталоги являются скрытыми (их имена начинаются с точки). Их список можно просмотреть стандартной командой `ls -la`.

Практически все эти файлы и подкаталоги доступны пользователю для изменения. Но вносить в них изменения без явной необходимости не рекомендуется.

Кроме того, в домашнем каталоге имеется два обычных (не скрытых) подкаталога, связанных с особенностями учебной системы.

## ***Подкаталог WWW***

Подкаталог предназначен для размещения в нём файлов страниц Web, разрабатываемых пользователем. Файлы могут быть как обычными файлами HTML, так и файлами PHP. Допустимы также графические файлы в форматах jpeg и png.

Текстовые файлы (HTML, PHP) должны иметь кодировку KOI8-R.

Обращение к странице через сервер Web выполняется через URL `http://localhost/~имя/файл.HTML`, где *имя* — регистрационное имя пользователя, *файл.HTML* — имя файла запрашиваемой страницы.

## ***Подкаталог public***

Этот подкаталог является ссылкой на подкаталог `/home/public` и предназначен для хранения общедоступных учебных, методических и других материалов.

Все пользователи студенты могут только читать файлы в этом подкаталоге.

Пользователи преподаватели имеют право записи в этот подкаталог для размещения в системе своих учебных и методических материалов.

## **Настройка рабочего окружения**

В отличие от ОС Windows, где внешний вид рабочего стола экрана, окон и элементов управления ими, а также способы управления окнами заданы самой операционной системой и не могут быть изменены (не считая цвета и формы кнопок, что не меняет сути дела), в системе X-терминалов всё управление окнами обеспечивается специальным прикладным процессом — так называемым менеджером окон. Именно этот процесс определяет вид рабочего стола, окон и элементов управления ими, а также и сами способы взаимодействия с системой.

Пользователь может по собственному усмотрению выбрать любой менеджер окон из тех, которые установлены в системе. Кроме этого пользователь может задать список и расположение задач, запускающихся по умолчанию при входе в систему. Само собой разумеется, что если в менеджере окон поддерживается режим выбора темы, то пользователь может установить себе любую тему, в том числе созданную самостоятельно.

Информация обо всех настройках сохраняется в файлах в домашнем каталоге пользователя и на настройки рабочего окружения других пользователей никак не влияет.

## *Выбор оконного менеджера*

По умолчанию в системе загружается менеджер окон IceWM. Это лёгкий, компактный менеджер окон, предоставляющий дополнительные возможности рабочего стола в виде настраиваемой панели задач (такие возможности в других менеджерах окон обеспечиваются отдельными программами) с возможностью выбора темы и настройки всех его элементов. Наличие панели задач делает работу в нём похожей на работу в ОС Windows.

Для смены менеджера окон необходимо щёлкнуть курсором мыши на кнопку в левом нижнем углу на панели задач и в открывшемся меню выбрать пункт **Window Managers**, а во вновь открывшемся списке — желаемый менеджер окон.

Не все оконные менеджеры одинаково удобны в работе. Встречаются поистине ужасные варианты, разрабатывавшиеся для специальных целей. Поэтому при выборе рекомендуется придерживаться следующего списка:

- BlackBox;
- AfterStep и его варианты;
- fwvm2 и его варианты;
- twm.

Популярные в последнее время менеджеры KDE и Gnome являются на самом деле сложными рабочими средами, включающими помимо собственно менеджеров окон большое число сопутствующих программ. Они очень требовательны к ресурсам и в многопользовательской системе X-терминалов неэффективны.

После удачной смены менеджера окон внешний вид рабочего стола радикально изменяется. Необходимо иметь в виду, что в новом менеджере окон панели задач может не быть совсем, либо она не будет предназначена для запуска программ и вызова меню.

Традиционным способом вызова меню задач (в том числе для запуска программы терминала, из командной строки которой можно запустить любую программу, а также для попытки выбрать другой менеджер окон) является щелчок на пустом месте экрана (иногда с удержанием) левой или правой кнопки мыши — рекомендуется поэкспериментировать, так как конкретные действия зависят от загруженного менеджера окон.

После окончания сеанса работы, выхода из системы и повторного входа снова будет загружен менеджер окон по умолчанию — IceWM. Для того, чтобы по умолчанию загружался другой менеджер окон

пользователю необходимо в своём домашнем каталоге создать файл с именем `.xsession`, в котором в самом простейшем случае нужно указать имя программы желаемого менеджера окон. Дополнительно в этом же файле могут быть указаны команды чтения дополнительных баз данных ресурсов системы XWindow и запуска программ по умолчанию. В этом случае все программы (кроме менеджера окон) должны запускаться в фоновом режиме (со знаком `&` на конце), а сама программа менеджера окон должна быть указана в файле последней и без символа `&`.

Для многих менеджеров окон возможна «тонкая» настройка на нужды и предпочтения конкретного пользователя. Но здесь далее будет рассмотрена только настройка менеджера окон, принятого в системе по умолчанию — IceWM. Пользователей, желающих работать в других менеджерах окон, отошлём к документации, сопровождающей систему, и к книге [2].

### *Настройка IceWM*

Менеджер окон IceWM имеет разветвлённые возможности по настройке своего внешнего вида. Здесь будут рассмотрены только наиболее полезные с практической точки зрения возможности. Если пользователю этих возможностей окажется мало, он может изучить оставшиеся возможности самостоятельно по входящей в систему документации.

Многие (но не все) возможности настройки предоставляет утилита `icewmconf`.

Любые работы по подстройке менеджера IceWM надо начинать с создания в своём домашнем каталоге подкаталога `.icewm` (имя начинается с точки)

```
mkdir ~/.icewm
```

в котором и будут храниться все файлы, определяющие индивидуальные параметры настроек пользователя.

**Настройка темы рабочего стола.** Представляет собой выбор внешнего вида панели задач, кнопок и меню, рамок и управляющих элементов окон приложений.

Для выбора новой темы достаточно щёлкнуть курсором мыши по кнопке в левом нижнем углу на панели задач (кнопка главного меню менеджера окон) и в открывшемся меню выбрать пункт **Темы** (кнопки мыши при этом нажимать не надо). В открывшемся вслед

за этим меню второго уровня будут представлены имеющиеся в системе темы. Для выбора любой из них достаточно щёлкнуть по ней курсором мыши.

Новая тема установится только до конца текущего сеанса работы пользователя. При следующем входе в систему будет выбрана тема по умолчанию.

Чтобы сделать выбранную тему загружаемой по умолчанию необходимо воспользоваться услугами конфигурационной программы `icwpref`. Среди закладок её меню есть закладка выбора темы. Список тем соответствует списку тем в главном меню менеджера окон. После выбора нужной темы необходимо сохранить установленную конфигурацию, нажав кнопку `Save` в нижней части окна программы. Новая тема при этом не устанавливается. Она будет автоматически загружена лишь после повторного входа пользователя в систему.

**Настройка пунктов главного меню.** В последних версиях IceWM для этой цели может использоваться утилита `iceme`. Но в более ранних версиях менеджера такая утилита отсутствовала. Поэтому рассмотрим подстройку главного меню «традиционным» способом.

В менеджере окон IceWM файл, описывающий всю структуру главного меню по умолчанию (для всех), хранится в каталоге `/etc/X11/icewm` и называется `menu`. Этот файл является текстовым и имеет понятную структуру.

Для создания собственного варианта главного меню необходимо скопировать этот файл в подкаталог `.icewm` своего домашнего каталога

```
cp /etc/X11/icewm/menu ~/.icewm
```

и исправить содержимое этой копии по своему усмотрению в любом текстовом редакторе.

Новое меню будет активировано при следующем входе пользователя в систему.

**Настройка панели задач (Taskbar).** Внешний вид часов текущего времени, индикатор почтового ящика, а также индикаторы нагрузки процессора и сети настраиваются посредством утилиты `icwpref`.

Далее описывается настройка, позволяющая расположить на панели задач кнопки быстрого запуска некоторых приложений.

По умолчанию перечень и внешний вид кнопок быстрого запуска, а также соответствующие кнопкам команды запуска описаны в файле

`toolbar`, расположенном в каталоге `/etc/X11/icewm`. При этом на панели задач реально появляются только те кнопки, программы для которых реально установлены в системе. Файл `toolbar` является текстовым и имеет простую структуру.

Для создания собственного варианта расположения кнопок на панели задач необходимо скопировать этот файл в подкаталог `.icewm` своего домашнего каталога

```
cp /etc/X11/icewm/toolbar ~/.icewm
```

и исправить содержимое этой копии по своему усмотрению в любом текстовом редакторе.

Новая конфигурация кнопок на панели задач вступит в силу при следующем входе пользователя в систему.

При правке файла единственную сложность может представлять лишь подбор иконок для изображения кнопок на панели задач. Эта проблема рассматривается далее.

**Выбор иконок для менеджера IceWM.** В IceWM принята весьма своеобразная система работы с иконками. Менеджер окон потенциально (для разных целей) может использовать каждую иконку в трёх размерах:  $16 \times 16$ ,  $32 \times 32$  и  $48 \times 48$  точек. Причём выбор иконки конкретного размера выполняется на основании явно указанного размера в имени иконки.

Для упрощения понимания этого процесса приведём пример. Пусть для гипотетического приложения с именем `pr1` создана иконка с именем `ipr1` и это имя иконки указано менеджеру IceWM для отображения в элементах управления (в меню, панели задач и т. п.).

На самом деле для отображения иконки `ipr1` менеджер IceWM будет искать на диске файлы со следующими именами:

- для размера  $16 \times 16$  точек — `ipr1_16x16.xpm`;
- для размера  $32 \times 32$  точек — `ipr1_32x32.xpm`;
- для размера  $48 \times 48$  точек — `ipr1_48x48.xpm`.

Файлы именно с такими именами должны присутствовать на диске, иначе иконка для соответствующего размера просто не будет найдена.

Размер иконки выбирается в зависимости от настроек IceWM. Для главного меню и панели задач, настроенных по умолчанию, всегда используются иконки  $16 \times 16$  точек. При этом файлы с иконками другого размера на диске могут не присутствовать.

Небольшой набор иконок для IceWM по умолчанию хранится в каталоге

```
/usr/X11R6/lib/X11/icewm/icons
```

и именно в этом каталоге IceWM ищет по умолчанию все свои иконки.

Пользователь может создать свои собственные иконки. Как уже было показано, эти иконки должны иметь названный выше размер, формат `xpm` и имя в стандарте IceWM. В большинстве случаев требуются иконки только в размере  $16 \times 16$  точек. Размещать их надо в подкаталоге `icons` подкаталога `.icewm` домашнего каталога пользователя (полный путь `~/.icewm/icons/`). Например, в случае нашего гипотетического примера полный путь и имя файла иконки должны выглядеть так:

```
~/.icewm/icons/ipr1_16x16.xpm
```

Использование иконок пользователя в файлах конфигурации ничем не отличается от использования иконок по умолчанию. Для окончания примера покажем, как должна выглядеть строка файла `toolbar`, описывающая кнопку быстрого вызова нашей программы `pr1`:

```
prog "Программа pr1" "ipr1" pr1
```

В строке описания четыре поля. Первое — ключевое слово `prog`; второе — подсказка, всплывающая при наведении курсора мыши на кнопку; третье — ключевая часть имени иконки; четвертое — собственно команда запуска программы, может дополнительно содержать ключи командной строки.

### *Запуск файлового менеджера рабочего стола*

В менеджере окон IceWM не предусмотрена возможность размещения ссылок на программы, папки и файлы в виде иконок на рабочем столе. Строго говоря, для терминалов с низким разрешением, которыми оснащён учебный класс, такая возможность скорее вредна и лишь мешает в работе. Да и для терминалов с высоким разрешением ценность такой возможности весьма сомнительна, и для многих пользователей является скорее делом привычки, чем необходимостью.

Однако для желающих работать «как в MS Windows» в ОС Linux на сегодняшний день имеется несколько вариантов таких программ, получивших общее название «файловые менеджеры».

В системе X-терминалов установлен файловый менеджер `dfm`. Его запуск можно в любой момент осуществить из командной строки окна терминала, введя команду

```
dfm
```

Если же есть желание запускать его по умолчанию при каждом входе в систему, то можно добавить его запуск в файл `.xsession` непосредственно перед запуском менеджера окон. **ВНИМАНИЕ!** Команда `dfm` должна быть указана без завершающего знака `&`.



После запуска менеджера `dfm` на рабочем столе в левом верхнем углу сразу появляется несколько иконок, заданных по умолчанию. Доступ к главному меню `dfm` выполняется щелчком кнопки мыши на свободном месте рабочего стола.

Остальные подробности работы с файловым менеджером `dfm` пользователям предлагается изучить самостоятельно. При необходимости можно как всегда воспользоваться подсказкой `man`.

### *Настройка свойств задач (widgets)*

Практически для всех оконных приложений, запускаемых в системе XWindow, можно задать значения параметров, определяющие их внешний вид и функциональные возможности. Эти параметры задаются либо через аргументы командной строки (что неудобно при необходимости часто запускать приложение), либо через механизм переопределения значений ресурсов.

Строго говоря, все ресурсы сгруппированы в так называемые виджеты (widgets), образующие древовидную структуру. Но в большинстве случаев для изменения значения ресурса достаточно знать только имя самого ресурса и имя приложения в системе (или имя класса приложений). Сложность здесь заключается в том, что в некоторых случаях надо указывать имя приложения (записывается буквами в том же самом регистре, как и само имя файла приложения), а в некоторых случаях — имя класса приложений (тогда несколько букв будут прописными, а остальные строчными).

Собственно имена всех ресурсов конкретного приложения часто можно получить из руководства `man`.

Для применения механизма переопределения ресурсов пользователь должен создать в своём подкаталоге файл `.Xdefaults`, в который записать новые значения ресурсов для тех приложений, которые ему хочется донастроить.

Приведём пример такого файла. В данном случае выполняется донастройка двух терминальных программ: `xterm` и `rxvt`.

```
XTerm*Font: *-fixed-medium-r-normal--15*-koi8-r
XTerm*Foreground: white
XTerm*Background: black
```

```
rxvt*font: *-fixed-medium-r-normal--15*-koi8-r
rxvt*foreground: white
rxvt*background: black
```

Как можно видеть, выполняется настройка приложений класса **XTerm** и приложения **rxvt** (которое почему-то в класс терминалов не включено). Задаются шрифты (с русскими символами в кодировке **koï8-r**) и цвета текста и фона. Детальное толкование приведённых здесь имён ресурсов, а также имена и назначение других ресурсов можно узнать через команды

```
man xterm
```

и

```
man rxvt.
```

Если пользователь не создавал в своём домашнем каталоге файл **.xsession**, то созданный файл **.Xdefaults** должен считываться автоматически при каждом входе пользователя в систему.

Если же пользователь создал файл **.xsession** в своём домашнем каталоге, то необходимо проследить, чтобы в нём до запуска менеджера файлов была записана команда

```
xrdb -screen -merge $HOME/.Xdefaults
```

Это обеспечит считывание файла ресурсов, причём таким образом, что новые значения ресурсов будут лишь добавлены к обширнейшему списку уже определённых системой ресурсов (перезапишут одноимённые и оставят действительными все другие).

Если чтение файла **.Xdefaults** не выполняется по каким-то другим причинам, придётся создать в своём домашнем каталоге файл **.xsession** со следующим минимальным содержанием:

```
xrdb -screen -merge $HOME/.Xdefaults  
icewm
```

Здесь предполагается, что необходимо запускать менеджер окон **IceWM**.

Для более глубокого знакомства с механизмом управления ресурсами системы **XWindow** рекомендуется прочитать документацию **man** на следующие программы: **X**, **xrdb**, **xprop**, **appres**, **listres**, а также некоторые другие, на которые есть ссылки в документации названных программ.

## *Просмотр шрифтов, доступных в системе*

При изменении значений параметров ресурсов шрифтов часто необходимо знать какие вообще шрифты установлены в системе и из какого набора шрифтов возможен выбор.

В разных системах XWindow могут быть установлены разные наборы шрифтов. С точки зрения поддержки кириллицы о каком-либо «минимальном стандартном» наборе говорить вообще не принято. Поэтому при задании шрифта пользователю просто необходимо ознакомиться с имеющимися в наличии шрифтами.

Необходимо иметь в виду, что хотя в большинстве современных систем XWindow установлены как растровые, так и векторные шрифты, для работы на экране (в частности в программах терминалов) настоятельно рекомендуется выбирать именно растровые шрифты, которые значительно лучше адаптированы к удобному (для глаз) представлению символов на экранах с низким разрешением (72...100 точек на дюйм).

Надо сказать, что обычно в системе XWindow шрифты задаются в специальной форме — так называемой **X Logical Font Description Conventions** (примерный перевод: Соглашения по описанию логических шрифтов системы X). В этой форме каждый шрифт задаётся как прямой перечень параметров (атрибутов) шрифта, заданных в строгом порядке и разделённых знаками «минус» без пробелов.

