

Глава 12. Программы для работы с текстом

12.1. Несколько слов о форматах текстовых файлов

Как вы знаете, любой файл, в том числе и файлы, в которых сохранены текстовые документы того или иного вида, представляет собой просто последовательность байтов. Символы текста кодируются разными значениями байта или последовательностей байтов. Информация о том, как именно эти символы должны располагаться на странице, тоже кодируется с помощью неотображаемых управляющих символов, типа конца строки или символа табуляции. В простейших случаях число управляющих кодов ограничивается 32-мя первыми значениями байта (или кода ASCII), а все остальные значения байта используются для кодирования информационных символов. Именно такие файлы мы и называем ASCII-файлами. Примерами таких файлов могут служить файлы, создаваемые редакторами типа встроенного редактора программы Midnight Commander, файлы, создаваемые программой `notepad` в Windows и `vi` в UNIX.

Со временем появилось желание расширить возможности форматирования текста, а значит потребовалось увеличить число кодов, используемых в качестве управляющих, и в качестве таких кодов стали даже использовать последовательности байтов (символов ASCII). Но существенно то, что эти формирующие последовательности (почти) не мешают вам прочитать текст, содержащийся в файле, с помощью любого простейшего средства просмотра или простейшего текстового редактора. Примерами таких файлов могут служить файлы, создаваемые редакторами типа Лексикон, файлы в формате `html`.

Третий тип — это файлы, использующие собственный формат для представления текста (в которых символы текста тоже представлены специальными последовательностями). Существеннейшее отличие форматов третьего типа от двух предыдущих заключается в том, что и просматривать и создавать файлы в таких форматах без специальных программ практически невозможно. Например, HTML-файлы можно редактировать с помощью `Notepad`, но невозможно делать то же самое с файлами формата MS Word 97. Иногда трудно с первого взгляда отнести файл к тому или иному типу. Например, файлы формата Post Script формально относятся ко второму типу, поскольку весь читаемый текст там представлен в кодах ASCII, однако в этих файлах так много формирующих вставок, что текст можно найти лишь с большим трудом, почти как в файлах третьего типа.

Из вышесказанного следует, что даже для просмотра некоторых типов текстовых файлов (не говоря уж об их редактировании) требуются специальные программные средства. Часто для просмотра файлов пользователь применяет привычный ему текстовый редактор. Но встречаются ситуации, когда информация представлена в незнакомом для этого редактора формате. Самая большая проблема приверженцев Linux — форматы, используемые в продуктах Microsoft. Пока большинство пользователей ПК создают тексты в MS Word, приходится либо изыскивать текстовый редактор, который понимает форматы Word, либо находить программы-переводчики из формата Word в один из открытых стандартных форматов. Впрочем, даже если информация представлена в "простом" коде ASCII, вы, просматривая какой-либо файл, можете столкнуться с "нечитаемым" текстом из-за различия используемых кодировок русского языка. Поэтому сначала давайте рассмотрим вопрос о том, как прочитать (или просмотреть) файлы различных форматов или в различных кодировках.

12.2. Программы для просмотра текстов в разных форматах

Где-то я читал, что в традициях UNIX было создавать отдельную команду для каждого элементарного действия. Это наблюдение хорошо иллюстрируется наличием в Linux целой совокупности отдельных программ для просмотра файлов. Конечно, если вы работаете в каком-либо файловом менеджере, типа Midnight Commander или Konqueror, то вы будете использовать встроенные в них средства просмотра файлов. Но в некоторых случаях, может оказаться полезным и знание того, как просмотреть содержимое того или иного файла, работая просто в терминале.

12.2.1 Традиционные средства UNIX для просмотра текстовых файлов

Самым простым средством просмотра файла является, наверное, команда `cat`. Выведя содержимое текущего каталога с помощью команды `ls`, вы можете также вывести на экран содержимое любого из имеющихся файлов командой `cat`. Правда возникает одно неудобство: если файл большой, то в результате на экране остаются только последние его строки, все остальное "убегает вверх". Можно, конечно, пролистать несколько экранов с помощью клавиш `<Shift>+<PgUp>`, но возможность эта тоже ограничена (некоторой величиной, задаваемой в окружении, по умолчанию — величиной в 1000 строк).

Для организации постраничного вывода существует команда-фильтр `more`. Её можно применять в двух эквивалентных вариантах:

```
[user]$ cat file.txt | more
```

или

```
[user]$ more file.txt
```

Команда `less` представляет собой улучшенный и доработанный вариант команды `more`, который рекомендуется использовать вместо `more` во всех случаях. Имейте в виду, что команда `less` используется для вывода на экран страниц интерактивной подсказки `man`.

Для просмотра только нескольких последних строк текстового файла существует специальная команда `tail`, которой в качестве параметра можно указать количество выводимых строк. Можно предположить, что наличие такой команды было очень полезно в те времена, когда текстовые файлы создавались с помощью программы `cat`, путем прямого ввода с терминала, таким вот примерно образом:

```
[user]$ cat >> file.txt
```

В этом случае после любого перерыва в работе просто необходимо было просмотреть последние из введенных строк, чтобы вспомнить, на чем была остановлена работа.

Однако те времена давно уже прошли. Для создания текстов стали использовать текстовые редакторы, а для сохранения — не простые ASCII-файлы, а специальные, достаточно изощренные, форматы, позволяющие сохранить не только сам текст, но и информацию об абзацах, страницах, стилях, шрифтах и много что еще. И для просмотра таких файлов стали требоваться специальные программы (на сленге компьютерщиков — *вьюеры*), которые не выводят на экран все специальные символы форматирования, а преобразуют их в соответствующие отступы, выступления, пробелы, шрифты и т. д. Рассмотрим такие просмотрщики для двух распространенных форматов файлов.

12.2.2 Программа Acrobat Reader (версия 4.05)

Программа Acrobat Reader предназначена для просмотра файлов формата Portable Document Format (pdf), который широко распространен в компьютерном мире и используется в разных операционных системах и на разных платформах. В этот формат легко преобразуются документы формата PostScript. Файлы формата pdf очень часто встречаются в Сети. Не обязательно иметь возможность создавать такие файлы (программы их создания распространяются на коммерческой основе), но уметь их читать надо. Как раз для этого и служит программа Acrobat Reader фирмы Adobe Systems Incorporated, распространяемая бесплатно как в версии для Windows, так и в версиях для UNIX, и в частности для Linux. В настоящее время версия 4.05 этой программы для Linux может быть бесплатно получена на сайте фирмы <http://www.adobe.com> (для Windows уже выложена версия 5.0).

Для установки программы Acrobat Reader 4.05 вам потребуется 12 Мбайт дискового пространства, а для работы с программой — 32 Мбайт ОЗУ. Если у вас была установлена версия 3.0 этой программы, ее необходимо предварительно удалить.

Прежде чем начать установку, выберите один из двух возможных вариантов установки: сетевой или локальный (на отдельном компьютере).

Локальная инсталляция гарантирует вам доступ к программе на данном компьютере и может обеспечить значительно большую производительность (особенно в сравнении с работой по загруженной сети). Для локальной инсталляции Вы, естественно, должны иметь право записи в тот каталог, куда будет устанавливаться программа.

Сетевая инсталляция облегчает решение задач администрирования и поддержки программы, особенно в тех случаях, когда программу используют многие пользователи. Очевидно, что легче осуществлять обновление одного экземпляра программы на сервере, чем обновлять множество копий на отдельных компьютерах.

Сетевая инсталляция должна выполняться администратором сервера.

Однако, сетевая инсталляция имеет и некоторые недостатки. Сервер должен быть высоконадежным и устанавливаемые файлы должны быть установлены в каталог, который должен монтироваться одинаковым образом (в одну и ту же точку монтирования) на всех рабочих станциях сети.

По умолчанию программа Acrobat Reader устанавливается в каталог `/usr/local/Acrobat4` (проверьте с помощью команды `df`, что на соответствующем разделе диска достаточно свободного места).

Если вы получили файл с программой Acrobat Reader с Web-сайта, требуется вначале разархивировать установочные файлы с помощью программ `gunzip` и `tar`. После этого вам необходимо перейти в каталог, содержащий установочный скрипт и запустить его:

```
[root]# ./INSTALL
```

Инсталляционный скрипт выводит на экран несколько вопросов, в частности о том, в какой каталог разместить файлы программы. На каждый вопрос предлагается вариант ответа, который можно принять, просто нажав клавишу `<Enter>`. Если предлагаемый по-умолчанию ответ вас не устраивает, введите собственный вариант ответа. В остальном надо просто следовать инструкциям, выдаваемым программой установки. Если вы где-то ошиблись, можно прервать установку комбинацией `<Ctrl>+<C>` и начать все заново.

В конце концов, появляется сообщение о том, что установка успешно завершена (естественно, по-английски, что-то вроде "installed successfully").