Часто пользователю не нужно задавать все параметры шрифта, а нужно указать лишь группу шрифтов, удовлетворяющую определённым требованиям (например, чтобы кодировка была «koi8-r»). А из этой группы система должна сама выбрать подходящий шрифт. В этом случае задаётся не шрифт, а шаблон выбора шрифта, в котором нужные параметры заданы явно, а вместо всех остальных стоит знак шаблона — звёздочка (\*). Например, при задании любого шрифта в кодировке koi8-r достаточен такой шаблон

**\*-koi8-r**

(здесь заданы **два** последних параметра шрифта — система кодировки и вариант кодовой таблицы в этой системе). При задании такого шаблона в командной строке надо не забыть взять его в кавычки, чтобы экранировать от замены интерпретатором.

Важно понимать, что далеко не любой шаблон может соответствовать хотя бы одному шрифту в системе. Например, при наличии в системе следующих шрифтов:

```
-cronyx-courier-bold-o-normal--14-100-100-100-m-80-koi8-r
-cronyx-courier-bold-o-normal--17-120-100-100-m-90-koi8-r
-cronyx-courier-bold-r-normal--14-100-100-100-m-80-koi8-r
-cronyx-courier-bold-r-normal--17-120-100-100-m-90-koi8-r
-cronyx-courier-medium-r-normal--14-100-100-100-m-80-koi8-r
-cronyx-courier-medium-r-normal--17-120-100-100-m-90-koi8-r
-cronyx-fixed-medium-r-normal--10-100-75-75-c-60-koi8-r
-cronyx-fixed-medium-r-normal--13-120-75-75-c-80-koi8-r
```

шаблон, приведённый выше, будет соответствовать им всем (любому из них), тогда как шаблон

```
**-courier**-koi8-r
```

только первым шести, а шаблон

```
**-courier**-75**-koi8-r
```

вообще ни одному.

Для просмотра имеющихся в системе шрифтов и состава их символов можно пользоваться следующими командами: `xlsfonts`, `xfd`, `xfontsel`. Последняя команда предоставляет удобное графическое окно, в котором можно вручную смоделировать шаблон шрифта, получив при этом информацию о наличии, внешнем виде и количестве шрифтов в системе, соответствующих заданному шаблону.

## Глава 3

### ВЫПОЛНЕНИЕ ЧАСТО ВСТРЕЧАЮЩИХСЯ ОПЕРАЦИЙ

Большинство операций и команд, описанных в настоящей главе, предназначены для ввода в командную строку ОС. Командная строка доступна пользователю через программу (окно) эмулятора терминала `xterm` или `rxvt` (в дальнейшем для краткости называются просто окнами терминала).

Одновременно может быть запущено несколько окон терминала. Более удобной в работе является программа `rxvt`.

При вводе команды и указании имён файлов необходимо внимательно следить за регистром букв, так как все имена команд и имена файлов (каталогов) являются чувствительными к регистру.

Ввод любой команды должен заканчиваться нажатием клавиши `Enter`. При этом команда выполняется и на экране снова появляется стандартное приглашение интерпретатора команд.

Однако если во введённой строке несбалансированы кавычки или скобки, интерпретатор команд будет считать, что ввод команды не окончен и продолжается на следующей строке. При этом на новой строке появляется укороченное приглашение

>

и ожидается продолжение ввода команды. Когда недостающая часть команды будет введена и снова нажата клавиша `Enter`, вся команда будет выполнена и снова появится стандартное приглашение командной строки.

Если же такая ситуация является ошибкой, то для отмены неверно введённой команды и получения «нормального» приглашения командной строки необходимо нажать клавиши `Ctrl C`.

Восстановить в командной строке предыдущую команду (команды) для последующего выполнения или редактирования можно нажатием клавиш `↑`, `↓`.

Редактировать командную строку можно при помощи клавиш `←`, `→`, `Backspace` и других.

Дополнительно к материалу настоящей главы настоятельно рекомендуется познакомиться с книгами [2–4].

## Вход в систему

Каждый сеанс работы начинается со входа пользователя в систему. Для входа в ответ на приглашение системы (login, Password) необходимо набрать своё регистрационное имя и пароль. При этом пароль вводится «в тёмную» — на экране ничего не отображается. Если всё набрано правильно, то на терминале появляется рабочий экран пользователя (по умолчанию на рабочем экране отображается рабочий стол оконного менеджера IceWM). Если при вводе регистрационного имени и пароля были допущены ошибки, то появляется сообщение **Login incorrect**, и требуется повторить ввод сначала.

Регистрационное имя (называемое также логин или ник) и пароль пользователю присваивает и вводит их в систему системный администратор (дежурный инженер, преподаватель) перед самым первым сеансом работы в системе. Эта процедура называется регистрацией пользователя. Приветствуется, когда пользователь сам придумывает имя для себя и сообщает его администратору. Имя должно начинаться со строчной английской буквы и может состоять только из строчных английских букв и цифр. Длина имени от трёх до десяти символов.

Пароль пользователь должен придумать самостоятельно и ввести его по команде администратора во время процедуры регистрации.

## Выход из системы

Закончив сеанс работы в системе, пользователь должен выйти из системы. Для этого прежде всего необходимо закрыть все запущенные в сеансе работы приложения. После этого закрыть сеанс одним из следующих способов:

а) нажать на клавиатуре терминала одновременно клавиши **Ctrl** **Alt** **Backspace**;

б) с помощью мыши открыть главное меню оконного менеджера (например для IceWM щёлкнуть по экранной кнопке в левом нижнем углу экрана), и выбрать в нём пункт **Logout**. Подтвердить выход в появившемся диалоговом окне.

В результате на экране терминала должно снова появиться приглашение системы (login, Password).

## Выполнение команд в стандартном экране пользователя

По умолчанию пользователю системы X-терминалов предоставляется стандартное рабочее окружение, достаточное для выполнения практически любых работ. Это окружение использует менеджер окон IceWM, обладающий также ограниченными функциями рабочего стола.

После входа в систему на экране отображается рабочий стол. В его нижней части располагается панель задач. На панели задачи слева направо расположены следующие элементы управления и индикации:

- кнопка открытия главного меню;
- кнопка открытия окна списка запущенных приложений;
- иконки быстрого запуска избранных приложений;
- кнопки выбора активного рабочего стола;
- пустое пространство;
- индикатор почтового ящика (может отсутствовать);
- часы.

Общая идея использования панели задач аналогична такому же элементу в ОС MS Windows. Так пустое пространство заполняется индикаторами открытых окон по мере запуска программ, нажатие кнопки главного меню открывает иерархическое меню программ, расположенных по назначению и выполняемым функциям, и т. д.

Описанный далее внешний вид менеджера IceWM и его элементов управления может быть легко изменён и дополнен пользователем под свои нужды и потребности. Описание того, как это сделать, содержится в разделе «Настройка IceWM».

### *Запуск программ из меню*

Для открытия главного меню достаточно щёлкнуть указателем мыши на кнопке главного меню, находящейся в левом нижнем углу экрана на панели задач.

Если мышь по каким-либо причинам не работает, меню можно открыть комбинацией клавиш **Ctrl Esc**.

Выбор нужной программы выполняется перемещением по иерархическим пунктам меню либо с помощью курсора мыши (запуск выбранной программы — щелчком левой кнопки мыши на строке меню с её именем), либо с помощью клавиш клавиатуры **←**, **→**, **↑**, **↓** (запуск программы — нажатием клавиши **Enter** на строке с её именем).

Главное меню разделено на несколько зон. Самая верхняя зона содержит прямые ссылки на несколько наиболее полезных программ. В частности, здесь есть ссылка на программу эмулятора терминала `rxvt`, необходимую для выполнения всех лабораторных работ.

### *Запуск программ из панели задач*

На панель задач выведено несколько иконок наиболее часто используемых программ. Для запуска любой из этих программ необходимо просто навести курсор мыши на выбранную иконку и сделать двойной щелчок левой кнопкой.

По умолчанию на панели находится три иконки программ:

- эмулятора терминала `rxvt`;
- лёгкого клиента `Web chimera`;
- программы просмотра файлов форматов `ps` и `pdf gv` (графического интерфейса к программе `ghostscript`).

При необходимости пользователь может сам добавить на панель иконки необходимых программ. Делается это редактированием текстового файла `~/icewm/toolbar` (см. раздел «Настройка IceWM»).

### *Переключение между окнами запущенных задач*

Простейший способ переключения — использовать комбинацию клавиш `Alt Tab`.

Второй простой способ (но требующий работы с мышью) — щёлкнуть указателем мыши на индикаторе требуемого окна в панели задач.

Можно щёлкнуть курсором мыши по кнопке открытых приложений. и в открывшемся окне выбрать необходимое приложение.

### *Смена активного рабочего стола*

Возможность запуска нескольких программ одновременно и переключения между их окнами в процессе работы — сильная сторона современных многозадачных ОС. Но в условиях учебного класса X-терминалов с ограниченными возможностями по разрешению экранов при запуске уже двух-трёх программ работа сильно усложняется.

Снять проблему позволяет интересное свойство менеджера окон IceWM (встречающееся и во многих других менеджерах), называемое «виртуальными рабочими столами». По умолчанию в распоряжении каждого пользователя имеется четыре рабочих стола. При входе в систему пользователь всегда оказывается на первом рабочем столе.



Рабочий стол, который видит на экране пользователь, называется активным.

Любое приложение может быть запущено на любом рабочем столе и, вообще говоря, может открывать свои окна на любом рабочем столе. Обычно для открытия окон приложение использует активный рабочий стол.

Для смены рабочего стола достаточно щёлкнуть курсором мыши на номере нужного рабочего стола, находящемся на панели задач. С клавиатуры то же самое можно сделать, нажав клавиши `Ctrl Alt n`, где `n` — одна из клавиш `1`, `2`, `3`, `4`, соответствующих номерам рабочих столов.

Предположим, что необходимо открыть два окна терминала. В одном предполагается выполнение команд ОС, а другой нужен для оперативного получения справки (например, командой `man`) по набираемым командам.

Рекомендуется следующий порядок работы.

Предположим, что по умолчанию активным рабочим столом является стол 1.

Любым способом запустить программу эмулятора терминала. Окно программы откроется на текущем (активном) рабочем столе.

Переключиться на другой рабочий стол (например, на стол 2), щёлкнув курсором мыши на номере рабочего стола в панели задач или нажав `Ctrl Alt 2`.

Опять же любым способом запустить программу эмулятора терминала. Окно программы откроется на текущем (активном) рабочем столе, то есть уже на столе 2.

Продолжить работу с открытыми окнами терминалов, переключаясь по мере необходимости между столами 1 и 2.

### ***Работа с эмулятором терминала***

Эмулятор терминала предоставляет в системе XWindow интерфейс командной строки ОС Linux со всеми её возможностями. В системе учебного класса X-терминалов установлено два эмулятора — `xterm` и `rxvt`. Пользоваться можно любым из них, но на практике удобнее работать с эмулятором `rxvt`.

Запуск `rxvt` выполняется либо через главное меню (прямая ссылка на него есть в верхней части меню), либо через щелчок курсором мыши на иконке, находящейся на панели задач.

Любую другую программу (включая копию `rxvt`) можно запустить, набрав в командной строке окна терминала её имя. При

этом, если запускаемая программа рассчитана на работу в системе XWindow, откроется новое окно запущенной программы, а в окне терминала новое приглашение командной строки не появится, так как интерпретатор команд терминала будет ожидать завершения работы запущенной программы.

Эта особенность запуска программ имеет интересные следствия. Во-первых, запускаемая программа может выдавать какие-либо диагностические сообщения в обычные потоки стандартного вывода и стандартной диагностики (`stdout` и `stderr`). И этот вывод появится в окне терминала. Во-вторых, аварийно завершить работу программы можно, сделав активным окно терминала и нажав в нём клавиши `Ctrl` `C`.

Если необходимо, чтобы параллельно с работой запущенной программы в окне терминала можно было продолжать ввод других команд, программа должна запускаться в фоновом режиме, то есть на конце команды должен быть набран символ амперсанда (`&`).

В остальном работа в командной строке эмулятора терминала протекает так же, как в обычной терминальной командной строке.

### *Запуск терминального менеджера файлов*

Для системы X-терминалов, где основной объём работы выполняется в командной строке окна терминала с клавиатуры, наиболее удобным менеджером для управления файлами является аналог знаменитого Norton Commander, называемый Midnight Commander. Для его запуска в командной строке достаточно набрать команду

`mc`

Работа с `mc` близка к работе в Norton Commander с «поправкой» на особенности ОС Linux. Так, например, в `mc` невозможно переключаться между логическими дисками, так как все файловые системы в ОС Linux монтируются в одно дерево (разных логических дисков просто нет). Основные же «кнопочные» команды у этих менеджеров совпадают.

Особенностью `mc` является то, что запускаемые из-под него программы стартуют не в этом же, а в параллельно запускаемом окружении (образуют параллельный процесс).

При сокрытии панелей (команда `Ctrl` `O`) на экране также появляется приглашение командной строки параллельного окружения, а не собственного окружения `mc`.

Для исключения случайного размножения процессов `mc` не позволяет запускать команды «из-под панелей», если в параллельном

окружении выполняются какие-либо работы. Процесс слежения использует не очень совершенный алгоритм и это периодически приводит к курьёзам, когда параллельный процесс свободен, а запуск команды «из-под панелей» всё равно заблокирован.

Блокировка выражается в том, что на экране появляется окно с сообщением «Интерпретатор уже озабочен выполнением команды». Лучший способ разблокировать `mc` заключается в имитации снятия параллельной программы с выполнения. Для этого после закрытия окна сообщения необходимо набрать `Ctrl O`, `Ctrl C`, `Ctrl O`. Сразу после этого необходимо обратить внимание на текущий каталог. Он может измениться!

Если это не помогло и сообщение продолжает появляться, значит ситуация серьёзнее. Лучший способ в этом случае заключается в полном перезапуске `mc` (выйти из него в командную строку, нажав клавишу `F10` и подтвердив выход во всплывающих окнах, и снова дать в командной строке команду `mc`).

**ВНИМАНИЕ!** При необходимости задания переменных окружения оболочки необходимо выйти из `mc`. Выданные из-под `mc` (и при открытых, и при скрытых панелях) команды задания значений переменных окружения игнорируются.

### *Быстрый повторный ввод команд*

При работе в командной строке терминала без запущенного менеджера файлов `mc` для вызова в командную строку предыдущей команды для последующего выполнения или редактирования необходимо нажать клавиши `↑`. Несколько нажатий на эту клавишу позволяет «прокрутить» через командную строку несколько предыдущих команд. Для обратной «прокрутки» можно воспользоваться клавишей `↓`.

Редактировать командную строку можно при помощи клавиш `←`, `→`, `Backspace` и других.

Несколько десятков последних вводимых пользователем команд (в том числе в предыдущем сеансе работы) запоминаются системой в так называемой истории команд (`history`). Для просмотра списка этих команд используется команда `history`. Список команд выдаётся на терминал в виде таблицы, в которой каждая команда помечена своим порядковым номером. Для быстрого повтора (без редактирования) какой-либо команды можно просто ввести в командную строку комбинацию

**!**номер

где **номер** — порядковый номер команды в списке.

Для более удобного просмотра всего списка команд можно рекомендовать следующие конвейеры:

```
history | more
```

или

```
history | less
```

А для выдачи на терминал только команд определённого типа

```
history | grep шаблон
```

где **шаблон** — регулярное выражение, описывающее условие поиска.

Для быстрого получения в командной строке одной из ранее введённых команд для повторного запуска или редактирования можно также воспользоваться механизмом полуавтоматического поиска по истории команд. Для этого нажать клавиши **Ctrl** **R** и начать вводить любую характерную часть команды, которую надо получить в командную строку. При этом в командной строке будут последовательно появляться команды из истории команд, удовлетворяющие уже введённой части команды.

При появлении нужной команды просто приступить к её редактированию стандартным способом, нажав, например, клавишу **←**, либо нажать **Enter** для немедленного её выполнения.

Если нужную команду найти не удаётся, отменить поиск, нажав клавиши **Ctrl** **C**.

При работе в файловом менеджере **mc** ведётся своя история команд (отдельно от истории команд командной строки терминала). Команды из этой истории команд можно «прокрутить» комбинациями клавиш **Alt** **P** и **Alt** **N**.

## Получение подсказки

При работе в командной строке системы возможно получение подробной подсказки по любой команде. Для этого используется команда **man**. Например, для получения подсказки по команде **ls** достаточно ввести в командной строке

```
man ls
```

Для получения подсказки по самой команде **man** можно ввести

```
man man
```

Иногда на выданной странице с подсказкой сообщается, что более полную информацию можно получить в системе **info**. Эта система является более поздней по времени разработки и предоставляет

информацию в структурированной гипертекстовой форме. В этом случае получение подсказки выполняется командой

```
info команда
```

где **команда** — команда, по которой запрашивается подсказка.

Естественно, возможно получение подсказки по работе с самой системой **info**

```
info info
```

Выдаваемая командами **man** и **info** информация является информацией из первоисточника. Она в обязательном порядке (по крайней мере **man**) присутствует в любой системе Linux. Как правило, эта информация гораздо объёмнее и полнее, чем приводимая в различных книгах и справочных пособиях для пользователей Linux (UNIX).

Более полная информация может содержаться лишь в справочных материалах, сопровождающих исходные тексты программ. Такая информация, как правило, адресована лишь узким специалистам, не снабжена средствами для быстрого доступа и далеко не всегда присутствует в системе.

К сожалению, почти все справочные страницы, вызываемые описанными командами, написаны на языке оригинала (английском).

### Англо-русский словарь

Для чтения документации на английском языке время от времени бывает необходимо пользоваться англо-русским словарём.

В учебной системе установлен интерактивный англо-русский словарь, имеющий достаточно большую словарную базу. Словарь рассчитан на использование в режиме текстового терминала. Словарь позволяет переводить только отдельные слова; он не предназначен для перевода фраз и предложений. При запросе перевода какого-либо слова на экран выдаётся полная словарная статья, соответствующая этому слову в словаре.

Для перевода какого-либо слова достаточно дать команду

```
dict слово
```

В результате на экран будет выдана словарная статья, посвящённая этому слову.

Если слова в словаре не оказалось, будет предложен список из похожих слов, имеющих в словаре. Иногда он бывает полезен, иногда нет.

Подчеркнём ещё раз, что возможен перевод только с английского языка на русский.

## Просмотр характеристик и состояния системы

Во время сеанса работы в системе пользователь может получить разнообразную информацию о состоянии системы. Не претендуя на полноту, укажем здесь несколько полезных команд. Многие из них имеют несколько режимов работы. Подробную информацию о них можно получить в справочнике `man`.

`dmesg` — печатает в `stdout` (на терминал) системные сообщения из буфера ядра. Среди этих сообщений наиболее интересными являются сообщения об обнаруженной аппаратной конфигурации вычислительной установки, включая все задействованные интерфейсы.

`top` — печатает на экране и периодически обновляет текущую информацию о загрузке ресурсов системы, а также список наиболее ресурсоёмких процессов. Для завершения программы необходимо нажать `Ctrl C`.

`who` — печатает список пользователей (точнее, их регистрационных имён), работающих в настоящий момент в системе.

`df` — печатает таблицу, отображающую заполненность смонтированных дисковых разделов системы.

`du` — вычисляет место на диске, занятое файлами текущего или указанного каталога. Эту команду настоятельно рекомендуется периодически выполнять в своём домашнем каталоге каждому пользователю, чтобы следить за суммарным объёмом файлов в домашнем каталоге и своевременно принимать меры к удалению лишних файлов.

## Печать

В системе X-терминалов любой пользователь может распечатать необходимые ему материалы на принтере, входящем в состав системы.

Принтер используется в режиме коллективного доступа. Любой пользователь может отправить свои данные на печать, и эти данные будут напечатаны в порядке общей очереди.

Пользователь может в любой момент отменить печать своего задания, стоящего в очереди на печать. Пользователь не может влиять на приоритет своего задания печати и на продвижение заданий других пользователей в очереди печати.

В системе установлен матричный 9-игольчатый принтер, рассчитанный на чёрно-белую печать на широкой бумаге (чуть больше формата А3). Технические характеристики принтера позволяют выводить как текст, так и графические изображения. Однако из-за общего весьма низкого качества вывода графических изображений на

принтеры подобных конструкций, а также сильный расход при этом красящей ленты вывод последних рекомендуется ограничить лишь необходимыми в целях защиты учебных работ объёмами.

В подсистеме печати имеется две очереди заданий к одному принтеру. Очередь «по умолчанию» использует для печати мощный набор программных фильтров, позволяющих (теоретически) не задумываться о типе печатаемого файла (простой текст, текст программы, бинарная картинка, текст HTML и т. д.). Достаточно просто отправить файл в эту очередь и он будет автоматически проведён через фильтр, который обеспечит его правильную печать.\*

Для большинства случаев печати рекомендуется использовать именно эту очередь.

К сожалению, используемый алгоритм определения типа файла не может «предугадать» намерений печатающего. Кроме того, для некоторых типов файлов в качестве фильтров используются громоздкие программы, взаимодействие с которыми плохо проработано, либо эти программы сознательно удалены из системы при инсталляции. Поэтому если при печати файла возникли какие-либо проблемы (обычно речь идёт об ошибках при печати текстового файла, имеющего специальную разметку), необходимо попробовать выполнить печать через вторую очередь «gaw».

Необходимо также иметь в виду, что по техническим соображениям подсистема печати учебного класса настроена таким образом, что в очереди «по умолчанию» вывод информации выполняется лишь на листы формата А4.

Далее рассматриваются основные команды управления подсистемой печати.

---

\* Для интересующихся подробностями. Система фильтров построена таким образом, что при постановке задания на печать в очередь файл обрабатывается определённым набором фильтров с целью получить его эквивалент в формате Postscript (PS). И в очередь ставится только документ в формате PS. При непосредственном выполнении печати этот документ может быть просто направлен на устройство печати, если оно само поддерживает язык PS. Если же (как в нашем случае) устройство печати управляется другим языком (в случае матричного принтера это язык ESC-последовательностей), то при непосредственном выполнении печати документ из очереди прогоняется через фильтр Ghostscript, который программно интерпретирует язык PS и преобразует входной поток в набор команд, понятных заданному устройству вывода.

## *Быстрая отмена печати*

Если одно или несколько заданий на печать были посланы ошибочно, быстро удалить из очереди все задания данного пользователя можно командой

```
lprm -
```

для очереди «по умолчанию» и командой

```
lprm -P raw -
```

для очереди «raw».

Команда удалит из очереди только задания пользователя, запустившего её. Никакие задания других пользователей удалены не будут.

В связи с тем, что принтер имеет собственный буфер печатаемых данных, уже начавшаяся ошибочная печать может не остановиться. В этом случае после выполнения описанной выше команды для остановки печати необходимо обратиться к дежурному инженеру.

## *Рекомендуемые команды печати*

Команды рассчитаны только на печать через очередь «по умолчанию». Вся печать при этом выполняется в графическом режиме принтера (поэтому относительно долго, зато с поддержкой шрифтов разного начертания и кегля). Дополнительно каждый печатаемый лист снабжается информацией о пользователе, имени файла, дате и времени печати. Также выполняется автоматическая нумерация страниц.

Для печати текстового файла в общем случае необходимо дать команду

```
a2ps файл
```

Если команда завершается успешно, то на терминал будет выдана информация о количестве страниц, которые в итоге будут напечатаны.

Если же на терминале появляется сообщение об ошибке обработки файла, это вероятнее всего означает, что система печати попыталась выбрать фильтр, который в учебной системе с ней не согласован. В этом случае можно порекомендовать команду

```
cat файл | a2ps
```

Описанными выше командами можно печатать текстовые файлы большинства встречающихся на практике типов: чистые тексты, тексты программ и т. п., а также текстовые производные бинарных файлов, полученные командами типа `od`.



Печать бинарных файлов картинок выполняется через конвертирование картинок любого формата в формат Postscript. Следующая пара команд обеспечит печать на принтере картинки, хранящейся в файле `picture.jpg`

```
convert picture.jpg picture.ps
```

```
lpr picture.ps
```

Желательно не забыть после печати удалить промежуточный файл `picture.ps`.

### *Печать чистого текста*

Иногда при печати текстового (реже бинарного) файла необходимо гарантированно избежать системы фильтров, применяемой в очереди «по умолчанию». Специально для этой цели в системе предусмотрена очередь «raw» (буквальный перевод «сырой, необработанный»). Файлы, попадающие в эту очередь, передаются устройству печати безо всяких преобразований байт за байтом.

При печати текстового файла необходимо иметь в виду следующее:

а) печать текста будет выполняться символами, начертание которых определяется встроенным знакогенератором принтера;

б) в знакогенераторе принтера есть символы кириллицы (русские буквы), но по историческим причинам они расположены в альтернативной кодировке (другие названия: «кодировка MS-DOS», «CP866»);

в) кодировка русских текстов в системе X-терминалов — `koi8-r`, то есть для печати на принтере в этом случае необходима перекодировка текста;

г) управление сменой шрифтов возможно только специально вставленными в текст ESC-последовательностями (другие программы их не понимают) и только в пределах возможностей знакогенератора принтера;

д) возможна печать на всю ширину принтера, однако переносы слишком длинных строк сам принтер выполнять не будет — задача определения максимальной длины строки и переноса «лишнего» текста на новую строку должна решаться самим пользователем.

К счастью, в составе системы имеется набор готовых фильтров, обеспечивающих полуавтоматическое решение этих задач. Приведём пример команды печати текстового файла на широкую бумагу в две колонки (команда может быть набрана в одну строку или разорвана при наборе как показано ниже на любом символе | ):

```
cat файл | fold -w 65 -s | pr -2 -W 132 |  
iconv -f koi8-r -t cp866 | lpr -P raw
```

В данной команде `cat` подаёт файл на вход конвейера. Фильтр `fold` формирует строки не длинее 65 символов, перенося их длинные «хвосты» на следующие строки, причём в качестве точек разбиения выбираются пробелы. Фильтр `pr` превращает получившийся поток в двухколоночный, исходя из общей ширины зоны печати 132 символа. Фильтр `iconv` изменяет кодировку русских букв потока. Наконец `lpr` преобразует получившийся поток в задание на печать и ставит его в очередь «raw».

Более подробную информацию по каждому из использованных в команде фильтров можно как всегда получить через систему помощи `man`.

### *Просмотр заданий в очереди печати*

Для очереди «по умолчанию» просмотр очереди печати выполняется командой

```
lprq
```

Для очереди «raw» команда немного длинее:

```
lprq -P raw
```

Результаты работы обеих команд выдаются на терминал.

### *Удаление задания из очереди печати*

При необходимости удалить конкретное задание необходимо первоначально просмотреть очередь печати и идентифицировать в этой очереди удаляемое задание (найти его и определить его номер в очереди).

Для удаления задания из очереди «по умолчанию» используется команда

```
lprm номер
```

Для удаления задания из очереди «raw» используется команда

```
lprm -P raw номер
```

В обоих случаях пользователь может удалить только задания, поставленные в очередь им самим. При необходимости очистить очередь от всех заданий (забытых другими пользователями) необходимо обратиться к дежурному инженеру.

## Работа с дискетами

В системе предусмотрена возможность использования дискет 3,5” в качестве носителей данных. С их помощью пользователь может ввести в систему какие-либо файлы данных, подготовленные «на стороне», либо выполнить резервирование своих важных файлов, подготовленных в системе.

Порядок работы с дискетами в ОС Linux серьёзно отличается от порядка работы с другими носителями данных. Причины этого заключаются в следующем.

Как известно, в отличие от таких ОС как MS DOS, MS Windows и некоторых других, в ОС UNIX вообще и в ОС Linux в частности на уровне файловой системы нет понятия диска, обозначаемого буквой (или каким-либо другим способом). Вместо этого файловые системы всех физических носителей данных должны быть смонтированы в единое дерево каталогов. Файлы, соответствующие какому-либо носителю информации, в файловой системе ОС Linux будут представляться как файлы, расположенные в каком-либо каталоге единой файловой системы.

В принципе, точно так же можно получить доступ и к файлам на дискете, смонтировав её файловую систему в какой-либо специально выделенный для этого каталог в файловой системе сервера. Проблема здесь заключается в том, что при этом пользователи, выполнив необходимые им операции с файлами, забывают отмонтировать файловую систему перед извлечением дискеты из дисковода. Никакого предохранителя от таких действий в дисководах гибких дисков IBM PC не предусмотрено. В результате пользователи рискуют испортить себе файловые системы на дискетах и, что ещё хуже, привести всю многопользовательскую систему в состояние постоянных попыток синхронизации с файловой системой, которой уже нет.

Возможен также вариант использования системы автоматического монтирования, при котором дискета монтируется в дисководе автоматически при обращении к заданному подкаталогу, соответствующему её файловой системе, а после этого, если к дискете нет обращений в течение некоторого короткого времени, выполняется автоматическое отмонтирование. Но у этого варианта та же проблема. Нетерпеливый пользователь может вытащить дискету из дисковода до того момента, как она будет отмонтирована автоматикой (никакого специального сигнала при этом не формируется).

Поэтому в рассматриваемой системе для доступа к файлам на дискетах используется третий вариант, опирающийся на пакет программ `mtools`. Этот пакет предоставляет набор команд, аналогичный командам для работы с файлами ОС MS DOS (не путать с командами оболочки Norton Commander!). Отличие заключается в том, что все команды этого пакета начинаются с буквы `m` (видимо, «на память» об ОС MS DOS) и немного отличаются ключи управления режимами работы команд. Так команда просмотра содержимого каталога в ОС MS DOS называется `dir`, а в пакете `mtools` — `mdir`; команда копирования в ОС MS DOS называется `copy`, а в пакете `mtools` — `mcopy` и т. д.

Удобство пакета заключается в том, что при обращении к дискете используется знакомая аббревиатура буквенного имени дисководов (в зависимости от дисковода `a:` или `b:`). Неудобство — в том, что с файлами на дискетах невозможно работать ни из каких файловых менеджеров (и, естественно, ни из каких редакторов и других программ). Имена всех копируемых файлов необходимо указывать в командной строке. Однако для удобства всё же допускается задание групп файлов с помощью стандартных метасимволов.

### *Просмотр содержимого дискеты*

Операция выполняется командой  
`mdir диск`

где `диск` — буква с двоеточием, соответствующая дисководу, — `a:` или `b:`.

Команда имеет несколько ключей, которые однако на практике практически не используются. Поэтому интересующихся отправим к справочному руководству `man`.

Результат работы команды выводится в стандартный поток вывода (на экран терминала). Особенностью вывода является то, что в первом столбце выводится имя файла в форме, в которой оно отображается в стандарте ОС MS DOS (8 символов имя файла и 3 символа расширение). А полное (длинное) имя файла, соответствующее его форме в ОС Linux, выводится в последнем столбце. Именно на это, последнее имя надо ориентироваться, указывая имя файла в командной строке остальных команд `mtools`.

При необходимости просмотреть содержимое подкаталога на дискете команда должна выглядеть так

`mdir диск/подкаталог`

где **подкаталог** — имя подкаталога на диске. Обратите внимание, что подкаталоги должны разделяться символом «прямой слеш» (/), как это принято в ОС UNIX.

Например, если на дискете, установленной в первый дисковод (соответствует диску **A:** в ОС MS DOS), существует подкаталог **test**, содержимое которого надо просмотреть, команда будет выглядеть так

```
mdir a:/test
```

### *Копирование файлов с дискеты*

Прежде всего пользователи должны иметь в виду, что имена файлов на дискете должны быть составлены только из латинских букв и цифр. Имена файлов, содержащие русские буквы, в пакете **mtools** будут представлены абракадаброй, задать которую в командной строке практически невозможно.

Для копирования обычных файлов с дискеты в дисководе **a:** на диск системы используется команда

```
mscopy a:файл .
```

Точка на конце команды означает копирование в текущий каталог. Файл может быть записан как шаблон, содержащий стандартные метасимволы (\*, ?), или как единичный файл.

При необходимости копирования всех файлов, содержащихся на дискете **a:** в отдельном каталоге, используется команда

```
mscopy -/ a:каталог .
```

или для более новых версий **mtools**

```
mscopy -s a:каталог .
```

В данном случае должен быть указан конкретный каталог, имеющийся на дискете. Копирование файлов (вместе с каталогом) выполняется в текущий каталог на диске системы.

Если на дискете всё-таки оказались имена файлов или, что ещё хуже, каталогов с именами, содержащими русские буквы, предлагается следующая последовательность действий.

а) Скопировать все файлы и каталоги дискеты на диск системы командой

```
mscopy -s a:* .
```

Здесь ключ **-s** означает копирование со всеми подкаталогами, а метасимвол **\*** раскрывается во все файлы в корневом каталоге дискеты.

б) С помощью файлового менеджера (например **mc**) на диске системы переименовать нужные файлы и каталоги, а ненужные удалить.

## *Копирование файлов на дискету*

Операция полностью аналогична предыдущей. Копирование одного файла на дискету в дисководе **a:** выполняется командой

**mscopy файл a:**

Здесь **файл** — полное или сокращённое имя файла в файловой системе диска сервера. В простейшем случае (если не указаны элементы пути) это имя файла в текущем каталоге.

В командной строке можно через пробел указать несколько имён файлов или использовать шаблон, описывающий группу файлов.

Возможно копирование и группы файлов, содержащихся в каталоге. В этом случае применяется уже упоминавшийся ключ **-s**.

Однако с точки зрения производительности процесса (один большой файл копируется значительно быстрее, чем несколько маленьких), и экономии места на дискете рекомендуется группу файлов перед копированием упаковать в архив и копировать на дискету один файл архива вместо группы файлов.

Если файлы копируются просто для резервного хранения рекомендуется использовать стандартный архиватор системы UNIX **tar** и утилиту сжатия **gzip**. В результате будет получен файл архива с расширением **.tar.gz** или **.tgz**.

Если же предполагается использовать копируемые файлы в других ОС (например MS Windows), то необходимо использовать архиватор **zip**. В результате будет получен файл с расширением **.zip**. Этот архиватор удобен тем, что имеется сегодня практически в любой реализации ОС Linux и создаёт файлы архивов в формате, совместимом с широко распространённым архиватором **pkzip** версии 2.04g, а также с большинством версий архиватора WinZip.