После установки в том каталоге, куда вы установили программу (напомню, что по-умолчанию это /usr/local/Acrobat4), создаются несколько подкаталогов:

- каталог bin содержит скрипт для запуска программы,
- каталог Reader содержит файлы подсказки и некоторые платформенно-зависимые файлы (смотри примечание ниже),
- каталог Fonts содержит шрифты,
- каталог Browsers содержит скрипт для запуска Acrobat Reader из браузера Netscape.

Примечание

Если вы выбрали сетевой вариант инсталляции, то в каталоге Reader будут установлены несколько вариантов бинарных файлов, и скрипт запуска программы будет выбирать тот вариант бинарного файла, который нужен для рабочей станции, с которой производится вызов программы!

Наконец, в завершение инсталляции, вы можете сделать так, чтобы не было необходимости каждый раз для запуска программы Acrobat Reader сначала переходить в каталог, где расположен скрипт запуска. Для этого надо создать ссылку на этот скрипт в одном из каталогов, указанных в переменной \$PATH. Для примера создадим такую ссылку в каталоге /usr/bin:

```
[root]# ln -s /usr/locale/Acrobat4/bin/acroread /usr/bin/acroread
```

Для того, чтобы было удобно запускать программу в графической среде, остается только создать значок ("иконку") на рабочем столе (если вы не помните, как это сделать, смотрите описание используемой Вами графической среды) и на этом процесс инсталляции программы можно считать завершенным.

Теперь вы можете запустить Acrobat Reader (естественно, для этого должен быть запущен графический режим), используя команду **acroread** или щелчком по иконке на рабочем столе.

Если вы не создали ссылку, то надо указывать в командной строке полный путь к скрипту запуска:

```
[user]$ /usr/local/Acrobat4/bin/acroread
```

Для того, чтобы просмотреть с помощью Acrobat Reader какой-то конкретный файл или даже несколько, можно сразу указать имена этих файлов в командной строке. Например,

```
[user]$ acroread /user/share/docs/Samba/Collection.pdf
```

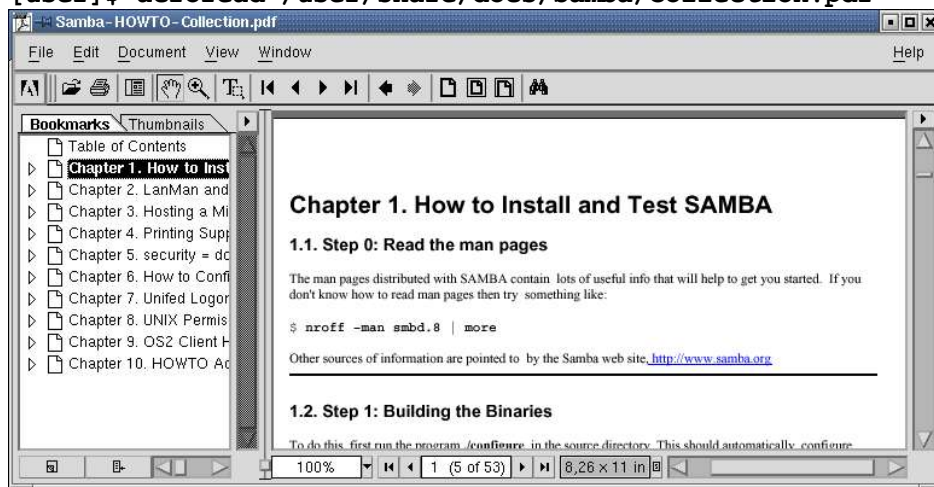


Рис. 12.1. Окно программы Acrobat Reader

Если вы введете команду **acroread <filename>**, когда Reader уже запущен, активная копия Reader отобразит заданный файл.

Если вы хотите получить подсказку по использованию программы, введите одну из команд

```
[user]$ acroread -help
```

```
[user]$ acroread -helpall
```

С помощью программы Acrobat Reader вы можете преобразовать файлы формата PDF в файлы формата PostScript уровней 1 или 2 при условии, что вы делаете это в командной строке графического режима. Для этого надо выполнить примерно следующую команду:

```
[user]$ cat sample.pdf | acroread -toPostScript > sample.ps
```

С помощью аналогичных команд (более подробное описание которых вы сможете найти в документации к программе) можно преобразовывать сразу группу файлов.

Для обратного преобразования PDF-файла в файл формата PostScript уровня 2, надо набрать следующую команду:

```
[user]$ acroread -toPostScript -level2 pdf_file_1
```

Я не буду подробно описывать все меню и возможности программы Acrobat Reader, поскольку она широко распространена и многим читателям знакома. Лучше остановимся подробнее на специфических средствах просмотра файлов Linux.

12.2.3 Программа gv

Программа `gv` (или `ghostview`) разработана Иоганнесом Плассом (Johannes Plass) и предназначена для просмотра файлов формата PostScript и PDF (рис. 12.2).

После ее запуска без указания имени файла основное окно программы будет пустым. Чтобы открыть какой-то файл, надо щелкнуть по клавише **Open** в левой колонке. Появится окно выбора файла (рис. 12.3), с помощью которого можно перемещаться по структуре каталогов и выбрать нужный файл. После этого имя файла появится в специальном поле в верхней рамке основного окна программы. Там же отображается несколько кнопок управления программой, которые образуют своеобразное меню. Только чтобы пользоваться этим меню, по кнопкам надо не просто щелкнуть, а удерживать левую кнопку мыши нажатой. С помощью кнопки **File** вы получаете доступ к пунктам **Open** (то же самое, что и упомянутая выше кнопка **Open** в левом вертикальном столбце), **Reopen** (Перечитать), **Print document** (Отпечатать документ), **Save document** (Сохранить документ).

Кнопка **State** (Состояние) позволяет включить или изменить некоторые параметры программы, в частности, получить доступ к отдельному окну настроек программы, изображенному на рис. 12.2.

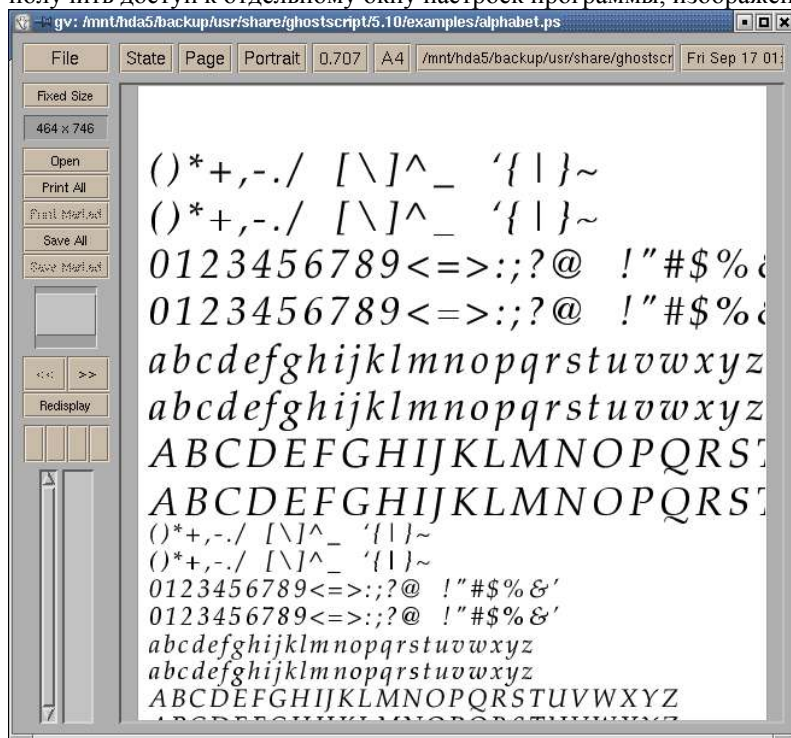


Рис. 12.2. Окно программы `gv`

Кнопка **Page** служит для перехода к следующей или предыдущей странице документа. Такие же переходы можно осуществить при помощи двух небольших кнопок, обозначенных символами `<<` и `>>` в левой вертикальной колонке. Рядом с кнопкой **Page** расположена кнопка, не имеющая постоянного названия, потому что она служит для задания одной из 4 возможных ориентаций вывода текста. Текст можно выводить на страницу, ориентированную обычным образом (Portrait), с поворотом на 90 градусов (Landscape), на 180 градусов (Upside-Down, т. е. вниз головой), на 270 градусов (Seascape).

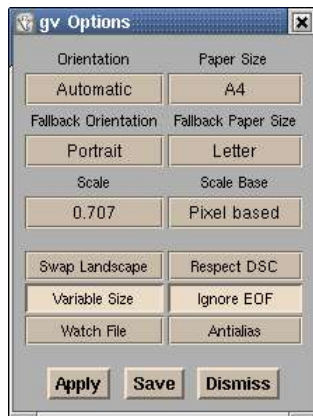


Рис. 12.3. Окно выбора файла для просмотра в gv

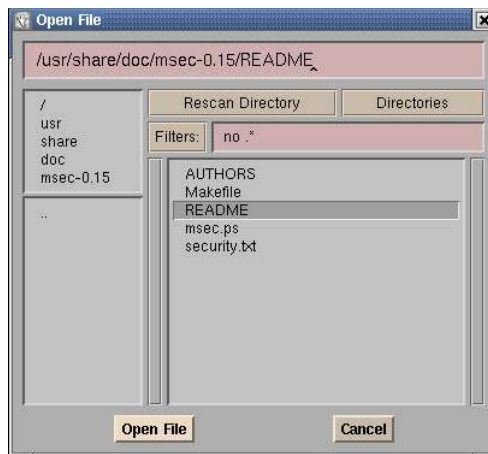


Рис. 12.4. Окно настроек программы gv

Следующая кнопка служит для задания масштаба изображения, который и указывается на этой кнопке. Еще одна кнопка служит для выбора формата бумаги. Это нужно для организации печати, ибо программа gv и есть средство для предварительного просмотра того, что получится на бумаге.

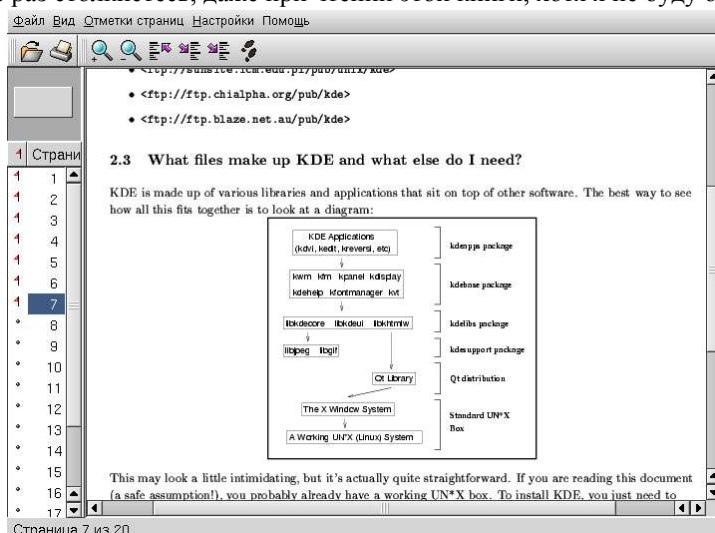
В левом столбце имеется еще одна интересная вещь: серый прямоугольник над кнопками >> и <<. Внутри этого прямоугольника имеется еще один прямоугольник, который можно захватить мышкой и подвигать в пределах внешнего прямоугольника. Прodelайте это и вы поймете назначение этого элемента: выводить в окно просмотра различные части страницы.

12.2.4 Просмотрщики файлов PS, PDF и DVI из KDE

Пользовательский интерфейс программы gv несколько непривычен для человека, работавшего только с Windows. Между тем, в составе интегрированной графической среды KDE имеются две программы с более традиционным обличком. Это "Просмотрщик PS/PDF" и "Просмотрщик DVI" (так эти программы именуются в заголовках собственных окошек и в меню KDE). Интерфейс у них очень похож и объясняется это тем, что они работают через одну оболочку — KviewShell. Однако в меню KDE они значатся как отдельные программы, и в заголовке окна каждая из них выводит собственное название.

Замечание:

Здесь можно попутно отметить, что в UNIX очень часто применяется такой прием: создается "рабочая лошадка" (back-end), к которой затем пишут красивые оболочки (front-end) на разные случаи, например, для текстового и графического режимов или для разных вариантов библиотек. С примерами такого подхода вы еще не раз столкнетесь, даже при чтении этой книги, хотя я не буду отдельно выделять эти моменты.



На рис. 12.5 приведен внешний вид окна программы "Просмотрщик PS/PDF" (разработчики Wilco Greven, David Sweet, Mark Donohoe, David Faure, Daniel Duley и Espen Sand.), которая иначе называется Kghostview, что, по-видимому, указывает на ее происхождение от программы, рассмотренной в предыдущем подразделе. О родстве с предыдущей программой говорит также то, что практически все

пункты меню повторяют аналогичные пункты `gv`, только меню организовано несколько иначе. Это видно и из рисунка. Более подробно рассказывать о программе не имеет смысла, достаточно того, что вы знаете о ее существовании. А пользоваться ею вы легко научитесь самостоятельно.

Рис. 12.5. Внешний вид окна программы "Просмотрщик PS/PDF".

О программе "Просмотрщик DVI" я тоже не буду рассказывать, ограничившись уже приведенным упоминанием о ней и сообщением о том, что она предназначена для просмотра файлов формата DVI, создаваемых системой верстки `TEX`.

12.2.5 Пакет WordViewer

Как уже говорилось выше, одна из самых больших проблем для пользователей Linux — это работа с файлами в форматах MS Word (и других программ из пакета MS Office). Ведь пока что большинство текстов создается именно в этом формате. Наиболее распространенное средство просмотра таких файлов, — это программа (точнее библиотека программ) `wv`, которая раньше носила более полное название `mswordview`. Библиотека `wv` предназначена для получения доступа к файлам форматов MS Word 2000, 97, 95 и 6 из операционных систем типа UNIX, в частности из Linux. В состав дистрибутива ALT Linux Junior 1.0 включена версия 0.6.5 этого пакета, авторами которого являются Dom Lachowicz и Caolan McNamara (первый разработчик). Если у вас этот пакет не установлен, вы можете найти его на сайте <http://www.wvware.com>.

Идея, реализованная в этом пакете, очень проста: раз мы не имеем средств для просмотра файлов в формате MS Word, то давайте преобразуем текст из этих файлов в какой-то открытый формат. В качестве последнего можно выбрать один из следующих форматов: HTML, PS, PDF, LaTeX, DVI (формат издательской системы `TEX`), ABW (формат текстового редактора AbiWord), Wml (формат, используемый в персональных органайзерах PDA и устройствах типа Web-телефонов), ASCII-текст. Вызов отдельных библиотечных процедур может быть использован в других приложениях. Разработчики обещают, что вскоре станет возможным и обратное преобразование: из перечисленных открытых форматов — в формат MS Word. Пакет состоит из отдельных программ, каждая из которых предназначена для преобразования doc-файла в определенный формат. Этот формат указывается непосредственно в названиях отдельных программ пакета: `wvAbw`, `wvCleanLatex`, `wvDVI`, `wvHtml`, `wvLatex`, `wvMime`, `wvPDF`, `wvPS`, `wvRTF`, `wvSimpleCLX`, `wvText`, `wvWml`. Кроме того, в состав пакета входят две вспомогательных утилиты: `wvVersion`, которая служит для получения информации о версии документа MS Word, и `wvSummary` — эта утилита выводит общую информацию о документе, которую в самом MS Word можно ввести через пункт меню **Файл | Свойства**. Вот как выглядит вывод этих команд (для использованного мной в этом примере файла служебные данные не были введены; впрочем, если их ввести по-русски, то радости будет не много больше, поскольку вывод получим в кодовой странице CP-1251, так что текст на экране будет нечитаемым).

```
[user@linux tmp]$ wvVersion book-p1.doc
Version: word8, Encrypted: No
```

```
[user@linux tmp]$ wvSummary book-p1.doc
```

```
The title is Å
The subject is
The author is kos
The keywords are
no comments found
The template was Normal.dot
The last author was kos
The rev # was 12
The app name was Microsoft Word 8.0
PageCount is 1
WordCount is 52757
CharCount is 300716
Security is 0
Codepage is 0x4e3 (1251)
```

Основной утилитой пакета является программа `wvWare` (или `wvConvert`), вызов которой осуществляется следующим образом

```
wvWare [OPTION...] filename.doc > filename.html
```

Основные опции:

- `-x --config=config.xml`
Указывает на используемый выходной фильтр.
- `-c --charset=charset`
Задаёт кодовую страницу для `iconv`.

- `-p --password=password`
Задает пароль для зашифрованных документов Word.
- `-d --dir=dir`
Задает каталог, в котором будет сохранена создаваемая графика.
- `-v -version`
Выдает версию пакета `wvWare`.
- `-? -help`
Выводит краткую справку по использованию программы.

Выходной фильтр, указываемый опцией `-x` или `--config=`, задает формат выходного файла. Если эта опция не задана, то выходной фильтр ищется в текущем каталоге или (если в текущем не нашли) по месту установки пакета. По умолчанию используется фильтр `wvHtml.xml`, т. е. `doc`-файл преобразуется в формат HTML.

После преобразования полученный файл можно просмотреть с помощью соответствующей программы просмотра, например, `html`-файл — с помощью любого Web-браузера. Конечно, при этом некоторые возможности форматирования, имеющиеся в Word, теряются, и это надо иметь в виду, просматривая полученные файлы.