## *Очистка дискеты*

Обычно не имеет большого смысла создавать на дискете разветвлённую систему каталогов. Дискета имеет небольшую ёмкость, и, как правило, все файлы записываются в корневой каталог. Поэтому рассмотрим удаление конкретных файлов только из корневого каталога. Для случая, когда на дискете имеются вложенные каталоги с файлами, рассмотрим только команду полной очистки дискеты.

Удаление с дискеты, находящейся в дисководе **a:**, файла с заданным именем выполняется командой

**mdel a:файл**

Здесь **файл** — имя файла, который необходимо удалить.

Если на дискете есть только файлы в корневом каталоге, то полная очистка дискеты выполняется командой

```
mdel a:*
```

Если же на дискете есть вложенные каталоги, то полная очистка дискеты (без форматирования) выполняется командой

```
mdeltree a:
```

Команда может выдать сообщение, что невозможно удалить корневой каталог (/). Это сообщение не влияет на правильность выполнения команды и может быть проигнорировано.

### *Согласование кодировок русского текста (перекодирование)*

В учебной системе X-терминалов для представления русских символов применяется традиционная для ОС Linux в России кодировка koï8-r. Применение кодировки Unicode, получающей в последнее время всё большее распространение, на данном этапе существования системы было признано нецелесообразным.

При переносе в систему файлов из других ОС, в частности из ОС MS Windows, возникает проблема несовместимости кодировок русского языка. Эта проблема относится только к текстовым файлам (бинарные файлы переносятся без изменений) и является на самом деле более общей, чем представляется многим первоначально. На самом деле отличаются не только кодировки, но и способы представления концов строк в текстовых файлах. Так в файлах ОС MS Windows концом строки считается комбинация символов `\r\n`, в файлах ОС UNIX (Linux) — символ `\n`, а в файлах ОС MacOS (Apple Macintosh) — символ `\r`. Эти отличия возникли исторически. Какой из способов задания конца строки является «идеологически более правильным» обсуждать сегодня бессмысленно.

Надо заметить, что на сегодняшний день для многих программ ОС Linux, работающих с текстовыми файлами, (но не для всех!) не имеет большого значения оканчивается ли строка на `\n` или на `\r\n`. Складывается впечатление, что разработчики прикладных программ ОС Linux (но не других ОС UNIX) учли, что время от времени в эту ОС вносятся «чужеродные» файлы из ОС MS Windows, которые надо обрабатывать стандартными программами. Из-за этого в некоторых случаях бывает довольно трудно определить какие именно преобразования необходимо сделать с файлом, чтобы привести его к структуре, соответствующей стандартам ОС UNIX.

С другой стороны, для исключения глупых и необъяснимых ошибок при работе с такими файлами приведение их к стандартной структуре ОС UNIX крайне желательно.

Такое приведение структуры файла к стандарту ОС, включая принятую в системе кодировку национального (локального) языка (в нашем случае русского), называют перекодированием.

Ещё раз подчеркнём, что перекодировке подлежат только текстовые файлы, то есть файлы, содержащие чистый текст и тексты программ на исходных языках программирования. Обычно такие файлы имеют одно из следующих расширений: txt, c, cpp, cc, asm, HTML, PHP. Но могут иметь и другие расширения или вообще их не иметь.

Бинарные файлы, не подлежащие перекодировке ни при каких условиях, обычно являются файлами архивов, картинок, музыки, видео, исполняемых программ и модулей в объектном коде, а также файлами, формируемыми сложными коммерческими программными комплексами типа MS Office, AutoCAD и т. п. Такие файлы обычно имеют одно из следующих расширений: zip, arj, rar, tar, gz, tgz, jpg, gif, exe, o, a, so, doc, xls, dwg и многие другие.

В некоторых редких случаях могут возникать «накладки». Так по старой традиции, сложившейся в мире ОС MS DOS в те времена, когда пакета MS Office ещё не существовало, текстовые файлы с документацией часто получали расширение doc. Эти файлы при переносе в ОС Linux нуждаются в перекодировке в отличие от файлов, созданных в пакете MS Office с тем же расширением, которые в перекодировке не нуждаются (но и просмотрены стандартными средствами ОС Linux быть не могут).

Необходимо также учитывать, что хотя бинарные файлы архивов, созданные в других ОС, перекодированию не подлежат, после их распаковки могут создаваться текстовые файлы, которые должны быть перекодированы.

Как показывает практика, для успешного выполнения операции перекодирования в исходном текстовом файле не должно быть «лишних», нетекстовых символов. К сожалению, практически все текстовые (!) редакторы ОС MS Windows (включая Блокнот/Notepad) время от времени вставляют в файл бинарные коды, о назначении которых можно только догадываться. Эти бинарные коды при редактировании файлов в ОС MS Windows никак не отображаются, в файле могут появляться где угодно и принимают разные значения,



но при этом не позволяют (с точки зрения программ ОС Linux) рассматривать такие файлы как чисто текстовые.

Единственно верного способа борьбы с такими «неправильными» символами не существует, так как их появление в файле не соответствует никакой логике. Просто пользователю необходимо быть готовым к тому, что программа перекодирования после преобразования части файла может выдать сообщение о неправильном символе или структуре файла. В этом случае надо попытаться выполнить преобразование альтернативной программой перекодирования, а если это не поможет, то найти в файле ошибочный код и удалить его с помощью любого обычного текстового редактора — в большинстве случаев это будет правильным решением.

Для выполнения перекодирования в системе применяются две программы — `iconv` и `recode`. Каждая из этих программ обеспечивает перекодирование между довольно большим количеством кодировок (не только русских). Программа `recode` кроме того обеспечивает замену вариантов представления концов строк.

При перекодировании файлов из ОС MS Windows в ОС UNIX (Linux) необходимо знать кодировку исходного файла. Дело в том, что в зависимости от того, в каком окружении готовился файл, его кодировка может быть либо `cp866` (Code Page 866, называемая также «альтернативной» или просто «кодировкой DOS»), либо `cp1251` («родная» кодировка русских текстов ОС MS Windows). Если исходная кодировка точно неизвестна, необходимо провести предварительные эксперименты с перекодированием из обеих кодировок и оценить получившийся результат.

Изредка встречается также третий вид кодировки — Unicode, часто называемая также UTF-8. (Строго говоря, Unicode и UTF-8 — разные кодировки. Но применительно к кириллице они настолько похожи, что на практике их можно считать совпадающими.) Этот вид кодировки является 16-битным. В этой кодировке могут быть без каких-либо проблем представлены многоязычные тексты любой степени сложности. Но размеры файлов с национальными (в том числе и русскими) текстами получаются вдвое больше. Кроме того, отработанность стандартного ПО ОС Linux для работы с такими файлами всё ещё недостаточна. Но то, что русский текст в файле записан в кодировке Unicode, можно установить почти однозначно по одному его внешнему виду.

Перекодирование из кодировки `cp866` в `koï8-r` может быть выполнено одной из команд

```
iconv -f 866 -t KOI8-R infile >outfile
```

или

```
recode 866..KOI8-R <infile >outfile
```

Здесь файл `infile` — исходный файл, а файл `outfile` — файл результата.

Внимание! Обратите внимание на символы перенаправления потоков (`<` и `>`), а также на то, что файлы `infile` и `outfile` должны быть обязательно разными! При необходимости получить файл с исходным именем файл `outfile` надо потом просто переименовать в файл `infile` стандартной командой `mv`

```
mv outfile infile
```

Перекодирование с одновременным изменением способа представления концов строк выполняется командой

```
recode 866/CR-LF..KOI8-R <infile >outfile
```

Перекодирование из кодировки `cp1251` в кодировку `koi8-r` выполняется аналогичными командами

```
iconv -f 1251 -t KOI8-R infile >outfile
```

```
recode 1251..KOI8-R <infile >outfile
```

```
recode 1251/CR-LF..KOI8-R <infile >outfile
```

Перекодирование из кодировки `UTF-8` в кодировку `koi8-r` выполняется командами

```
iconv -f UTF-8 -t KOI8-R infile >outfile
```

```
recode UTF-8..KOI8-R <infile >outfile
```

```
recode UTF-8/CR-LF..KOI8-R <infile >outfile
```

Возможна и обратная перекодировка файлов в кодировку ОС MS DOS либо ОС MS Windows. За подробными инструкциями отошлём читателей к справочному руководству `man`.

## Работа с флэш-дисками

В учебной системе X-терминалов предусмотрена возможность работы с различными USB-устройствами пользователей, которые могут рассматриваться как дисковые накопители. Далее все такие устройства именуются флэш-дисками.

К сожалению, периодически на рынке появляются «новейшие» устройства такого типа, отрицающие некоторые уже сложившиеся стандарты. А регулярное обновление ОС сервера учебного класса невозможно. Поэтому какие-то из устройств могут оказаться неработоспособными в системе учебного класса.

Гарантируется работа с флэш-дисками ёмкостью до 512 Мбайт, а также с флэш-картами, подключаемыми через стандартные USB считыватели.

Для работы с любым флэш-диском пользователю необходимо обратиться к дежурному инженеру. Инженер выполнит подключение диска к системе и монтирование его файловой системы с указанием пользователю точки монтирования (каталога в котором окажется файловая система флэш-диска). По окончании работы с флэш-диском пользователь должен сообщить об этом дежурному инженеру, который выполнит размонтирование файловой системы и отключение флэш-диска от системы.

Пользователи должны стремиться к минимизации времени нахождения своих флэш-дисков в составе системы. Необходимо стремиться к тому, чтобы флэш-диски подключались к системе только на время, необходимое для копирования файлов. Работать на подключённых к системе флэш-дисках не разрешается!

### **Работа с архивными файлами**

Архивные файлы могут использоваться пользователями для различных целей. Наиболее частая цель применения архивных файлов в учебной системе — копирование рабочих файлов на дискету и с дискеты. Во-первых, как уже было сказано, один большой файл копируется значительно быстрее, чем группа маленьких. Во-вторых, упрощается команда копирования. В-третьих, даже при копировании одного большого файла архивирование (вернее, сжатие) позволяет уменьшить его объём, что в свою очередь ускоряет процесс копирования, а иногда и просто позволяет уместить большой файл на дискете.

Надо заметить, что процесс, который в настоящее время среди программистов и пользователей принято называть архивированием, на самом деле состоит из двух этапов, которые могут для определённых целей применяться и порознь.

Первый из них — это собственно архивирование, — создание файла архива из нескольких (или одного) рабочих файлов и каталогов.

Второй этап — выполнение сжатия получившегося файла архива для уменьшения занимаемого им на диске места.

На практике для хранения файлов могут применяться как несжатые файлы архивов, так и отдельные сжатые файлы, не имеющие формата архивов.

Представление об архивировании файлов как о процессе, при котором одновременно происходит формирование единого файла архива со сжатием входящих в него файлов, появилось относительно недавно и укоренилось в умах пользователей и программистов, благодаря широкому распространению ОС MS DOS с её «рассчитанными на пользователя» программами архивирования.

В ОС UNIX и, в частности, в учебной системе на базе ОС Linux применяются следующие виды архивных файлов: `.tar`, `.tar.gz`, `.tgz`, `.cpio`, `.zip`. В некоторых версиях ОС Linux возможно использование архивов и некоторых других форматов.

Форматы `.tar` и `.cpio` стандартизованы POSIX как форматы файлов для обмена данными между вычислительными системами, в том числе основанными на разных платформах.

### *Файлы формата tar*

`tar` — стандартный формат архива в мире ОС UNIX. Представляет собой копию группы рабочих файлов как бы записанных на магнитную ленту. Создаётся, распаковывается и управляется командой `tar` (Tape ARchiver — буквально «архиватор на ленту»). Все файлы в архиве несжаты.

Создание архива выполняется командой

```
tar cf имя.tar список_файлов
```

Здесь `имя.tar` — имя создаваемого файла архива. Расширение `.tar` само по себе не появляется. Его надо указывать в командной строке вручную. `список_файлов` — это обычный список файлов и каталогов через пробел. Каждый указанный каталог будет упакован в архив вместе со всеми содержащимися в нём файлами и подкаталогами. Допускается указание обычных шаблонов командной строки ОС UNIX. Так команда

```
tar cf file.tar *
```

выполнит создание архивного файла `file.tar` и упакует в него все файлы и каталоги (вместе со всеми подкаталогами и их файлами) текущего каталога.

Распаковка файлов и каталогов из архива выполняется командой

```
tar xf имя.tar
```

Здесь `имя.tar` — имя разворачиваемого файла архива.

Архивный файл всегда распаковывается в текущий каталог, если только в нём не указаны файлы с абсолютными путями.\*

---

\* Такие архивные файлы создаются и управляются особыми вариантами команды `tar`, которые здесь не рассматриваются.

## *Файлы формата gz и tar.gz*

`tar.gz` — просто сжатый файл предыдущего формата. Сжатие выполняется командой `gzip` (это не то же самое, что формат `zip` архиватора `pkzip!`), восстановление — командой `gunzip`.

`gz` — просто файл любого типа, сжатый командой `gzip`.

Команда `gzip` обеспечивает достаточно эффективное сжатие файлов большого размера. Именно поэтому она широко применяется для сжатия файлов архивов типа `tar`.

Сжатие файла (любого, в том числе файла архива) выполняется командой

```
gzip файл
```

Здесь `файл` — имя сжимаемого файла с расширением.

В результате выполнения команды первоначальный файл из файловой системы удаляется, а вместо него появляется `файл.gz`, причём расширение `.gz` всегда приписывается к имени файла справа независимо от его первоначального расширения (которое сохраняется). Иными словами, при сжатии файла архива с именем `file.tar` в результате выполнения команды

```
gzip file.tar
```

получим файл с именем `file.tar.gz`.

Восстановление первоначального (несжатого) файла выполняется командой

```
gunzip файл.gz
```

В результате выполнения команды сжатый файл из файловой системы удаляется, а на его месте появляется первоначальный файл (с именем без расширения `.gz`).

## *Файлы формата tgz*

`tgz` — архивный файл, созданный аналогично `tar`, но с неявным вызовом `gzip` в процессе создания. Команда создания архива выглядит так:

```
tar czf имя.tgz список_файлов
```

Здесь `имя.tgz` — имя создаваемого файла архива. Расширение `.tgz` само по себе не появляется. Его надо указывать в командной строке вручную. `список_файлов` — это обычный список файлов и каталогов через пробел. Каждый указанный каталог будет упакован в архив вместе со всеми содержащимися в нём файлами и подкаталогами и дополнительно сжат командой `gzip`. Допускается указание обычных шаблонов командной строки ОС UNIX.

Распаковка архива вместе с восстановлением файлов выполняется командой

```
tar xzf имя.tgz
```

Команда работает аналогично команде `tar xf` (см. выше).

### *Файлы формата `cpio`*

`cpio` — альтернативный к `.tar` формат архива. Создаётся, распаковывается и управляется командой `cpio`. Все файлы в архиве несжаты.

Архивы данного формата распространены значительно меньше, чем архивы форматов `tar` и `tgz` (хотя имеют перед ними определённые преимущества). В настоящее время они применяются в основном во внутрисистемных целях. Поэтому их создание и распаковка здесь не рассматриваются. При крайней необходимости всю информацию по обслуживанию этого формата архивов можно, как всегда, получить из справочной системы `man`.

### *Файлы формата `zip`*

`zip` — «классический» формат архивного файла, пришедший из ОС MS DOS, то есть появившийся значительно позже описанных выше форматов. На сегодняшний день создаётся, распаковывается и управляется большим количеством программ во многих ОС. Так в ОС MS DOS это пара программ `pkzip/pkunzip`, в MS Windows — `WinZip`, а также `WinRAR` (необходимо установить флаг, что архивный файл должен быть сделан в формате `zip`). В ОС Linux формат архива поддерживается парой программ `zip` (создание архива) и `unzip` (распаковка файлов из архива).

Создание архива выполняется командой

```
zip имя_архива список_файлов
```

Здесь `имя_архива` — имя файла архива без расширения, `список_файлов` — список файлов через пробел, которые необходимо поместить в архив.

Команда помещает в архив только явно указанные файлы. Естественно, допускается группы файлов указывать шаблонами. Если в списке файлов окажется каталог, то он будет помещён в архив БЕЗ вложенных файлов и подкаталогов. Все файлы и каталоги по умолчанию подвергаются сжатию.

Для включения в архив всего содержимого указанных каталогов применяется ключ `-r`. Например, команда

```
zip -r имя_архива *
```

поместит в архив все файлы и каталоги со всеми их файлами и подкаталогами, хранящиеся в текущем каталоге.

Распаковка файла архива выполняется командой

```
unzip имя_архива.zip
```

Здесь `имя_архива.zip` — имя файла архива с расширением.

Распаковка архива производится в текущий каталог. В отличие от команды `pkunzip` ОС MS DOS, здесь распаковка всегда выполняется с построением дерева подкаталогов, входящих в файл архива.

### *Разбиение большого файла на части*

Случается, что большой файл, даже обработанный программой сжатия, не уместается на одной дискете (или каком-то другом доступном носителе данных). В этом случае для записи на внешний носитель его можно разбить на части и использовать несколько экземпляров внешнего носителя.