Насколько я могу судить, именно программы пакета `wv` используются для открытия файлов MS Word в некоторых текстовых редакторах для Linux, например, в `AbiWord`.

12.2.6 Программы-перекодировщики кодовых страниц

Как известно, для представления символов русского алфавита существует несколько альтернативных кодировок или кодовых страниц.

В Unix-системах наиболее распространенной является кодировка `KOI8-R`, Microsoft использует `CP-1251` или `CP-866` (DOS). Существуют еще `ISO8859-5`, `UTF`, `Unicode` (подробнее о кодировках можно прочитать в *разд. 9.3*).

Если вы встретили файл, содержимое которого "не читается", то вам потребуется программа перекодировки. Как говорится в `RU.LINUX.FAQ`: "Перекодировщиков `CP1251` и `CP866` --> `KOI8-R` просто огромное количество. Не надо писать новых ;-). Наиболее широко распространены `iconv` (входит в поставку `glibc`) и `GNU recode`."

Программа `iconv` запускается следующим образом:

```
[user]$ iconv -f866 -tKOI8-R -o <outfile> infile
```

Если не указать выходной файл (опция `-o`), то результат будет выдаваться на экран (используя фильтры `more` или `less` можно удобно просмотреть файл). Чтобы получить список всех возможных кодировок (а он огромен!), дайте команду `iconv --list`, а для получения помощи: `iconv -?` или `iconv --usage`. Между прочим, `man`-страница не выдается. Впрочем, других опций все равно очень немного: только `--verbose` (сообщать дополнительные сведения), `-V`, и `--version` (обе опции служат для вывода версии программы).

Программу `recode` можно найти на <http://www.iro.umontreal.ca/~pinard/recode/>.

Запускается она примерно так:

```
[user]$ recode CP1251..KOI8-R winfile.txt
```

Кроме упомянутых команд вы можете воспользоваться программой "Russian Anywhere", которая существует как в версии для Windows (где я с ней и познакомился), так и в версии для командной строки Linux (создатели обещают выпустить и графическую оболочку). Эту программу можно скачать (как в исходных кодах, так и в виде исполняемого модуля) с сайта разработчика <http://www.livotov.org/software/>.

Исполняемый модуль программы имеет название `re`. Его лучше поместить в один из каталогов, указанных в переменной `PATH`.

Программа вызывается из командной строки. Для того, чтобы перекодировать какой-то файл, который "не читается", в кодировку `KOI8-R`, надо дать команду:

```
[user]$ re <SourceFile> <DestFile> ? K
```

где :

- `<SourceFile>` — исходный (не читаемый) файл;
- `<DestFile>` — перекодированный файл;
- `?` — сообщает RE, что кодировка исходного файла не известна и `re` должна проанализировать файл и самостоятельно определить его кодировку;
- `K` — задает кодировку для результирующего файла (в данном случае `KOI-8`).

Если вы знаете кодировку исходного файла, вы можете указать ее вместо символа `"?"`. Например, если вы хотите перекодировать файл `letter.txt`, который был создан в Windows, и вы знаете, что файл сохранен в кодировке `1251`, то надо дать команду:

```
[user]$ re letter.txt letter-koi.txt W K
```

После этого, просмотрев файл letter-koi.txt, вы увидите вполне читаемый русский текст в KOI8-R.

Полный формат вызова перекодировщика:

```
[user]$ re options filename_from filename_to cp_from cp_to [s/d/f] [u/l/s]
```

где options: [-v] [-E|-R|-N] [-e|-s]

- -v — выдавать информацию о ходе обработки;
- -n — не выдавать информацию о ходе обработки (задано по умолчанию);
- -E — преобразовывать все символы p, H из русских в английские;
- -R — преобразовывать все символы p, H из английских в русские;
- -N — оставлять все p, H (русские и английские) как в исходном тексте (задано по умолчанию);
- -e — перекодировать все символы 0x80 — 0xFF;
- -s — перекодировать только 64 символа русского алфавита (задано по умолчанию),

a cp_from и cp_to — любой из следующих символов, обозначающих возможные кодировки (по умолчанию — W,K).

Таблица 12.1. Обозначение кодировок в программе re.

Символ	Кодировка	Символ	Кодировка
W	Windows	_ (подчеркивание)	_xhe
D	Dos	%	%hex
K	KOI-8	\\	\\hex
L	Latin	G	Graph_win
I	Iso	<	binhex
H	HEX	+	+UTF7-
S	ShiftKbrd	C	C_MIC
M	Mac	Y	Y_c16
A	AFF	Z	Z_c32
O	Odd(UTF8_1)	F	F(UTF8_2)
B	Base64	P	Pict
E	Express	N	N_Estl
T	T-Html	V	V_Vpp855
U	User	X	X_sp
- (тире)	uue	J	J_diff

Как уже было сказано, если cp-from="?", то программа пытается самостоятельно определить кодировку исходного файла.

Если у вас по каким-либо причинам не оказалось ни одной из указанных программ-перекодировщиков, то для просмотра содержимого файла можно воспользоваться одним из браузеров Интернет, которые изначально ориентированы на работу с разными кодировками. Например, сгодится обычный lynx:

```
[user]$ lynx -assume_local_charset cp866 file.txt
```

Можно также загрузить "нечитаемый" файл в Netscape Navigator, после чего поменять кодировку через меню View / Character Set.

12.3. Проверка правописания

Прежде, чем перейти к рассмотрению текстовых редакторов под Linux, необходимо кратко рассмотреть программу проверки правописания ispell. Дело в том, что проверка правописания — это одна из функций, которую должен иметь современный текстовый редактор, и многие из них подключают для выполнения этой функции именно ispell (или aspell). Существует русифицированный вариант этой программы, которой был разработан Владимиром Рогановым и Константином Книжником.

Установка ispell состоит из двух этапов: вначале надо установить саму программу, а затем установить словарь русского языка. Для установки самой программы ispell я воспользовался пакетом ispell-3.1.20-23.i386.rpm, а для его русификации — пакетом ispell-russian-3.1.20-23.i386.rpm. Оба пакета входили в состав дистрибутива Black Cat Linux 6.02. Для установки первого пакета достаточно дать команду

```
[root]# rpm -i ispell-3.1.20-23.i386.rpm
```

а для второго — команду

```
[root]# rpm -i ispell-russian-3.1.20-23.i386.rpm.
```

После этого в каталоге /usr/lib/ispell появятся файлы русского словаря russian.aff и russian.hash (остальные словари, например, немецкий) можно удалить, если вы не собираетесь производить проверку правописания на этих языках.

Для проверки текста теперь достаточно дать команду следующего вида:

```
[user]$ ispell -drussian edit.htm
```

Естественно, имя файла edit.htm здесь взято для примера; вы должны подставить имя того файла, который вы хотите проверить, причем файл должен находиться в текущем каталоге, иначе надо указать полное имя файла с указанием пути.

```
оных      File: edit.htm

в большинстве книг, посвященных UNIX, я не буду приводить здесь его описания.
<BR>Наибольшую известность среди UNIX-оных редакторов имеют <B>TeX </B>и<B>

00: ивых
01: ловых
02: новых
03: оных
04: овях
05: овух
06: о вых
07: о-вых
08: ровых
09: совых
10: свых
11: твых
12: вых

[SP] <number> R)epl A)ccept I)nsert L)ookup U)ncap Q)uit e(X)it or ? for help
```

Рис. 12.6. Проверка правописания с помощью ispell

Принцип работы программы ispell очень прост: каждое встречающееся в файле слово должно иметься в словаре программы. Если слово в словаре не найдено, считается, что найдена ошибка, и на экран выводится сообщение, пример которого можно увидеть на рисунке 12.5. В самой верхней строке выведено обнаруженное ошибочное слово и имя проверяемого файла. Ниже выведено несколько строк (число можно задать) из этого файла, содержащих обнаруженную ошибку. Если в словаре обнаружены слова, похожие на ошибочное, то они выводятся ниже (с порядковыми номерами). Далее следует строка подсказки и командная строка программы.

В командной строке в нижней части можно использовать следующие клавиатурные команды.

Таблица 12.2. Команды замены в программе ispell.

Команда	Выполняемое действие
<R>	Заменить ошибочное слово (программа предложит набрать правильное слово в нижней строке экрана)
<пробел>	Пропустить данное вхождение слова
<A>	Пропустить все вхождения данного слова в текущей сессии работы с программой
<I>	Пропустить это слово и включить его в персональный словарь (который хранится в файле .ispell_russian в домашнем каталоге пользователя)
<U>	То же самое, только слово записывается в нижнем регистре (маленькими буквами)
<Q>	Немедленный выход из программы (вначале запрашивается подтверждение, а проверяемый файл остается не измененным; сделанные замены не проводятся)
<X>	Прервать проверку, записать проведенные изменения и выйти из программы
<!>	Временный выход в оболочку shell

Если в качестве команды ввести порядковый номер одного из предложенных программой вариантов замены, то программа заменит ошибочное слово на слово, соответствующее набранному порядковому номеру варианта замены. Только номера надо вводить в точности так, как они предлагаются программой, т. е. с предшествующими значащим цифрам нулями (если они есть). И набирать номер надо, не вводя предварительно команду <R>, иначе ошибочное слово будет заменено просто на соответствующую цифру.

Программа `ispell`, как уже упоминалось, используется в качестве модуля проверки правописания во многих текстовых редакторах, например, в Emacs она обеспечивает проверку правописания непосредственно в процессе подготовки текста.

Если подумать о принципе проверки, заложенном в программу, легко понять, что с ее помощью можно проверить только очень ограниченный класс ошибок, а именно, орфографические ошибки, состоящие в неправильном написании слов. Очевидно, что не будут обнаружены никакие ошибки в грамматических конструкциях, согласовании слов и т. д.

Еще один недостаток программы, с которым я столкнулся, проявляется в тех случаях, когда на текущем диске мало свободного места, меньше, чем необходимо для записи исправленного файла. Программа в таком случае записывает только ту часть файла, которая поместилась, и теряет все остальное. Никаких предупреждений при этом не выдается.

Если недостатка дискового пространства нет, то после внесения исправлений программа записывает исправленную версию файла, а исходный файл сохраняет, добавив к его имени расширение `.bak`.

12.4. О трех типах текстовых редакторов

Редактирование текстовых файлов (с текстами на естественном языке, либо с текстами программ) — одна из наиболее часто выполняемых работ на любом компьютере и в любой операционной системе. Может быть, поэтому для Linux разработано уже очень много текстовых редакторов (на www.linuxlinks.com перечислены около 100 наименований, и это еще, вероятно, не все). Так что выбрать есть из чего. И стоит уделить некоторое время оптимальному выбору редактора.

Конечно, чтобы такой выбор был обоснован, в идеале надо опробовать все редакторы или большинство из них. Это, очевидно, невозможно, так что приходится положиться либо на случай, либо на мнение кого-то из знакомых или авторов компьютерных книг. (Кстати, неплохой, на мой взгляд, обзор текстовых редакторов содержится в книге А.Федорчука "Офис, графика, Web в Linux" [П1.6]. Я тоже попробую изложить свои впечатления о некоторых текстовых редакторах для Linux.)

Из всего множества различных текстовых редакторов рядовой пользователь обычно выбирает два-три, с которыми постоянно работает. Он заучивает до автоматизма управляющие комбинации клавиш, привыкает к определенному образу, через пункты меню или щелчки мышкой, выполнять стандартные операции редактирования и, вообще, приноравливается к среде редактора. Поэтому для смены редактора должны быть достаточно веские причины.

По моему мнению, рядовому пользователю, часто использующему компьютер для редактирования файлов, необходимо освоить по крайней мере 3 редактора.

Один из них — это мощный текстовый процессор, работающий в режиме WYSIWYG, обеспечивающий широкие возможности форматирования текста и массу дополнительных возможностей, отсутствующих в более простых редакторах. К этому типу я бы отнес текстовые редакторы (процессоры) из пакетов StarOffice, Applixware, KOffice, отдельные текстовые процессоры Maxwell и WordPerfect 8, AbiWord, а также издательскую систему T_EX. Все редакторы этого типа я буду называть текстовыми процессорами. Правда, некоторые авторы, например, тот же А.Федорчук, все же делят их на просто редакторы и процессоры, однако я не вижу оснований для этого. Собственно говоря, весь вопрос в возможностях форматирования текста, которые предоставляет редактор. И какой-либо точный критерий для деления редакторов этого класса на два подкласса предложить трудно, если вообще возможно.

Второй необходимый редактор — это редактор для создания или правки ASCII файлов, работающий в графическом режиме. С помощью этого редактора Web-мастер может, например, редактировать html-странички, в нем можно написать письмо для последующей отправки по e-mail и т. д. Это должен быть редактор графического режима, потому что во многих случаях в графическом режиме работать легче и удобнее, чем в текстовом. Примерами таких редакторов являются KEdit и KWrite из KDE, nedit.

И все же надо уметь пользоваться и одним из консольных текстовых редакторов, потому что Вы, как единственный пользователь (и даже суперпользователь) персонального компьютера, должны уметь отредактировать конфигурационные файлы, причем в любой ситуации, даже тогда, когда графический режим не загружается. Выбор редакторов этого типа очень широк: vi, vim, bvi, Nvi, Elvis, Levee, vile, Wily, joe, aee, Fred, gred, le, lpe, Zed, Emacs, CoolEdit. Давние приверженцы UNIX чаще всего используют vi или его усовершенствованную версию vim, но тем, кто переходит на Linux из среды Windows проще будет освоить CoolEdit, имеющий большое сходство со встроенными редакторами программ Norton Commander и FAR.

Исходя из этих рассуждений нижеследующее изложение разбито на три больших части, каждая из которых посвящена одному из выделенных типов редакторов. Первый вариант раздела с описанием каждого рассматриваемого ниже редактора был создан в том самом редакторе, который в этом разделе описан (эти первые варианты размещались мной на сайте <http://linux-ve.chat.ru>) [Примечание 22]. Я надеюсь, что после

чтения этих разделов вы сможете определиться с выбором текстовых редакторов. Конечно, критерии выбора могут у каждого оказаться свои. Но думаю, что для нас, русскоязычных пользователей, немаловажным фактором является возможность вводить и редактировать тексты на русском языке. Желательно также, чтобы пункты меню и сообщения программы тоже были русифицированы (правда, до некоторой степени с английскими терминами здесь можно мириться, особенно если есть хорошее описание программы на русском, потому что число пунктов меню обычно невелико и их смысл можно просто запомнить).

12.5. Консольные редакторы ASCII-файлов

Начнем с рассмотрения редакторов текстового режима, т. е. работающих в консоли. Говоря о таких редакторах просто нельзя не упомянуть о редакторах **vi** и **Emacs**, но более основательно будет рассмотрен встроенный редактор оболочки Midnight Commander – **CoolEdit**.

12.5.1 Редакторы типа vi

Редактор **vi** (или его несколько доработанные потомки) по умолчанию включается в любую UNIX-подобную систему, в том числе и во все дистрибутивы Linux. Все приверженцы UNIX, имеющие значительный стаж работы с этими ОС, знают и используют этот редактор. Описание редактора **vi** вы сможете найти если не в любой, то уж точно в большинстве книг, посвященных UNIX. У редактора **vi** имеется несколько потомков, которые в чем-то его улучшают и усовершенствуют. Это такие редакторы как **Vim**, **bvi**, **Nvi**, **Elvis**, **Levee**, **vile**, **Wily** (редактор **Vim** даже вызывается в Linux по команде **vi**). Краткий обзор редакторов этого класса вы можете найти в статье А. Фомичева "Текстовые редакторы для ОС UNIX" (Открытые системы, № 4, 1994 г.). Однако для тех пользователей, которые мигрировали в Linux из среды Windows, все эти редакторы вряд-ли покажутся удобными. Причина в том, что в этих редакторах нет привычных меню и подсказок (насколько можно судить по **Vim** версии 5.3) и надо запомнить множество клавиатурных комбинаций для ввода команд. При этом работа в редакторе организована в виде двух отдельных режимов — ввода текста и ввода команд, и новичок часто просто путает режимы. В общем без печатного руководства за освоение редакторов этого типа я бы вам браться не рекомендовал. И хотя те, кто освоил **vi**, часто прибегают к его помощи для редактировании простых файлов (особенно, конфигурационных файлов и скриптов), я считаю, что есть более удобные средства, а поэтому не рассматриваю редакторы типа **vi** (хотя не упомянуть о них не мог).

12.5.2 Редактор Emacs

Наибольшую известность среди редакторов, используемых в Linux, имеет **Emacs**. Он существует как в варианте для текстового режима, так и в варианте для графической оболочки. Некоторые даже говорят, что **Emacs** — это не редактор, а образ жизни, а в другом источнике его называют религией. Если вы хотите ближе познакомиться с **Emacs**, я могу рекомендовать вам недавно вышедший русский перевод книги Р. Столлмана о нем; поищите эту книгу, например, в виртуальном магазине "Болеро". Прекрасное вводное описание этого редактора вы найдете в книге А. Федорчука "Офис, графика, Web в Linux". Я, однако, считаю, что начинающим пользователям для редактирования ASCII-файлов целесообразно использовать **CoolEdit** — встроенный редактор файлового менеджера Midnight Commander.