В ОС UNIX (Linux) предусмотрена стандартная утилита разбиения файла на части. Команда

```
split -b 1440k file
```

разобьёт файл `file` на части, примерно равные по размеру ёмкости одной дискеты 3" 1,44 Мб (этот размер задан ключом `-b`). Части будут иметь имена `хаа`, `хаб` и т. д. При этом исходный файл сохраняется в файловой системе.

Теперь каждую часть можно копировать на отдельную дискету.

Обратите внимание, что команда `split` в отличие от многих подобных команд ОС MS DOS и Windows просто разрезает файл на части, не снабжая эти части никакой информацией об исходном файле. Исходное имя файла, включая его тип, пользователь должен помнить сам!

### *Сборка файла из частей*

Пусть на диске в текущем каталоге имеется последовательность файлов `хаа`, `хаб` и т. д., и известно, что это полная последовательность частей, на которые был разрезан большой файл. Пусть также известно (а это должно быть известно!), что большой файл имел имя `arch.tar.gz`. Тогда восстановление исходного файла выполняется командой

```
cat x* >arch.tar.gz
```

Данная команда предполагает, что в текущем каталоге нет файлов, не являющихся частями файла `arch.tar.gz`, имена которых начинаются с буквы `x`.

## Создание и редактирование текстовых файлов

Для ОС Linux существует большое количество как универсальных, так и специализированных текстовых редакторов, работающих как в режиме текстовой консоли, так и в графическом режиме XWindow. Последние часто имеют более или менее развитые возможности WYSIWYG.

Однако, поскольку в идеологии применения ОС UNIX подавляющее большинство работ сводится к созданию и преобразованию файлов, содержащих чистый текст, для практической повседневной работы важно освоить именно текстовый редактор консоли, полностью управляемый с клавиатуры без привлечения дополнительных устройств (мыши и др.).

Изучать работу с любым редактором рекомендуется на практике. При этом лучше всего первое занятие посвятить только изучению команд редактора, создавая и редактируя пробный файл, не содержащий важной информации.

Не рекомендуется разбрасываться и пытаться изучать сразу несколько текстовых редакторов. Возможности разных редакторов могут сильно отличаться, но рядовой пользователь крайне редко в своей работе использует возможности, специфичные для какого-либо конкретного редактора.

Далее будут рассмотрены два текстовых редактора, установленных в учебной системе — `mcedit` и `vi`. Оба редактора обладают функционально полным набором команд, позволяющим выполнять подавляющее большинство действий, необходимых при создании и редактировании файлов, включая работу с блоками текста. Оба редактора не позволяют выполнять групповое редактирование файлов в общепринятом смысле этого термина, но тем не менее имеют встроенные возможности, позволяющие переносить блоки текста между файлами.

Оба редактора не являются простейшими и обладают многими интересными возможностями, но большинство пользователей используют их именно как простые (без излишеств) редакторы текста.

Редактор `mcedit` чаще всего устанавливается вместе с файловым менеджером `mc` (Midnight Commander). В силу этого на конкретной вычислительной установке его может не быть. Но с точки зрения современного пользователя компьютера он довольно прост в освоении.

Редактор `vi` обычно присутствует на любой вычислительной установке с ОС UNIX (Linux, FreeBSD и т. п.). Но освоить его



несколько сложнее. Впрочем, несмотря на внешнюю «непрезентабельность» `vi` даже по сравнению с `mcedit`, возможности его по редактированию текстовых файлов на сегодняшний день превосходят возможности даже многих «продвинутых» редакторов с графическим интерфейсом.

Поэтому студентам специальности ВТ настоятельно рекомендуется освоить именно редактор `vi`. Студенты остальных специальностей могут остановить свой выбор на `mcedit`.

### *Редактор `mcedit`*

Если файл ещё не существует, то не имеет смысла набирать текст для безымянного файла. Поэтому перед вызовом редактора необходимо сразу определиться с его именем. Если же файл, который необходимо отредактировать, уже существует, то необходимо знать его точное имя.

Вызов редактора осуществляется командой  
`mcedit filename`

где `filename` — имя редактируемого файла.

Если пользователь работает в файловом менеджере `mc`, то для редактирования существующего файла достаточно привести курсор на имя файла и нажать клавишу `F4`.

Редактирование текста в `mcedit` выполняется традиционным способом, поэтому здесь опишем только неочевидные особенности системы команд редактора.

Сохранение отредактированного текста выполняется нажатием клавиши `F2`. При этом редактор запросит подтверждение операции сохранения (другие редакторы обычно этого не делают).

Если в файле ничего не было исправлено, то выйти из него можно нажатием клавиши `F10` или двойным нажатием клавиши `Esc`.

Для отмены многих управляющих операций (всплывающих окон) также необходимо нажать клавишу `Esc` дважды.

При работе с блоками начало и конец блока обозначаются нажатием клавиши `F3`. Выделение блока столбцами — `Shift F3`.

Копирование блока — `F5`. Перемещение блока — `F6`. Удаление блока — `F8`. Откат неверно выполненной операции — `Ctrl U`.

Копирование блоков текста между разными файлами выполняется хитрым способом через специальный внешний файл. Порядок действий следующий:

- загрузить на редактирование первый файл (из которого будет браться копируемый блок);
  - выделить блок, подлежащий копированию (клавишей **F3**);
  - нажать клавишу **F9**. При этом откроется верхнее меню, в котором стрелками выбрать пункт **Файл** и в выпавшем меню пункт **Копировать в файл**. Нажать **Enter**;
  - в появившемся окне будет предложено имя файла, в который будет записан блок. Имя файла можно изменить, но для копирования одного блока удобнее оставить его по умолчанию (тогда при обратной вставке блока будет автоматически искаться то же имя файла);
  - нажать клавишу **Enter** и тем самым закончить операцию сохранения блока в промежуточном файле;
  - выйти из редактирования текущего файла (клавиша **F10**). При этом редактор закрывается и управление передаётся обратно в командную строку;
  - загрузить на редактирование второй файл (в который должен быть вставлен блок);
  - установить курсор на точку вставки блока;
  - нажать клавишу **F9**. При этом откроется верхнее меню, в котором стрелками выбрать пункт **Файл** и в выпавшем меню пункт **Вставить файл**. Нажать **Enter**;
  - в появившемся окне будет предложено имя файла, из которого будет вставлен блок. Имя файла можно изменить, но если на этапе сохранения блока его не меняли, то и в данном случае менять его не надо;
  - нажать клавишу **Enter** и тем самым закончить операцию чтения блока из файла;
  - сохранить изменения в файле (клавиша **F2** с подтверждением);
  - выйти из редактора (клавиша **F10**).
- Иногда при попытке набрать или даже просмотреть русский текст ничего не получается. Это означает, что необходимо настроить редактор на работу с русскими буквами. Настройка выполняется в общем меню программы **ms**. Для этого необходимо, выйдя из редактора и находясь в **ms**:
- нажать клавишу **F9** — появится верхнее меню;
  - в меню выбрать **Настройки** и в выпавшем меню — **Биты символов . . .**;
  - отметить пункты **Полный 8-битный вывод** и **Полный 8-битный ввод**;

- нажать клавишу `Enter`;
- сохранить сделанные настройки на будущее, нажав клавишу `F9` и выбрав в меню **Настройки**, **Сохранить настройки**.

## *Редактор vi*

Этот редактор необычен своей системой команд и наличием нескольких режимов работы. При этом он обладает мощностью, в некоторых случаях превосходящей современные «продвинутые» экранные редакторы. Чаще всего при редактировании файла используется три его режима работы: командный, строковый и ввода текста (режимы названы условно, в литературе они могут иметь другие названия).

В современной ОС Linux обычно устанавливается не собственно `vi` (который не является свободно распространяемым), а его «свободный» аналог `vim`, обладающий многими дополнительными возможностями, в частности возможностью одновременного редактирования нескольких файлов в разных окнах на экране. Однако здесь рассмотрим только необходимый минимум команд, присущий обоим редакторам.

Запуск редактора выполняется командой

```
vi filename
```

где `filename` — имя редактируемого файла (даже если его ещё нет и он только создаётся).

После запуска редактор оказывается в **командном режиме**. В этом режиме возможно перемещение курсора по тексту и выполнение операций копирования и удаления элементов текста (символов, слов, строк, блоков), а также поиск по тексту. Команды (в отличие от многих других редакторов) отдаются символьными клавишами (и на экране не отображаются):

`h`, `j`, `k`, `l` — перемещение по тексту влево, вниз, вверх, вправо (в учебном классе X-терминалов и на многих других установках возможно также применение обычных клавиш со стрелками);

`x` — удалить символ, под которым стоит курсор;

`r` — заменить символ, под которым стоит курсор, на символ, введённый сразу после нажатия клавиши;

`d d` (нажатие клавиши `d` два раза подряд) — удалить текущую строку (строка запоминается во внутреннем буфере);

`P` — вставить удалённую строку после текущей строки;

**P** (**Shift P**) — вставить удалённую строку перед текущей строкой;

**Y Y** — скопировать текущую строку в буфер (для вставки строки на новое место применяются описанные выше команды вставки).

Любой команде может предшествовать коэффициент повторения, набираемый «втёмную» (на экране не отображается) как число. Например, для копирования в буфер не одной, а пяти строк необходимо последовательно нажать клавиши **5 Y Y**. Скопированные строки как единый блок могут быть вставлены в нужное место текста командами **P** или **P**.

Поиск в тексте задаётся командой-клавишей **/**. При этом в нижней строке экрана появляется соответствующий символ, отмечающий строку, в которую необходимо ввести шаблон поиска. Шаблон может представлять собой как простую подстроку, так и сложное регулярное выражение. После ввода шаблона необходимо нажать **Enter**. Будет показано первое найденное совпадение или сообщение, что поиск неудачен.

Повтор поиска по тому же выражению выполняется клавишей **n**. Обратный поиск — клавиша **N** (**Shift n**).

Для отмены неверно выполненной операции необходимо нажать **u**. Для отмены операции отмены — нажать **Ctrl r**.

**Строковый режим** работы является аналогом строкового редактора **ed** и может быть полезен для выполнения команд, требующих указания дополнительных аргументов. Это команды сохранения файла, смены редактируемого файла, поиска и замены элементов текста и многие другие. Переход в строковый режим осуществляется из командного нажатием клавиши-команды **:**. При этом в нижней строке появляется соответствующий символ, и далее ввод команды осуществляется в эту строку. По завершении ввода команды она запускается нажатием клавиши **Enter**. Отменить команду и вернуться в командный режим можно нажатием клавиши **Esc**.

Перечислим наиболее полезные команды:

**:q** — выход из редактора без сохранения файла. Если были изменения, редактор не позволит выйти;

**:q!** — выход без сохранения даже если были изменения;

**:w** — сохранить на диск текущий редактируемый файл;

**:w filename** — сохранить редактируемый файл на диск в файл с новым именем **filename**;

**:w!** — сохранить в файл, защищённый от записи (операция возможна только, если файл редактирует владелец);

**:n** — перейти на следующий файл. Операция возможна только если редактор запускался с указанием в командной строке нескольких файлов;

**:e#** — перейти к предыдущему файлу;

При переходах между файлами сохраняются данные, сохранённые в буфер копирования. Таким образом оказывается возможным копирование блоков между файлами.

Операция поиска и замены в простейшем случае выглядит так

**:s/шаблон/замена/**

словом «замена» будет заменено найденное слово «шаблон». Причём если в строке несколько слов «шаблон», то заменено будет только первое. При необходимости заменить в строке все слова «шаблон» команда выглядит так

**:s/шаблон/замена/g**

а при необходимости выполнить поиск и замену по всему файлу перед началом команды необходимо указать специальный признак

**:%s/шаблон/замена/g**

Обратите внимание, что шаблоном может быть как простая подстрока, так и сложное регулярное выражение.

Для ввода нового текста необходимо перейти в **режим ввода**. Существует несколько команд, после которых можно вводить текст:

**[i]** — ввод текста с текущего положения курсора; **[a]** — ввод текста после текущего положения курсора; **[A]** — ввод текста в конец текущей строки; **[o]** — вставить новую строку для ввода текста после текущей; **[O]** — вставить новую строку для ввода текста перед текущей; **[c]** — удалить текст от текущего положения курсора до конца строки и начать режим ввода;

В режиме ввода можно только вводить текст. Для выполнения других команд редактора необходимо выйти из режима ввода в командный режим. Для этого необходимо нажать клавишу **[Esc]**.

Нажатие клавиши **[Esc]** безопасно в том смысле, что даже если редактор уже находился в командном режиме, этой командой ничего нельзя испортить. Поэтому если что-то не получается, для начала нужно пару раз нажать клавишу **[Esc]**, чтобы гарантированно оказаться в командном режиме. Дальше можно пробовать вводить новые команды.

Если ничего не получается даже после нажатия клавиши `Esc`, вероятнее всего включён не тот регистр клавиатуры. Прежде, чем продолжать работу, необходимо убедиться, что отжата клавиша `CapsLock` (включён нижний регистр) и включён английский (а не русский) регистр ввода.

Возможности редактора `vi` столь многообразны, что нет никакой возможности изложить здесь даже их половину. Желающим в совершенстве освоить редактор настоятельно рекомендуется обратиться к справочному руководству, встроенному в редактор — оно вызывается командой

```
:help
```

а выход из него как и обычно командой

```
:q
```

(к сожалению, руководство на английском языке) и к дополнительной литературе [4].

## Программирование

ОС Linux представляет собой достаточно удобную среду программирования. Правда само понятие удобства нуждается в уточнении.

Программисты, привыкшие к интегрированным средам при написании программ в ОС Windows и MS DOS, в начале работы в ОС Linux склонны утверждать обратное. В самом деле, в ОС Linux (да и вообще в ОС UNIX) при разработке серьёзных программ интегрированные среды использовать не принято. И это не какая-то необъяснимая блажь программистов, не «вселенский заговор линуксоидов», а следствие наличия в системе хорошо подобранного и сбалансированного набора мощного переносимого программного обеспечения. Этот комплект программного обеспечения программисту необходимо освоить лишь один раз. В дальнейшем он сможет найти его в любой установке ОС UNIX.

Например, для программирования на языке Си в стандартный набор входят:

- стандартный текстовый редактор (любой из освоенных программистом);
- транслятор (чаще всего `gcc`);
- редактор связей (чаще всего `ld`);
- утилита управления статическими библиотеками `ar`;
- утилита управления проектами `make`;

– отладчик (более простой символьный `gdb` или более удобный графический `ddd`).

К этому можно добавить язык командной оболочки `sh`, который всегда «под рукой» у пользователя системы. А также мощные пакеты сопровождения и конфигурирования проектов (`CVS`, `autoconf` и некоторые другие), обеспечивающие почти безграничные возможности по сохранению промежуточных версий разрабатываемой программы и её переносу в родственные ОС. Последнее в рамках разрекламированных интегрированных сред программирования других современных ОС практически отсутствует. В них эту работу программисту приходится делать вручную.

### *Языки, установленные в системе*

В учебной системе X-терминалов установлены следующие языки программирования:

- Си (компилятор `gcc`);
- C++ (компилятор `g++`);
- виртуальная машина Java (команда `kaffe`);
- Tcl/Tk (интерпретатор `wish`);
- perl (интерпретатор `perl`);
- python (интерпретатор `python`);

а также ряд специфических узкоспециализированных языков, включая `yacc`, `lex`, `awk`, `octave`, `gnuplot` и др.

Любой из названных языков предоставляется компилятором, либо интерпретатором, обрабатывающим программу, записанную в обычном текстовом файле.

### *Создание файла с исходным текстом программы*

Текст программы на любом языке программирования должен быть записан в обычный текстовый файл с помощью любого применяемого в системе текстового редактора (например, `vi`, либо `mcedit`).

Расширение файлу присваивают, исходя из обычных соображений. Например, для текста программы на языке Си файлу присваивают расширение `.c`, для заголовочного файла в программе на Си — расширение `.h` и т. п.

## *Трансляция программы*

Так как в учебной системе студенты программируют в основном на языке программирования Си, здесь будет описан только процесс трансляции программ, написанных на этом языке. За подробностями по работе с программами, написанными на других языках программирования пользователям необходимо обращаться к системе `man`.

Простейшая программа (текст которой полностью содержится в одном файле) транслируется командой

```
gcc file.c
```

где `file.c` — файл с текстом программы. При этом выполняется трансляция программы и редактирование связей. В результате получается сразу исполняемая программа (если в процессе трансляции или редактирования связей не было сообщений об ошибках), которая по традиции записывается в файл `a.out`.

По умолчанию редактирование связей выполняется только со стандартной библиотекой языка Си. Если в программе используются функции из других библиотек (например, математические), необходимо специально указывать на это компилятору (см. следующий пункт).

Если желательно записать исполняемую программу в файл с другим именем, то это можно указать в командной строке ключом `-o`, например

```
gcc -o file file.c
```

выполнит трансляцию программы в файле `file.c` и результирующую исполняемую программу запишет в файл `file`.

Обратите внимание, что в системе UNIX и, в частности, Linux давать исполняемым файлам расширения типа `.exe` не принято, хотя и возможно (тот факт, что файл является исполняемой программой определяется не расширением имени файла, а правами доступа к файлу в файловой системе и неким «магическим кодом», записываемым транслятором в первых байтах такого файла). Общепринятой практикой является давать исполняемым файлам имена без расширений.

Если не нужно получать исполняемую программу (например, при трансляции программы, состоящей из нескольких исходных файлов), то при трансляции необходимо использовать ключ `-c`:

```
gcc -c file.c
```

даст в результате объектный файл `file.o`, который в дальнейшем можно использовать для сборки в большую программу вместе с другими объектными файлами.



Возможна трансляция программы только до уровня ассемблерных кодов:

```
gcc -S file.c
```

даст на выходе файл `file.s` с эквивалентом исходной программы на языке ассемблера. Не пугайтесь: этот язык ассемблера является вариантом языка ассемблера `gcc` для процессоров Intel и не очень похож на привычный язык ассемблера, используемый, например, в ОС MS DOS, хотя и описывает команды того же самого процессора.

При трансляции можно указать степень оптимизации объектного кода. По умолчанию выполняется быстрая трансляция, но без оптимизации. Включение оптимизации предполагает увеличение времени на трансляцию в обмен на более компактный и быстрый объектный код. Оптимизация включается ключом `-O` с дополнительным указанием уровня сложности. Уровень может изменяться от 0 (не оптимизировать) до 3 (максимальная оптимизация). Используемый обычно уровень оптимизации — 2:

```
gcc -O2 file.c
```

Необходимо иметь в виду, что оптимизированную программу сложно отлаживать в экранных отладчиках, так как результирующий код может иметь мало общего с исходным текстом программы.

### *Редактирование связей*

Вообще, для этой цели в системе имеется редактор связей `ld`. Но в большинстве случаев гораздо удобнее использовать возможности по редактированию связей самого компилятора `gcc`—. Если ему в командной строке указать несколько объектных файлов, то он сам «понимает», что необходимо выполнить редактирование связей и собрать исполняемую программу. Например

```
gcc file1.o file2.o file3.o
```

соберёт из трёх объектных файлов исполняемый файл, который по традиции получит имя `a.out`. Для изменения имени итогового файла также, как и в предыдущем пункте, применяется ключ `-o`.

Если необходимо указать дополнительные библиотеки, то используется ключ `-l`. Интересно, что хотя все библиотеки в системе имеют названия, начинающиеся с префикса `lib`, этот префикс при указании в ключе опускается. Например, для подключения к программе библиотеки математических функций, называемой `libm`, ключ должен иметь вид `-lm`:

```
gcc file1.o file2.o -lm
```

В данном случае исполняемая программа собирается из двух объектных файлов с дополнительным подключением библиотеки математических функций.

## *Запуск программы*

Для запуска программы системе необходимо указать полный путь к файлу программы. Исключение составляют только программы, хранящиеся в каталогах, описываемых переменной окружения `$PATH`. Именно поэтому такие программы можно запускать простым указанием в командной строке имени файла программы без указания полного пути.

Программа, которая получилась в результате трансляции (пусть имя её файла для определённости `a.out`), находится в текущем каталоге, который в целях безопасности не включён в список путей переменной `$PATH`. Поэтому для её запуска обязательно нужно явно указать полный путь к её файлу. В нашем случае для этого достаточно указать, что файл находится в текущем каталоге (используется сокращение `./`). Команда запуска выглядит так:

```
./a.out
```

Если при трансляции использовался ключ `-o`, то имя `a.out` должно быть заменено на соответствующее имя файла, указанное в ключе.

## *Автоматизация процесса разработки программы*

Если программа состоит из нескольких файлов, причём не только файлов собственно текста программы (`.c`), но и заголовочных (`.h`), транслировать такую программу при отладке становится неудобно. Необходимо постоянно помнить какие из файлов надо перетранслировать после последних внесённых изменений, чтобы получилась правильная исполняемая программа.

На помощь в этом случае приходит стандартная утилита `make`. В соответствии с составленным пользователем файлом временных зависимостей (называемым `make-файлом`) она автоматически будет при каждом вызове запускать трансляцию только тех файлов, которые изменены с момента последней трансляции, либо зависят от файлов, изменённых с момента последней трансляции.

Утилита обладает большой гибкостью в настройке и позволяет управлять не только трансляцией программ, но и любыми другими работами в системе, основанными на временных зависимостях между файлами. Здесь рассмотрим только её базовые возможности.

Стандартным именем для `make-файла` считается `makefile`. В этом файле в простейшем случае достаточно описать имя файла цели (обычно это имя итоговой исполняемой программы) и имена объектных (!) файлов, от которых зависит файл цели (то есть объектных

файлов, из которых должна быть собрана объектная программа). За этой строкой зависимостей должна быть указана команда получения файла цели из зависимых объектных файлов. Дополнительно могут быть указаны зависимости объектных файлов от своих заголовочных файлов. Зависимость объектных файлов от файлов исходного текста считается известной по умолчанию. Команды получения объектных файлов из исходных также считаются известными по умолчанию, но при желании могут быть переопределены явно.

Важно знать, что имена целей всегда должны записываться с первой позиции строки и отделяться от зависимых имён двоеточием. Команда (команды), выполняемая для получения цели должна записываться после строки зависимости с отступом от начала строки в одну табуляцию (именно табуляцию, а не набор пробелов!).

Например, если исполняемая программа `prog` должна быть собрана из объектных файлов `file1.o` и `file2.o` и при этом файл `file1.o` зависит от файла `head.h` (то есть в файле `file1.c` есть директива `#include "head.h"`), то `make`-файл будет выглядеть следующим образом:

```
prog: file1.o file2.o
    gcc -o prog file1.o file2.o
file1.o: head.h
```

Создав файл `makefile` с таким содержимым в том же каталоге, где расположены файлы `file1.c`, `file2.c` и `head.h`, для трансляции и сборки программы `prog` достаточно дать команду

```
make
```

Если теперь внести изменения в какой-либо один исходный файл и снова дать команду `make`, то будет выполнена перетрансляция только изменённого файла и снова будет автоматически собрана исполняемая программа.

### *Отладка программы*

При разработке программы любой программист сталкивается с необходимостью её отладки. Для программ, разрабатываемых в рамках лабораторных работ, в качестве простейшего, но мощного средства отладки рекомендуется использование отладочной печати. При этом процедура отладки заключается во вставке в текст программы в намеченные программистом точки операторов печати, которые в процессе выполнения программы будут выдавать на экран информацию о прохождении программой этих точек и, возможно, значений в этих точках заданных программистом переменных.

При правильном использовании метод отладочной печати даёт очень хорошие результаты, не требуя от программиста изучения дополнительных, часто довольно сложных специализированных средств отладки.

При отладке всегда предполагается, что программист в общем хорошо представляет себе логику работы программы и может указать ожидаемые значения переменных на том или ином этапе работы программы в той или иной её части. Сравнивая эти (ожидаемые) значения переменных с реально выдаваемыми операторами отладочной печати значениями программист может судить о правильности или неправильности работы той или иной части программы. Путём наблюдений за поведением программы и своих умозаключений программист должен уметь найти ошибки, мешающие правильной работе программы.

Однако в ОС Linux имеются и другие (специализированные) средства отладки [5], среди которых наиболее знакомым большинству программистов по технологии применения является интерактивный отладчик. На самом деле их даже два: **gdb**, ориентированный на работу в режиме командной строки текстового терминала, и **ddd**, представляющий собой полнофункциональный многооконный отладчик для работы в системе XWindow.

Оба отладчика позволяют (с разной степенью удобства) отлаживать программы по исходному тексту. Оба предоставляют набор средств для задания точек останова, пошагового выполнения программы, просмотра и изменения значений переменных, регистров процессора и сложных структур данных. У обоих отладчиков схожий набор команд управления, за тем лишь исключением, что в **gdb** все команды надо набирать в командной строке, а в **ddd** кроме набора в командной строке команды могут быть отданы щелчком мыши по соответствующим командным кнопкам в экранном меню.

Для отладки с использованием любого из отладчиков программа должна быть оттранслирована и собрана транслятором **gcc** с использованием дополнительного ключа **-g**, требующего включить в исполняемый код программы блок отладочной информации.

Работа с отладчиками требует определённых навыков и не рекомендуется начинающим программистам. Применять отладчик имеет смысл лишь для поиска «загадочных» ошибок, неотлавливаемых или трудноотлавливаемых отладочной печатью.

Оба отладчика при работе предоставляют интерактивный доступ к собственной системе помощи, пользуясь которой можно довольно

быстро освоить основные команды. К сожалению, помощь предоставляется только на английском языке.

Учитывая, что для отладки программ при выполнении лабораторных работ вполне достаточно метода отладочной печати, здесь работа с отладчиками не рассматривается. Желающие поэкспериментировать с отладчиками могут сделать это самостоятельно.

### *Особенности перехвата стандартного потока вывода командой `tee`*

При выполнении лабораторных работ студентам рекомендуется писать программы, использующие для ввода и вывода данных стандартные потоки ввода и вывода (`stdin`, `stdout`). Такой подход обеспечивает простой механизм отладки программ и демонстрации их работы. А при необходимости включить результаты работы программы (данные, выводимые на экран терминала) в отчёт достаточно использовать запуск программы с перенаправлением вывода результатов в файл:

```
./prog >t
```

В данном примере `./prog` — запускаемая из текущего каталога программа, а `t` — файл на диске в который будет перенаправлен вывод результатов работы программы. После выполнения команды файл `t` можно непосредственно распечатать на принтере или отформатировать и включить в большой файл отчёта.

К сожалению, таким способом затруднительно выполнять программы, ориентированные на диалог с пользователем (на экране ничего не видно). Теоретически решением проблемы должно стать использование фильтра-разветвителя потока вывода `tee`:

```
./prog | tee t
```

В данном случае стандартный поток вывода программы `prog` перехватывается фильтром `tee`, который одновременно записывает его в файл `t` и в стандартный поток вывода, по-прежнему связанный с терминалом.

На практике же, хотя фильтр `tee` «честно» выполняет свои функции, на экране всё равно ничего не будет видно. Причина здесь в том, что по умолчанию стандартный поток вывода буферизуется. Пока буфер потока не будет заполнен, данные на экране не появятся.

Решением проблемы является отключение буфера стандартного потока вывода. Сделать это можно только внутри прикладной программы (никакая внешняя команда ОС для этого не предусмотрена).

Программисту необходимо включить в функцию `main()` своей программы где-нибудь поближе к её началу следующую строчку:

```
setvbuf(stdout, NULL, _IONBF, 0);
```

И теперь после перетрансляции программы и запуска её командой типа

```
./prog | tee t
```

Выходные данные будут одновременно сохраняться в файле `t` и выводиться на экран терминала по мере их поступления без задержки в буфере.

## Web программирование

Для изучения Web технологий представления данных в сети в системе X-терминалов установлен стандартный функционально полный набор серверного программного обеспечения: сервер Web «Русский Apache», СУБД PostgreSQL, язык динамической генерации страниц (с возможностью получения данных из СУБД) PHP4.

Возможность создавать свои страницы на языках HTML и PHP и демонстрировать их через стандартные клиенты Web открыта для всех пользователей системы X-терминалов с момента регистрации. Для получения доступа к СУБД с целью создания своей базы данных необходимо обратиться к администратору системы. Доступ пользователей к чужим базам данных может быть разрешён администратором только по согласованию с их владельцами.

Свои Web страницы пользователи должны размещать в своих домашних каталогах в подкаталогах WWW. При этом обращение к страницам в рамках системы X-терминалов выполняется по следующему URL

```
http://localhost/~login/file.HTML
```

где `login` — регистрационное имя пользователя в системе X-терминалов, а `file.HTML` — запрашиваемый файл в подкаталоге WWW домашнего каталога пользователя (файлу стартовой страницы рекомендуется давать стандартное имя `index.HTML`).

Таким образом домашние страницы любого пользователя могут быть доступны любому другому пользователю. Достаточно только знать регистрационное имя этого пользователя в системе.

## *Расположение файлов в подкаталоге WWW*

Непосредственно в подкаталоге WWW должен размещаться стартовый файл страницы пользователя `index.HTML`. Все остальные файлы страницы пользователя могут размещаться как непосредственно в этом же каталоге, так и в подкаталогах данного каталога. При этом контроль правильности указания URL файлов возлагается на самого пользователя, разработавшего страницу. Контролировать необходимо не только собственно путь, но и регистр букв, которым заданы путь, имя и расширение каждого файла. Для исключения недоразумений рекомендуется все пути, имена и расширения файлов как в URL, так и на диске записывать только строчными английскими буквами.

**Важно помнить, что в системе X-терминалов для правильной обработки и представления страниц русский текст во всех файлах HTML и PHP должен быть представлен в кодировке `koï8-r`.**

## *Обращение к своим страницам Web*

Страницы пользователя, записанные в подкаталог WWW, доступны в системе X-терминалов через URL

`http://localhost/~login/file.HTML`

где `login` — регистрационное имя пользователя в системе X-терминалов, а `file.HTML` — запрашиваемый файл в подкаталоге WWW домашнего каталога пользователя.

Для обращения по URL и просмотра страниц возможно использование любого клиента Web. В системе X-терминалов установлено несколько клиентов Web, отличающихся по функциональным возможностям и потребляемым ресурсам. В связи с ограниченными ресурсами системы X-терминалов пользователи при выполнении лабораторных работ должны стремиться к использованию наименее требовательных к ресурсам клиентов, переходя к более требовательным лишь при необходимости.

Далее дано описание имеющихся в системе X-терминалов клиентов Web в порядке возрастания их возможностей, удобства использования и потребляемых ресурсов системы.

**Клиент Web lynx.** Простой клиент Web, ориентированный на работу в текстовом режиме (непосредственно в окне терминала). Современные версии поддерживают работу с таблицами и фреймами, а также формами, созданными на стандартном HTML, что в принципе

достаточно для подавляющего большинства приложений. Особо следует отметить великолепную поддержку локализаций и, в частности, полный набор кодировок русского языка.

Клиент не поддерживает никаких динамических расширений HTML, работающих на стороне клиента. В общем случае не позволяет просматривать нетекстовые объекты (картинки, файлы pdf и т.п.), но позволяет загрузить эти объекты на локальный диск для дальнейшего их просмотра другими программами.

Перечисленные ограничения, как ни странно, могут являться большим плюсом и значительно ускорять работу с сетью, так как исключают необходимость загрузки и обработки на стороне клиента большого объёма лишней (графической рекламной) информации.

К сожалению, многие современные сайтостроители проектируют свои сайты с единственной целью — заложить в них как можно больше «современных технологий» (зачастую совершенно не нужных для действительных целей сайта). В результате такие сайты в принципе теряют возможность получения с них информации без клиента, поддерживающего различные скриптовые языки. Из-за того, что такие сайты время от времени встречаются в сети, ценность клиента lynx снижается.

Клиент lynx имеет развитую систему команд, что позволяет пользователю довольно комфортно чувствовать себя в сети Internet при условии поиска, просмотра и сохранения чисто текстовой информации.

**Клиент Web Chimera.** Это уже графический клиент, позволяющий в большинстве случаев просматривать страницы HTML вместе со вставленными в них рисунками. К его достоинствам относится малая требовательность к ресурсам системы, что позволяет использовать его одновременно всем пользователям системы X-терминалов. В связи с этим он рекомендуется для отладки несложных страниц HTML.

Вместе с тем этот клиент имеет весьма скромные возможности по интерпретации языка HTML и не поддерживает многие его расширения, ставшие общепринятыми в последнее время. Не имеет также никаких средств для выполнения встроенных в страницы скриптов (JavaScript и пр.). Имеет проблемы (при использовании русского языка) с обработкой страниц HTML-форм.

Эти и (к сожалению) многие другие недоработки не позволяют рекомендовать его к использованию при выходе в сеть Internet и локальную сеть университета.



**Клиент Web mozilla.** Полноценный клиент Web, теоретически поддерживающий все современные технологии, с помощью которых сегодня создаются страницы в сети Internet. Поддержка многих технологий осуществляется специальными модулями (plug-in'ами), большинство из которых в базовый комплект не входит и должно загружаться из сети. На практике не все из загружаемых модулей правильно реализуют свои функции, так что реально обычному пользователю бывают доступны только наборы основных модулей, для наиболее распространённых технологий. Но и этого для повседневной работы более, чем достаточно.

За богатые возможности приходится дорого платить. Клиент долго загружается и занимает в оперативной памяти довольно много места. В условиях многопользовательской системы X-терминалов это означает, что одновременно с этим клиентом могут работать не более 3–5 пользователей.

В связи с этим для запуска клиента **mozilla** необходимо получить разрешение у дежурного инженера класса X-терминалов.

### *Подключение базы данных SQL*

В системе X-терминалов установлена система управления базами данных (СУБД) PostgreSQL. Эта СУБД потенциально позволяет каждому пользователю создать и вести свою базу данных, самостоятельно управляя доступом к ней других пользователей.

Для получения доступа к СУБД и создания своей базы данных пользователь должен обратиться к дежурному инженеру системы. Все дальнейшие операции со своей базой данных пользователь должен выполнять сам.