12.5.3 CoolEdit — встроенный редактор программы Midnight Commander

CoolEdit — это простая в использовании программа с привычными для большинства пользователей (особенно для тех, кто работал с Norton Commander под DOS или с FAR под WINDOWS) комбинациями управляющих клавиш. Кроме того, надо учитывать, что обычно любая работа с файлом начинается с файлового менеджера, поскольку вначале нужно, как минимум, найти файл. Midnight Commander как раз и является таким файловым менеджером, причем переход к редактированию осуществляется простым нажатием клавиши <F4> после установки подсветки на имя найденного файла. Это мелочь, но удобно. Учитывая изложенные соображения, я начну подробный рассказ о текстовых редакторах именно с **CoolEdit**. Встроенный редактор Midnight Commander (рис. 12.7) вызывается во время работы в этой программе нажатием клавиши <F4> при условии, что в инициализационном файле Midnight Commander установлена в 1 опция "use_internal_edit". Его также можно вызвать независимо от Midnight Commander просто из командной строки, командой **mcedit**. Однако его истинное имя все же CoolEdit, в этом вы можете убедиться, если в режиме редактирования нажмете клавишу <F9> и откроете пункт меню **Файл | Об авторах**.

Рис. 12.7. Внешний вид окна редактора Cooledit

Этот редактор обеспечивает выполнение большинства функций редактирования, присущих полноэкранным редакторам текста. С его помощью можно редактировать файлы практически любого размера, поскольку

```
edit.htm [M=] 0 L:1362+13 375/12271 *114084/486226)= 0 213 d5H
(р) Мне кажется, что для начинающих пользователей (к которым я отношу и
себя), для редактирования ASCII-файлов целесообразно использовать встроенный
редактор файлового менеджера <a href="mc.htm">Midnight Commander</a>.
Это простая в использовании программа с привычными для большинства
пользователей (особенно для тех, кто работал с Norton Commander под DOS или с FAR
под WINDOWS) комбинациями управляющих клавиш. Другой довод – если Вы
используете какой-то другой редактор, Вы должны загрузить вначале редактор, а потом
еще искать и загружать в него нужный файл, в то время как в MC просто
достаточно нажать клавишу [F4], установив подсветку на имени уже найденного файла
(ведь мы чаще начинаем работу с файлового менеджера, то есть вначале находим
файл, который хотим редактировать). Это мелочь, но удобно.
<br>Учитывая изложенные соображения, я начну рассказ о текстовых
редакторах именно с этого редактора.
<a NAME="12.2.1."></a>
<h4>12.2.1. Встроенный редактор программы <a href="mc.htm">Midnight
Commander</a>.</h4>
Встроенный редактор из Midnight Commander вызывается во время работы в
этой программе нажатием клавиши [F4]
при условии, что в инициализационном файле Midnight Commander-а
установлена в 1 опция "use_internal_edit". Однако его можно вызвать независимо от
<b>Midnight Commander</b> просто из командной строки, командой
<b>mcedit</b>.
```

верхняя граница для размера редактируемого файла составляет 16 Мбайт. Обеспечивается редактирование исполняемых (двоичных) файлов без потери данных.

Поддерживаются следующие возможности:

- копирование, перемещение, удаление, вырезание и вставка блоков текста;
- отмена предыдущих операций (по комбинации клавиш <Ctrl>+<U>);
- выпадающие меню;
- вставка файлов;
- макро-определения;
- поиск и замена по регулярным выражениям (другими словами, по шаблонам, сформированным с использованием специальных символов), а также собственный вариант операций поиска и замены, основанный на функциях scanf-printf;
- выделение текста комбинацией клавиш <Shift>+<стрелки> в стиле MS Windows — MAC (только для linux-консоли);
- переключение между режимами вставки-замены символа.

Редактор очень прост и практически не требует обучения (тем более, что можно найти версии, в которых основная часть пунктов меню русифицирована, такая версия была включена, например, в дистрибутив Black Cat 5.2). Для того, чтобы узнать, какие клавиши вызывают выполнение определенных действий, достаточно просмотреть выпадающие меню, вызываемые нажатием клавиши <F9> в окне редактора (рис. 12.8).

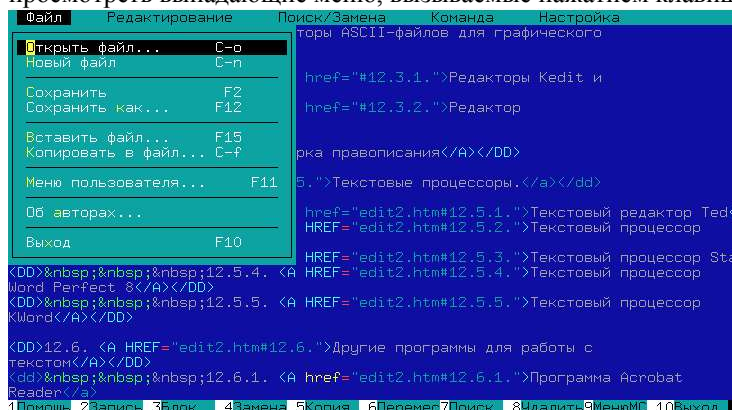


Рис. 12.8. Выход в меню Cooledit по клавише <F9>

Если вы работаете в консоли Linux, для работы с блоками текста можно использовать следующие комбинации клавиш:

- <Shift>+<клавиши стрелок> — выделение блока текста;
- <Ctrl>+<Ins> — копирует блок в файл cooledit.clip;
- <Shift>+<Ins> — производит вставку последнего скопированного в cooledit.clip блока в позицию курсора;
- <Shift>+ — удаляет выделенный блок текста, запоминая его в файле cooledit.clip.

Если у вас установлена программа **gpm** — драйвер мыши для консоли, то вы можете нажать на левую кнопку мыши в начале выделяемого блока, перенести курсор в конец блока и отпустить кнопку (тем самым выделить текст), а затем перенести курсор туда, куда надо вставить фрагмент, и нажать на правую кнопку мыши для вставки выделенного текста.

Редактор поддерживает макросы. Для того, чтобы определить макрос, нажмите <Ctrl>+<R>, после чего введите строки команд, которые должны быть выполнены. После завершения ввода команд снова нажмите <Ctrl>+<R> и свяжите макрос с какой-нибудь клавишей или комбинацией клавиш, нажав эту клавишу (комбинацию). Макрос будет вызываться нажатием <Ctrl>+<A> и назначенной для него клавиши. Макрос можно также вызвать нажатием любой из клавиш <Alt>, <Ctrl> или <Esc> и назначенной макросу клавиши, при условии, что данная комбинация не используется для вызова какой-либо другой функции.

Макро-команды после определения записываются в файл cedit/cooledit.macros в Вашем домашнем каталоге. Вы можете удалить макрос удалением соответствующей строки в этом файле.

При выполнении операций замены (по функциональной клавише <F4>) вы можете использовать функции поиска и замены scanf для поиска и замены в соответствии с шаблонами формата языка C. Вначале посмотрите map-страницы sscanf и sprintf, чтобы узнать, что такое шаблоны формата и как они работают. Приведем пример: предположим, вы хотите заменить все вхождения блоков текста, состоящих из открывающей скобки, трех разделенных запятыми чисел, и закрывающей скобки, на блок, состоящий из слова apples, третьего числа исходного блока, слова oranges и потом второго числа из исходного блока. Тогда в диалоговом окне, которое появится при вызове команды замены (<F4>), надо задать следующие шаблоны:

```
Enter search string
(%d,%d,%d)
Enter replace string
apples %d oranges %d
Enter replacement argument order
3,2
```

Последняя из этих строк говорит, что третье и второе число должны быть подставлены на места первого и второго аргументов. Рекомендуется все же при осуществлении замены пользоваться опцией "спрашивать подтверждение" ("Prompt on replace"), потому что программа считает совпадениями все случаи, когда число аргументов совпадает с заданным, хотя это не всегда означает полное совпадение. Scanf также не обращает внимания на количество символов пробела.

Встроенный редактор обрабатывает символы из второй половины кодовой таблицы (160+). Но когда редактируете бинарные файлы, лучше установить опцию **Биты символов** (Display bits) из меню **Настройки** в положение "7 бит", чтобы сохранить формат файла.

Для того, чтобы описать здесь все функции встроенного редактора, потребовалось бы слишком много места. Да в этом и нет нужды, поскольку для его использования достаточно запомнить, что все основные операции можно выполнить через пункты меню, которое вызывается нажатием клавиши <F9> в окне редактирования. Кроме того, можно прочитать map-страницу по команде man mcedit или info mcedit.

В завершение этой краткой справки по встроенному редактору программы Midnight Commander мне хочется рассказать о том, как осуществляется перенос фрагментов текста из одного файла в другой.

Если вы работаете в консоли, то эта задача решается через меню или с помощью следующих операций:

- отмечаем начало блока с помощью клавиши <F3>;
- перемещаем курсор к концу блока;
- отмечаем конец блока с помощью клавиши <F3>;
- набираем комбинацию <Ctrl>+<Insert>;
- закрываем этот файл, открываем другой;
- ставим курсор туда, куда хотим вставить данный фрагмент, и нажимаем комбинацию клавиш <Shift>+<Insert>.