Все утилиты управления базой данных расположены в системе X-терминалов в отдельном каталоге

```
/usr/local/pgsql/bin
```

Поэтому для простоты доступа к базе данных рекомендуется добавить этот путь к списку своих стандартных путей поиска (переменная окружения PATH). Впрочем, возможен и запуск утилит просто с указанием полного пути.

Основной утилитой работы с базой данных является командная оболочка **psql**. Её запуск выполняется командой

```
psql
```

если путь к ней прописан в переменной PATH или командой

```
/usr/local/pgsql/bin/psql
```

в противном случае.

Оболочка управляет базой данных с помощью запросов, формулируемых пользователем на стандартном языке SQL в командной строке. Результаты запросов выводятся на экран терминала. Кроме этого, оболочка предоставляет пользователю ряд дополнительных команд, облегчающих работу с базой данных.

Прежде чем начинать какой-либо серьёзный проект, студентам настоятельно рекомендуется потренироваться в формулировке и выполнении SQL-запросов.

Для резервирования созданной базы данных пользователю рекомендуется регулярно делать её дампы в файл и копировать этот файл себе на дискету. Дамп базы данных может быть получен следующей командой:

```
/usr/local/pgsql/bin/pg_dump имя >файл
```

где **имя** — имя базы данных (обычно совпадает с регистрационным именем пользователя), а **файл** — файл, в который записывается дампы.

## Создание документов в системе Т<sub>E</sub>X

В настоящем учебном пособии описывается только порядок работы с системой Т<sub>E</sub>X. Язык системы Т<sub>E</sub>X превосходно и полностью описан в [6]. Кроме того, в объёме, достаточном для выполнения несложной вёрстки, язык Т<sub>E</sub>X изучается в лекционном курсе.

### *Создание исходного файла документа*

Исходный документ Т<sub>E</sub>X может быть создан любым текстовым редактором, установленным в системе. Файл с текстом на Т<sub>E</sub>X должен иметь традиционное расширение **.tex**.

Файл можно подготовить и «на стороне», а затем внести в учебную систему. Но при этом необходимо помнить, что текст в файле, подготовленном для трансляции Т<sub>E</sub>X'ом в учебной системе, должен быть представлен в кодировке **koі8-g** и с концами строк, соответствующими стандарту ОС UNIX (только символ перевода строки).

Для того, чтобы Т<sub>E</sub>X «понимал» русский текст и правильно расставлял русские переносы в самом начале файла необходимо записать команду

```
\Russian
```

а чтобы по достижении конца файла работа Т<sub>E</sub>X'а автоматически завершалась с возвратом в ОС в конце файла необходимо записать команду

`\bye`

Если в тексте встречается значительный абзац на английском языке, требующий использования английских переносов, то переключиться на них можно, записав команду

`\English`

а для возврата к русским переносам — снова команду

`\Russian`

### *Трансляция документа*

Трансляция выполняется командой

`cyrtex file.tex`

где `file.tex` — файл с документом на  $\TeX$ .

В процессе трансляции  $\TeX$  будет сообщать о найденных ошибках. При этом после каждого сообщения он будет останавливаться и ждать помощи пользователя. В любом случае рекомендуется внимательно прочитать и попытаться понять сообщение об ошибке прежде, чем пытаться что-либо предпринять. Возможны два принципиально разных типа ошибок, порождающих разные запросы к пользователю и требующих разной реакции.

Первый тип ошибки — не найден какой-либо из заданных на чтение файлов. Сообщение  $\TeX$ 'а будет содержать предложение ввести правильное имя файла и закончится символом `:.`

Можно ввести правильное имя файла (если оно известно). Тогда, если файл будет найден,  $\TeX$  продолжит трансляцию. А можно просто нажать `[Ctrl] [D]` и вернуться в ОС.

Второй тип ошибки — неправильно записанная команда или конструкция самого языка  $\TeX$ 'а. Транслятор выдаст диагностическое сообщение о том, что именно ему не понравилось и закончит его символом `?`.

В этом случае возможно несколько вариантов ответа, но наиболее рациональными с точки зрения современных систем являются два:

`x` — проигнорировать ошибку и попытаться продолжить трансляцию, остановившись на следующей ошибке;

`q` — продолжить трансляцию в пакетном режиме, игнорируя все ошибки и ничего не выдавая на терминал (все сообщения будут записаны в файл `log-a`).

В результате трансляции формируются следующие файлы:

`file.dvi` — целевой результат трансляции, который теперь можно просмотреть и распечатать;

`file.log` — файл со всеми сообщениями, выданными системой на экран (или подавленными в пакетном режиме работы) в процессе трансляции документа.

### *Просмотр документа*

Для просмотра документа `dvi` в системе X-терминалов установлены две программы (так называемых `dvi`-драйвера): `xdvi` и `dvilx`.

**Программа `xdvi`** является стандартной программой просмотра документов `dvi` в системе `teTeX`, установленной сегодня практически во всех дистрибутивах ОС Linux (в том числе и в учебной системе X-терминалов). Она запускается и работает, но, к сожалению, по непонятной причине в системе X-терминалов использует для просмотра документа неверные шрифты. (За пределами учебного класса X-терминалов на других установках ОС Linux эти странности в её работе не проявляются.)

Поэтому её использование в классе не рекомендуется, и здесь она не описана.

**Программа `dvilx`** является версией для XWindow программы `tmView` независимого разработчика (не касающегося поддержки системы `teTeX`). Она великолепно работает в системе X-терминалов и рекомендуется к использованию.

Запуск программы осуществляется командой

```
dvilx file.dvi
```

где `file.dvi` — файл, который необходимо просмотреть.

Программа имеет множество возможностей по управлению с клавиатуры. Вот основные:

`q` — выход;

`+` — увеличить масштаб;

`-` — уменьшить масштаб;

`PgUp` — предыдущая страница;

`PgDn` — следующая страница;

стрелки — пошаговое перемещение по странице.

## *Печать документа*

Используя различные драйверы dvi, в принципе (теоретически) можно напечатать документ dvi на любом печатающем устройстве.

В системе X-терминалов печать выполняется с использованием драйвера dvi `dvips`, который конвертирует документ dvi в формат ps (Postscript), который «понятен» системе печати, установленной в системе X-терминалов.

Для печати документа необходимо дать команду

```
dvips file.dvi
```

где `file.dvi` — печатаемый файл.

## *Вставка иллюстраций в документ*

Из всех возможных способов вставки иллюстраций в документ на T<sub>E</sub>X'e рассмотрим лишь один, наиболее применяемый в последнее время. Его идея заключается в том, что в документ с помощью специальных команд вставляются картинки только в формате ps. Далее документ tex транслируется с получением формата dvi, а затем документ dvi конвертируется в формат ps, который уже может быть просмотрен вместе с иллюстрациями как единое целое, распечатан или сконвертирован в формат pdf для просмотра на экране на других платформах.

Формат ps является векторным, но в нём могут быть представлены и иллюстрации, которые в принципе не поддаются векторизации (например, фотографии). Иными словами, в формат ps могут быть сконвертированы практически любые графические изображения (вопросы качества конвертирования и последующего применения здесь не обсуждаются).

Конвертирование растровой картинки (например формата png) в формат ps выполняется командой

```
convert file.png file.ps
```

где `file.png` и `file.ps` соответственно имена файлов соответствующих форматов.

Вопросы получения «настоящих» векторных изображений в формате ps обсуждаются в следующем разделе.

Вставку изображения формата ps (точнее, eps — Encapsulated Postscript, но в данном случае это не очень важно) в документ на T<sub>E</sub>X'e обеспечивает пакет макрокоманд `epsf`. Для его использования в начале файла необходимо записать команду

```
\input epsf
```

а в том месте, куда необходимо вставить изображение, команду

```
\epsfbox{file.ps}
```

где `file.ps` — вставляемый файл изображения.

Получившийся документ транслируется в формат `dvi` обычным образом. Но просмотреть его стандартной программой просмотра `dvi` в общем случае нельзя.

Далее для получения документа в формате `ps` необходимо дать команду

```
dvips -o file.ps file.dvi
```

которая выполнит конвертирование файла `file.dvi` в формат `ps` и запишет результат в файл `file.ps`.

Получившийся документ можно просмотреть командой

```
gv file.ps
```

а напечатать на принтере стандартной командой

```
lpr file.ps
```

Команда

```
ps2pdf file.ps
```

выполнит конвертирование документа в формат `pdf` и запишет результат в файл `file.pdf` (это имя в командной строке можно явно не указывать). Получившийся документ в формате `pdf` нормально просматривается программой `AcrobatReader` старших версий на всех платформах, а также программой `gv` в ОС `Linux` (но не просматривается в ОС `Linux` программой `xpdf`).

## Работа с графическими файлами

Современный `Linux` хорошо «приспособлен» для работы с растровыми графическими файлами. Возможны следующие операции:

- захват копии экрана (окна);
- конвертирование между различными форматами файлов;
- редактирование изображений с достаточно широкими возможностями;
- создание новых изображений (рисование);
- просмотр графических файлов различных форматов.

Операции конвертирования, редактирования и создания изображений могут быть выполнены как в интерактивном, так и в пакетном режимах.

Работа с векторными графическими файлами в `Linux` всё ещё затруднена, особенно если речь заходит о векторных файлах, содержащих надписи на русском языке. Имеется несколько более или менее проработанных программ работы с векторными изображениями, в

том числе создания и редактирования изображений. Но большинство из них несовместимы по форматам создаваемых файлов, не могут работать с надписями на русском языке, а зачастую не поддерживают и полноценную конверсию в стандартные векторные форматы.

Справедливости ради заметим, что имеется несколько неплохо проработанных пакетов для построения различных диаграмм и графиков по наборам данных. Получившиеся изображения обычно могут быть сохранены в формате Postscript, что позволяет использовать их в дальнейшем не только для просмотра на экране и распечатки на принтере, но и для вставки в печатные документы (формат Postscript является стандартом *de facto* в полиграфии). Но обычно чем лучше проработан такой пакет, тем меньше он ориентирован на графический интерфейс — в хорошем пакете, как правило, упор сделан на специализированный управляющий язык программирования и работу в пакетном режиме.

В связи с этим далее будут рассмотрены в основном средства работы с растровыми графическими файлами. Исключение будет сделано только для программы `lgv` —, предназначенной для просмотра файлов `ps` и `pdf`.

В случае крайней необходимости для получения векторного изображения в формате `ps` из растрового могут быть использованы программы векторизации `potrace` и `autotrace`, а также программа `pstoedit`, позволяющая конвертировать изображения `ps` в некоторые другие векторные форматы (далеко не во все). Но результат их применения не всегда оправдывает ожидания и потраченное время.

### *Снятие копии экрана*

Здесь будет рассмотрено снятие копии только графического экрана в системе XWindow.

В ОС Linux, вообще говоря, существует несколько способов снятия копий экрана. Каждый из них обусловлен возможностями, предоставляемыми различными программами обработки графики и даже некоторыми графическими оболочками. Здесь будет рассмотрен «базовый» способ, предоставляемый самой системой XWindow. Возможно, он не самый удобный, зато работоспособен практически на любой установке.

Идея способа заключается в использовании утилиты `xwd` (видимо аббревиатура от «X Window Dump»), поставляемой в базовом комплекте XWindow. После запуска утилиты в командной строке окна эмулятора терминала курсор мыши меняет свою форму. Теперь

необходимо щёлкнуть курсором мыши по окну, графическую копию которого необходимо сохранить. Результат будет выдан в стандартный поток вывода.

В связи с особенностями управления окнами в оконном менеджере IceWM, а также малыми размерами графического экрана в учебной системе X-терминалов для практического применения данного способа лучше всего действовать следующим образом:

1) определиться с номером рабочего стола, на котором открыто окно, с которого необходимо снять копию. Предположим, что это рабочий стол № 1 (это стол по умолчанию, и большинство пользователей работают только с ним);

2) перейти на соседний рабочий стол (№ 2), нажав клавиши **Ctrl** **Alt** **2**;

3) запустить на этом рабочем столе программу эмулятора терминала (любую);

4) в командной строке терминала набрать команду

```
xwd >file
```

где **file** — имя файла, в который будет сохранено изображение окна. Имя файла лучше всего набирать без расширения. Формат получившегося файла не является широко распространённым, но «известен» многим программам работы с графикой для ОС Linux. После запуска команды курсор мыши поменяет форму;

5) перейти на предыдущий рабочий стол. В нашем случае (стол № 1) для этого надо нажать клавиши **Ctrl** **Alt** **1**;

6) щёлкнуть курсором мыши на окне, копию которого надо снять. При этом раздастся звуковой сигнал — в начале процесса сохранения, а затем второй сигнал — по окончании процесса сохранения. На диске формируется файл с заданным именем, а программа **xwd** автоматически завершает работу.

Просмотреть получившийся файл можно с помощью утилиты **xwud**, поставляемой в комплекте системы XWindow, (команда выглядит так:

```
xwud file
```

завершение просмотра нажатием любой клавиши в окне) или например программы **display** из комплекта системы |ImageMagick—, установленной в учебной системе.

Конвертировать получившуюся копию в другой формат можно программой **convert** из комплекта того же |ImageMagick—. Например, команда конвертирования в формат Postscript для последующей печати на принтере или вставки в документ выглядит так:



```
convert file file.ps
```

Команды `xwd` и `xwud` имеют ряд управляющих ключей, расширяющих возможности их применения. Как всегда с ними можно познакомиться в справочном руководстве `man`.

### *Программа xv*

Очень старая, но добротна сделанная программа просмотра графических файлов. Обладает удобным, хотя и немного непривычным пользовательским интерфейсом. Быстро стартует. Присутствует не во всех дистрибутивах Linux, но может быть легко собрана из свободно распространяемых исходных кодов практически в любом дистрибутиве. В силу перечисленных особенностей настоятельно рекомендуется к освоению и использованию по назначению.

Программа запускается командой

```
xv
```

При этом на экране появляется окно с заставкой. Для вызова окна со списком каталогов и файлов необходимо нажать клавиши `Ctrl V`.

Можно запустить программу для просмотра конкретного файла. В этом случае команда запуска выглядит так

```
xv picture
```

где `picture` — имя файла картинки.

Работа программы завершается по нажатию клавиш `Ctrl Q`.

Программа обладает рядом интересных возможностей, делающих её полноценным менеджером графических файлов. Здесь эти возможности не описываются.

### *Система ImageMagick*

Это целый комплекс программ, предоставляющий широкие возможности по работе с графическими файлами. Среди них:

- просмотр файлов;
- конвертирование между разными графическими форматами;
- коррекция (глубины цвета, границ изображения, размера изображения и многих других параметров);
- фильтрация (достаточно большой набор из наиболее распространённых фильтров);
- нанесение надписей на изображениях;
- синтез новых изображений;
- деление изображения на части и сборка изображения из частей

а также многое другое. Система рассчитана в основном на использование своих компонентов в пакетном режиме, что ограничивает возможности её использования конечными пользователями, но с другой стороны даёт возможность создания систем автоматического получения или преобразования графических изображений.

Среди наиболее полезных и простых в использовании укажем:

– **display** — программа для просмотра графических файлов.

Запускается командой

```
display file
```

где **file** — просматриваемый графический файл. Программа также имеет некоторые возможности по интерактивной обработке просматриваемого изображения, которые здесь не рассматриваются;

– **convert** — программа для конвертирования графических файлов между разными графическими форматами. Запускается командой

```
convert file.fmt1 file.fmt2
```

где **file.fmt1** — файл исходного формата, а **file.fmt2** — файл целевого формата. Обычно поддерживаются все наиболее распространённые форматы, включая **psx**, **tiff**, **jpeg**, **ps**, **pnm**. Кроме этого именно эта программа обеспечивает многие типы коррекции и преобразования графических изображений, которые могут быть заданы ключами командной строки. Список возможностей программы огромен. Для ознакомления с ним отправим читателя к руководству **man**.

## *Редактор GIMP*

Это замечательный интерактивный графический редактор, приближающийся по своим возможностям к знаменитому Photoshop и ориентированный на работу с растровыми изображениями. Название является аббревиатурой от «GNU Image Manipulation Program» (вольный перевод: программа GNU для преобразования изображений). Редактор уже достаточно хорошо отработан и может быть рекомендован к самому широкому использованию. Однако его развитие продолжается, исправленные и дополненные новыми возможностями версии выходят с завидной регулярностью.

Запускается редактор командой

```
gimp
```

В командной строке можно дополнительно указать имя файла, который будет редактироваться.

Возможности применения редактора в учебном классе ограничиваются низким разрешением большинства терминалов и малой глубиной их цветности.

Редактор потребляет существенные системные ресурсы, поэтому для его запуска в учебном классе необходимо получить разрешение дежурного инженера.

### *Программа gv*

Строго говоря, **gv** — это не самостоятельная программа, а графический интерфейс к системе **ghostview**, предназначенной для преобразования, просмотра и печати файлов в формате Postscript.

Программа позволяет просматривать файлы форматов ps и pdf (последний через неявное преобразование в ps).