Все, задача выполнена. Перенос фрагмента текста этим способом может быть произведен из одной виртуальной консоли в другую.

Но все это работает только в консоли. А при работе с тем же редактором в окне графической оболочки та же задача была для меня достаточно долгое время проблемой.

Я не сразу нашел способ ее решения (т. е. переноса фрагмента текста), который работает как в консоли, так и в окне графической оболочки. Этот способ состоит в переносе фрагмента текста через другой файл (по умолчанию используется файл ~/.cedit/cooledit.clip). Выделите фрагмент текста, выберите пункт меню **Файл | Копировать в файл** (File | Copy to file) и нажмите <Enter>. Затем переходите в другой файл, ставите курсор туда, куда надо вставить фрагмент, и выбираете пункт меню **Файл | Вставить файл** (File | Insert file).

12.6. Редакторы ASCII-файлов для графического режима

Очевидно, что было бы очень удобно, если бы редактирование ASCII-файлов в графическом режиме осуществлялось с помощью тех же редакторов, которые применяются в консольном режиме. Тогда не пришлось бы заучивать другие комбинации клавиш, менять привычную среду. Редактор CoolEdit, о котором было рассказано в предыдущем подразделе, может запускаться через эмулятор терминала и использоваться,

таким образом, в графическом режиме. Однако, при этом оказывается, что некоторые комбинации клавиш в эмуляторе не работают или работают не так, как в консоли. Кроме того, работа с мышкой в редакторах, изначально ориентированных на графический режим, организована гораздо лучше, чем в консольных редакторах, в силу чего повышается общее удобство работы (хотя, может быть, это чисто мое субъективное ощущение). Поэтому я все же рекомендую не ограничиваться использованием только консольного редактора для ASCII-файлов, а освоить один из многочисленных редакторов, ориентированных на обработку таких файлов в графическом режиме. Рассмотрим три подобных редактора.

12.6.1 Редактор KEdit

Редакторы KEdit и KWrite входят в состав графической среды KDE. Они предназначены для работы в графическом режиме, но работают с ASCII-файлами. Редакторы очень похожи, поэтому я расскажу вначале о KEdit, а затем просто укажу на отличия, имеющиеся во втором редакторе.

Редактор KEdit очень прост в использовании и, если вы вообще, когда-нибудь занимались редактированием текста, у вас не возникнет с ним проблем. К тому же, если KDE у вас уже русифицирован, то не возникнет никаких проблем с вводом и отображением кириллических символов. Более того, меню и сообщения программы тоже русифицированы.

Открыть файл для редактирования можно через меню или по комбинации `<Ctrl>+<O>`. При этом появляется окно, в котором можно выбрать файл для редактирования. Тут все привычно, и думаю, что дополнительных пояснений вам не потребуется. Кроме этого способа, работая в среде KDE, можно воспользоваться и методом "Drag and Drop". Это значит, что из файлового менеджера Konqueror вы можете прихватить файл мышкой и просто "бросить" его в окно KEdit.

Основные операции редактирования осуществляются с помощью клавиатуры. По клавише `<Insert>` происходит переключение между режимами вставки и замены символов. Перемещение по тексту осуществляется с помощью клавиш-стрелок и клавиш `<Page Down>`, `<Page Up>`, `<Home>` (переход в начало строки), `<End>` (переход в конец строки). Выделить блок текста (к сожалению, только линейный) можно с помощью мыши или клавишами стрелок при нажатой клавише `<Shift>`. Клавиша `<Delete>` удаляет символ справа от курсора или выделенный блок текста.

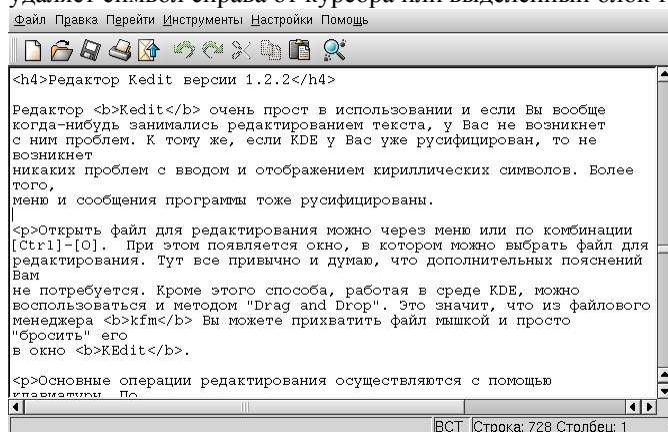


Рис. 12.9. Окно редактора KEdit.

Во многих редакторах, и в том числе в KEdit, операции перемещения по тексту и удаления, можно выполнять также с помощью "горячих" комбинаций клавиш (возможно, в связи с тем, что на старых терминалах не было клавиш стрелок?):

- `<Ctrl>+<A>` — переместить курсор в начало строки.
- `<Ctrl>+` — переместить курсор на один символ вправо.
- `<Ctrl>+<E>` — переместить курсор в конец строки.
- `<Ctrl>+<N>` — переместить курсор на одну строку вниз.
- `<Ctrl>+<P>` — переместить курсор на одну строку вверх.
- `<Ctrl>+<D>` — удалить символ справа от курсора.
- `<Ctrl>+<H>` — удалить символ слева от курсора.

Основные команды по переносу блоков текста из одного места в другое осуществляются с помощью тех же комбинаций, которые используются в редакторе из Midnight Commander и многих других редакторах, так что пальцы находят эти комбинации уже автоматически:

- `<Ctrl>+<C>` — скопировать выделенный текст в буфер обмена (clipboard);
- `<Ctrl>+<X>` — вырезать выделенный фрагмент текста и поместить его в буфер обмена;

- <Ctrl>+<V> — вставить фрагмент текста из буфера обмена в позицию курсора;
- <Ctrl>+<K> — удалить текст от курсора до конца строки и поместить его в специальный буфер (kill-buffer);
- <Ctrl>+<Y> — вставить фрагмент текста из специального буфера в позицию курсора.

С помощью комбинации <Ctrl>+<J> можно отформатировать текущий абзац текста. Форматирование, впрочем, заключается только в том, что очень длинные строки текста разбиваются на такие, длина которых не превышает величину, заданную в соответствующей опции пункта меню **Настройки**.

Комбинация <Ctrl>+<F> вызывает функцию поиска подстроки.

Для сохранения результатов редактирования в файле можно воспользоваться пунктом меню **Файл | Сохранить** или комбинацией <Ctrl>+<S>. Если вы хотите сохранить редактируемый текст в файле с другим именем, то тут надо действовать только через меню (**Сохранить как**).

KEdit поддерживает печать. Вы можете распечатать как весь редактируемый файл, так и выделенный фрагмент текста. Печать осуществляется на принтере, заданном по умолчанию. Дополнительно можно подключить любую из утилит печати, которые во множестве разработаны для UNIX.

Что меня особенно восхитило при первом знакомстве с KEdit, так это встроенная возможность проверки правописания и возможность просмотра и редактирования файлов как в кодировке CP-1251, так и в кодировке koi8-r.

Для смены кодировки надо поиграть выбором шрифтов в пункте меню **Настройки | Шрифт**. К сожалению, не все шрифты поддерживают обе кодировки, но после нескольких попыток удастся выбрать нужный шрифт. Для проверки правописания достаточно выбрать пункт меню **Редактирование | Проверка правописания**. Естественно, что в системе должна быть установлена программа *ispell*, и, кроме того, перед началом проверки стоит войти в пункт меню **Настройки | Проверка правописания** и выбрать русский словарь и желаемую кодировку символов (смотри рис. 12.10).

Кстати, пункт меню **Настройки** вообще желательно повнимательнее посмотреть перед началом редактирования каких-то файлов, выбрать нужные вам установки, после чего сохранить их (подпункт **Записать установки**).

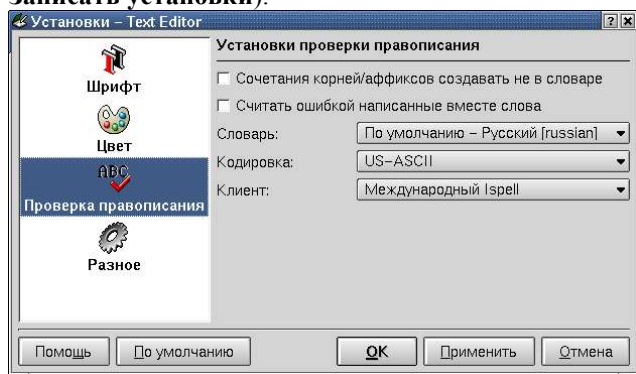


Рис. 12.10. Настройка проверки правописания в KEdit

Еще одна приятная, на мой взгляд, встроенная функция редактора KEdit — возможность вызова почтовой программы прямо из редактора. Я не смог проверить работу этой функции, поскольку на домашнем компьютере (изолированном) у меня некуда посылать письма. Но такой функцией я часто пользуюсь на работе, в MS Word, и она кажется мне очень полезной и удобной.