Запуск программы осуществляется командой

```
gv file
```

где **file** — файл формата ps или pdf, который необходимо просмотреть. Возможен запуск без указания имени файла с последующим выбором просматриваемого файла через меню.

Программа имеет непривычную систему управления. Все элементы управления просмотром располагаются сверху и слева. В частности, слева расположены список страниц и непривычный управляемый мышкой «джойстик» (вложенный прямоугольник), обеспечивающий перемещение окна просмотра по текущей странице документа.

### **Обработка данных**

Описать все возможности, предоставляемые учебной системой X-терминалов, не представляется возможным. Но всё же обратим внимание пользователей на некоторые фильтры, утилиты и пакеты обработки данных, которые могут оказаться полезными, правда, при условии, что пользователи найдут время изучить порядок работы с ними:

**awk** — мощный фильтр обработки и анализа табличной информации в пакетном режиме;

**sed** — пакетный (поточковый, неинтерактивный) текстовый редактор;

**gnuplot** — система построения 2- и 3-мерных графиков по наборам данных с выводом их на экран или в файл заданного графического формата, включая ps. Может работать как по программе, предварительно записанной в файл, так и в интерактивном режиме;

**plotutils** — набор программ, позволяющих получать графики в формате ps по наборам исходных данных, а также интерполировать сплайны и решать дифференциальные уравнения первого порядка;

**octave** — мощная система математического анализа данных, аналогичная по функциям коммерческой системе MathLAB;

**sc** — простая программа электронных таблиц для текстового терминала. Система команд близка к редактору **vi**;

**oleo** — более мощная программа электронных таблиц, которая может работать как в режиме текстового терминала, так и в XWindow (но не ждите от неё возможностей Excel!). Система команд базируется на редакторе Emacs.

## На досуге

ОС UNIX практически с самого начала своего существования была оснащена значительным набором средств, обеспечивающих взаимодействие работающих в ней пользователей. Современная ОС Linux, сохранив многие из стандартных средств ОС UNIX, добавила к ним многие новые. Некоторые из этих средств установлены в учебной системе.

### *Письмо другу (mail)*

В системе имеется возможность общения пользователей, работающих за разными терминалами. Для этого существует несколько различных средств (**write**, **talk** и др.), но самым удобным из них является команда **mail**, позволяющая послать корреспонденту письмо.

Команда **mail** существует в ОС UNIX с незапамятных времён и на самом деле является программой-клиентом к системе пересылки почтовых сообщений, в качестве которой сегодня обычно выступает какой-либо почтовый сервер (например, **sendmail** или **exim**). Если сервер подключён к сети Internet и правильно настроен, то с использованием команды **mail** в принципе можно отправить и прочитать письмо через службу E-mail. Единственной проблемой здесь является то обстоятельство, что программа **mail** предназначена для отправки и приёма только текстовых сообщений. Отправлять с её помощью бинарные файлы затруднительно, а читать практически невозможно.

Но для отправки корреспонденту короткого текстового сообщения программа является почти идеальным средством. Преимуществом программы **mail** перед другими средствами общения является то, что адресат совсем не обязательно должен в этот момент работать

в системе. Сообщение будет сохранено в его почтовом ящике. И когда адресат войдёт в систему, ему на терминал будет выдано сообщение

**You have mail**

Если же адресат уже работает в системе, то он получит такое сообщение как только выйдет в командную строку. Это снижает оперативность общения, но, во-первых, не мешает адресату работать с текущей программой, а во-вторых, гарантирует, что сообщение не будет потеряно.

Появление на экране сообщения

**You have mail**

означает, что в почтовом ящике пользователя появились новые письма. Для запуска процесса чтения писем достаточно дать в командной строке команду

**mail**

При этом на терминале появится список заголовков почтовых сообщений, находящихся в почтовом ящике пользователя. Каждая строка списка соответствует отдельному сообщению. Все строки пронумерованы.

Приглашение командной строки изменится на **&**, что является признаком работы в программе **mail**.

Для получения подсказки о командах достаточно ввести знак **?** и нажать **Enter**.

Список команд довольно обширен. Прокомментируем здесь только наиболее полезные.

Выход в приглашение командной строки программы **mail** и из программы осуществляется командой **q**, за которой следует **Enter**.

Если были прочитаны не все сообщения и необходимо продолжить их чтение в следующем сеансе, то вместо **q** надо использовать **x**.

Для чтения какого-либо сообщения достаточно набрать его номер и нажать **Enter**.

Для повторного получения списка заголовков сообщений надо ввести **f** и нажать **Enter**.

Для удаления текущего читаемого сообщения надо ввести **d** и нажать **Enter**.

Программа **mail** имеет довольно мощные возможности по управлению полученными сообщениями. Но поскольку в данном случае речь идёт о посылке простых сообщений от пользователя к пользователю, рекомендуется все прочитанные сообщения тут же удалять,

чтобы не занимать место на диске системы. Поэтому другие возможности здесь не описаны. Заинтересовавшиеся как всегда могут прочесть сопроводительную документацию.

Чтобы отправить сообщение пользователю системы необходимо знать его регистрационное имя (login). Для дальнейшего изложения предположим, что пользователь, которому посылается сообщение, имеет регистрационное имя в системе `qwerty`.

Для отправки сообщения пользователю необходимо дать команду

```
mail qwerty
```

После нажатия `Enter` будет выдан запрос

```
Subject:
```

который означает запрос однострочной темы сообщения. Эта строка будет выведена в заголовке сообщения при просмотре получателем. Тему желательно ввести, но можно и проигнорировать. В любом случае ввод этой строки должен заканчиваться нажатием `Enter`.

Далее безо всякого дополнительного приглашения нужно вводить текст сообщения. Сообщение может содержать несколько строк. Для того, чтобы закончить ввод сообщения нужно после очередного нажатия `Enter` в первую позицию новой строки ввести символ «точка» (.) и снова нажать `Enter`.

Будет выдано приглашение

```
Cc:
```

означающее запрос адреса, по которому необходимо отправить копию письма. Обычно его можно игнорировать и просто нажать `Enter`.

После этого письмо будет отправлено.

При таком способе отправки письма невозможно исправить ошибки, допущенные при наборе текста. Проблема решается стандартным для ОС UNIX способом. Текст письма предварительно набирается в любом текстовом редакторе и сохраняется в файл на диске. А затем стандартный поток ввода команды `mail` подменяется вводом из этого файла.

Пусть для нашего примера файлу будет дано имя `tm`. Посылка этого письма (текста, а не файла!) выполняется командой

```
mail -s "тема" qwerty < tm
```

Двойные кавычки вокруг темы позволяют ей состоять из нескольких слов (содержать пробелы).

### *Коллективная игра*

При наличии у студентов свободного от учёбы времени, например, после успешной сдачи экзамена или зачёта, в классе X-терминалов возможен запуск коллективной (сетевой) игры.

В системе установлено две игры:

`xscorch` — аналог известной игры «scorcher» из ОС MS DOS;

`freeciv` — аналог не менее известной игры «Civilisation» из ОС MS DOS и Windows.

Обе игры поддерживают групповую игру по сети. При этом все игроки должны начать игру одновременно (невозможно присоединиться позже, когда игра уже в разгаре) и каждый играет за себя.

Перед началом игры кто-то один должен запустить программу-сервер. Далее все игроки запускают программы-клиенты и соединяются с запущенным сервером. Игра начинается (в зависимости от игровой программы) подачей условной команды либо в программе-клиенте, либо в программе-сервере.

Детали (при наличии свободного игрового времени у студентов) можно узнать непосредственно в классе X-терминалов у дежурного инженера.

## ПРИЛОЖЕНИЕ

Операционная система Linux имеет чрезвычайно богатый программный инструментарий, который дополняется исчерпывающими справочными руководствами. Проблема освоения системы обычно состоит лишь в том, что для получения подсказки по какой-либо конкретной программе или команде необходимо знать, что программа (команда), выполняющая нужное действие в системе есть, а также необходимо знать её название. Для упрощения поиска нужных команд здесь приводится перечень наиболее используемых команд с краткими пояснениями их назначения.

### Манипулирование файлами

`ls` — просмотр списка файлов в каталоге;  
`cp` — копирование файлов;  
`mv` — перемещение файлов;  
`mkdir` — создание каталога;  
`cd` — переход в другой каталог;  
`rmdir` — удаление каталога;  
`rm` — удаление файлов;  
`mc` — файловый менеджер;  
`mcedit` — простой текстовый редактор;  
`vi` — мощный текстовый редактор;  
`find` — поиск файла по имени;  
`grep` — поиск файла по содержащемуся в нём тексту.

### Работа с дискетами

`mdir` — просмотр каталога дискеты;  
`mscopy` — копирование файла с дискеты или на дискету;  
`mdel` — удаление файла на дискете;  
`mdeltree` — удаление дерева файлов и каталогов на дискете.

### Перекодирование файлов

`iconv` — программа перекодирования;  
`recode` — программа перекодирования.



## Работа с файлами архивов

`tar` — создание/распаковка архивов `tar/tgz`;  
`gzip` — создание файлов `gz` (сжатие);  
`gunzip` — распаковка файлов `gz` (разжатие);  
`zip` — создание архивов `zip`;  
`unzip` — распаковка архивов `zip`;  
`cpio` — создание/распаковка архивов `cpio`.

## Печать

`lpr` — отправить файл на печать;  
`lpq` — посмотреть очередь печати;  
`lprm` — удалить задание из очереди печати;  
`iconv` — программа перекодирования;  
`recode` — программа перекодирования;  
`fold` — простое форматирование текста;  
`pr` — форматирование текста для печати.

## Работа с графикой

`xv` — просмотр растровых изображений;  
`display` — просмотр растровых изображений;  
`convert` — конвертирование, обработка и фильтрация изображений;  
`gimp` — редактирование растровых изображений;  
`gv` — просмотр векторных изображений `ps` и `pdf`.

## Программирование

`gcc` — транслятор `C`;  
`ld` — редактор связей;  
`ar` — программа управления статическими библиотеками;  
`make` — утилита управления трансляцией проекта;  
`gdb` — терминальный отладчик;  
`ddd` — графический отладчик;  
`yacc` — генератор программ синтаксического разбора;  
`lex` — генератор программ лексического разбора;  
`wish` — интерпретатор `Tcl/Tk`;  
`kaffe` — интерпретатор `Java`;  
`perl` — интерпретатор `perl`;  
`python` — интерпретатор `python`.

## Вёрстка на **T<sub>E</sub>X**

**cyrtex** — трансляция исходного файла с получением dvi;  
**dvilx** — просмотр файла dvi на экране;  
**dvips** — конвертирование файла dvi в формат ps и печать на принтере;  
**gv** — просмотр файлов ps и pdf;  
**ps2pdf** — конвертирование файла ps в формат pdf.

## Просмотр страниц **Web**

**lynx** — мощный клиент для текстового терминала;  
**chimera** — примитивный графический клиент;  
**mozilla** — мощный графический клиент.

## Обработка информации

**awk** — фильтр обработки табличной информации;  
**sed** — пакетный редактор текста;  
**gnuplot** — программа построения графиков по наборам данных;  
**plotutils** — набор программ: построение графиков по наборам данных, интерполирование сплайнов, решение дифференциальных уравнений;  
**octave** — мощная система обработки и анализа данных;  
**sc** — простые электронные таблицы для текстового терминала;  
**oleo** — более мощные электронные таблицы.

### Библиографический список

1. Керниган Б. В., Пайк Р. UNIX — универсальная среда программирования: Пер. с англ. — М.: Финансы и статистика, 1992. — 304 с.
2. Рассел Ч., Кроуфорд Ш. UNIX и Linux: книга ответов: Пер. с англ. — СПб.: Питер, 1999. — 304 с.
3. Тейнсли Д. Linux и UNIX: программирование в shell. Руководство разработчика: Пер. с англ. — Киев: Издательская группа BHV, 2001. — 464 с.
4. Вопросы и ответы по UNIX / Составление и пер. с англ. А. Крупника. — М.: МикроАрт, 1998. — 224 с.
5. Митчелл М., Оулдем Д., Самьюэл А. Программирование для Linux. Профессиональный подход: Пер. с англ. — М.: Вильямс, 2002. — 288 с.
6. Кнут Д. Всё про TeX: Пер. с англ. — Протвино: АО RDT<sub>E</sub>X, 1993. — 592 с.

## Оглавление

Предисловие .....	3
Глава 1. Система X-терминалов .....	5
Построение системы .....	5
Возможности, предоставляемые системой .....	6
Ограничения системы .....	8
Программное обеспечение, установленное в системе .....	8
Глава 2. Стандартное окружение пользователя .....	10
Домашний каталог пользователя .....	10
Файлы в домашнем каталоге .....	10
Подкаталог WWW .....	11
Подкаталог public .....	11
Настройка рабочего окружения .....	11
Выбор оконного менеджера .....	12
Настройка IceWM .....	13
Запуск файлового менеджера рабочего стола .....	16
Настройка свойств задач (widgets) .....	17
Просмотр шрифтов, доступных в системе .....	19
Глава 3. Выполнение часто встречающихся операций .....	21
Вход в систему .....	22
Выход из системы .....	22
Выполнение команд в стандартном экране пользователя .....	23
Запуск программ из меню .....	23
Запуск программ из панели задач .....	24
Переключение между окнами запущенных задач .....	24
Смена активного рабочего стола .....	24
Работа с эмулятором терминала .....	25
Запуск терминального менеджера файлов .....	26
Быстрый повторный ввод команд .....	27
Получение подсказки .....	28
Англо-русский словарь .....	29
Просмотр характеристик и состояния системы .....	30
Печать .....	30
Быстрая отмена печати .....	32
Рекомендуемые команды печати .....	32
Печать чистого текста .....	33

Просмотр заданий в очереди печати . . . . .	34
Удаление задания из очереди печати . . . . .	34
Работа с дискетами . . . . .	35
Просмотр содержимого дискеты . . . . .	36
Копирование файлов с дискеты . . . . .	37
Копирование файлов на дискету . . . . .	38
Очистка дискеты . . . . .	38
Согласование кодировок русского текста (перекодирование) .	39
Работа с флэш-дисками . . . . .	42
Работа с архивными файлами . . . . .	43
Файлы формата tar . . . . .	44
Файлы формата gz и tar.gz . . . . .	45
Файлы формата tgz . . . . .	45
Файлы формата spio . . . . .	46
Файлы формата zip . . . . .	46
Разбиение большого файла на части . . . . .	47
Сборка файла из частей . . . . .	47
Создание и редактирование текстовых файлов . . . . .	48
Редактор mcedit . . . . .	49
Редактор vi . . . . .	51
Программирование . . . . .	54
Языки, установленные в системе . . . . .	55
Создание файла с исходным текстом программы . . . . .	55
Трансляция программы . . . . .	56
Редактирование связей . . . . .	57
Запуск программы . . . . .	58
Автоматизация процесса разработки программы . . . . .	58
Отладка программы . . . . .	59
Особенности перехвата стандартного потока вывода командой tee . . . . .	61
Web программирование . . . . .	62
Расположение файлов в подкаталоге WWW . . . . .	63
Обращение к своим страницам Web . . . . .	63
Подключение базы данных SQL . . . . .	65
Создание документов в системе TEX . . . . .	66
Создание исходного файла документа . . . . .	66

Трансляция документа . . . . .	67
Просмотр документа . . . . .	68
Печать документа . . . . .	69
Вставка иллюстраций в документ . . . . .	69
Работа с графическими файлами . . . . .	70
Снятие копии экрана . . . . .	71
Программа xv . . . . .	73
Система ImageMagick . . . . .	73
Редактор GIMP . . . . .	74
Программа gv . . . . .	75
Обработка данных . . . . .	75
На досуге . . . . .	76
Письмо другу (mail) . . . . .	76
Коллективная игра . . . . .	79
Приложение . . . . .	80
Манипулирование файлами . . . . .	80
Работа с дискетами . . . . .	80
Перекодирование файлов . . . . .	80
Работа с файлами архивов . . . . .	81
Печать . . . . .	81
Работа с графикой . . . . .	81
Программирование . . . . .	81
Вёрстка на TEX . . . . .	82
Просмотр страниц Web . . . . .	82
Обработка информации . . . . .	82
Библиографический список . . . . .	83

## Учебное издание

*Александр Викторович Чернышов*

### РАБОТА ЗА X-ТЕРМИНАЛОМ

Под редакцией автора

Оригинал-макет выполнен в пакете  $\text{teTeX}$  с использованием кириллических шрифтов семейства LN.

Вёрстка в  $\text{T}_{\text{E}}\text{X}_{\text{e}}$ : А. В. Чернышов

По тематическому плану внутривузовских изданий учебной литературы на 2005 г., поз. 59

Лицензия ЛР № 020718 от 02.02.1998 г.

Лицензия ПД № 00326 от 14.02.2000 г.

---

Подписано к печати

Формат 60×88/16

Бумага 80 г/см<sup>2</sup> «Снегурочка»

Ризография

Объем 5,5 п. л.

Тираж 150 экз.

Зак. №

---

Издательство Московского государственного университета леса.  
141005. Мытищи-5, Московская обл., 1-я Институтская, 1, МГУЛ.

Телефон: (095) 588-57-62

e-mail: izdat@mgul.ac.ru