В общем, этот редактор с первого знакомства произвел на меня самое благоприятное впечатление, так что я рекомендую вам познакомиться с ним поближе. Возможно, вы даже выберете его для редактирования ASCII-файлов при работе в графическом режиме.

12.6.2 Редактор KWrite

Редактор KWrite, как уже было сказано, очень похож на KEdit. Первое отличие, которое бросается в глаза после загрузки в редактор какого-то файла — раскраска служебных слов. Раскраска задается подпунктами **Установить раскраску** и **Раскраска...** пункта **Настройки**. Поскольку я часто занимаюсь редактированием HTML-файла, эта особенность показалась мне удобной.

В основном меню появился один новый пункт — **Закладки** с подпунктами **Установить закладку** (<Alt>+<S>), **Добавить закладку** и **Очистить закладки** (<Alt>+<C>). Остальные пункты главного меню — те же, что и у KEdit.

Появилась функция отката <Ctrl>+<Z> и возможность выделения вертикального столбца (эта функция включается клавишей <F5>). По комбинации клавиш <Ctrl>+<I> в текущую строку вставляется отступ

(добавляется пробел в начало строки). По комбинации клавиш <Ctrl>+<U> отступ убирается (а если отступа не было — молча удаляется первый символ строки).

Пожалуй, это все, что появилось нового в KWrite по сравнению с KEdit. Но, к сожалению, кое-какие полезные функции пропали. В пункте **Файл** исчез пункт, касающийся отправки файла по электронной почте. В пункте **Правка** отсутствует возможность вставить дату. Исчезла также возможность выбрать шрифт в пункте **Настройка | Настройка KWrite**, что я считаю самым большим недостатком этой программы. А в остальном эти редакторы практически одинаковы, так что работать можно с любым из них. Я думаю, что со временем все возможности редактора KEdit будут иметься и в KWrite, и он станет "штатным" текстовым редактором оболочки KDE.

12.6.3 Текстовый редактор Nedit версии 5.1.1

Текстовый редактор Nedit создан группой авторов во главе с Марком Эделем (Mark Edel), распространяется на основе лицензии GPL, и его можно получить на Web-сайте <http://nedit.org>. Внешне он очень похож на два редактора, описание которых было дано в предыдущем разделе, но обладает гораздо большими возможностями, чем KEdit или KWrite.

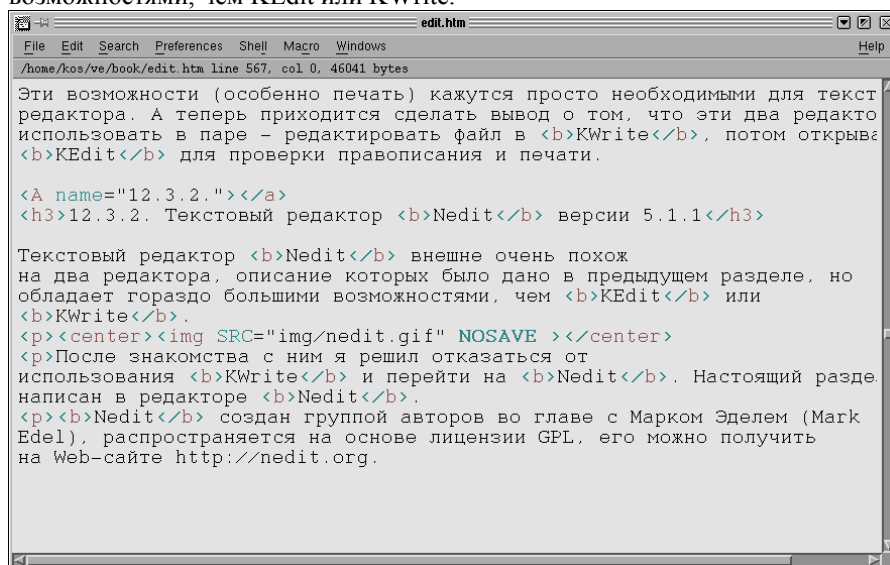


Рис. 12.11. Внешний вид окна редактора Nedit

После первого же знакомства с этим редактором я решил отказаться от использования KWrite и перейти на Nedit. А. Федорчук [П1.6] тоже считает, что Nedit приближается к идеалу текстового редактора, и даже называет его "лучшим редактором всех времен и народов".

Пожалуй, единственным (на мой взгляд, не очень существенным) недостатком Nedit по сравнению с KEdit или KWrite можно считать только то, что меню, сообщения и подсказки выдаются на английском языке. Правда, в меню нет такого пункта, как отправка почтового сообщения, но зато есть возможность выполнить любую команду оболочки shell или создать макрос, что с лихвой перекрывает отсутствие прямого вызова почтовой программы.

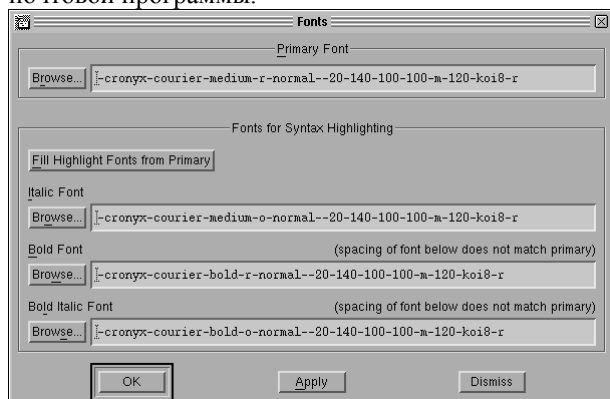


Рис. 12.12. Окно задания шрифтов в Nedit

Переключить редактор на ввод русскоязычного текста для меня не составило труда. Для этого нужно войти в пункт меню **Preferences/Text Fonts...** и задать нужные шрифты в строках ввода соответствующего окна (см. рис. 12.12). А, поскольку запомнить эти длинные строки с именами фонов довольно трудно, можно воспользоваться кнопкой **Browse...** и выбрать шрифт из выводимого списка (см. рис. 12.13). Выбирать надо шрифт, имеющий кодировку KOI8-R. Обратите внимание, что для выбора шрифта надо отметить по одному варианту в каждом из трех предлагаемых столбцов, иначе Вы, вероятнее всего, получите сообщение о том, что такой шрифт отсутствует.

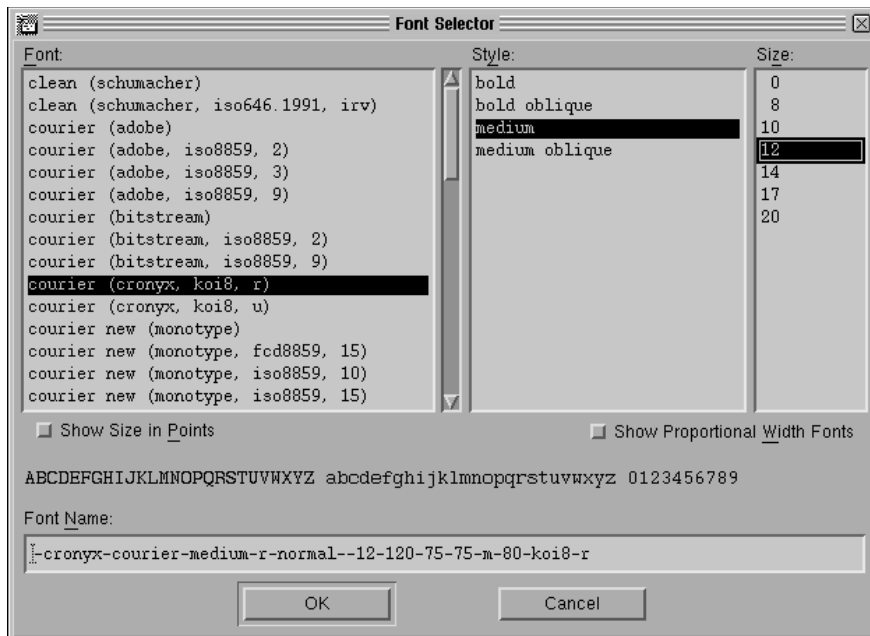


Рис. 12.13. Выбор шрифтов в Nedit

Учитывая, что в силу достаточно долгого периода использования таких редакторов, как редакторы файловых менеджеров FAR для MS Windows и Midnight Commander, у меня сложились устойчивые привычки по использованию некоторых комбинаций клавиш, мне было очень приятно обнаружить, что для вырезания, копирования и вставки блоков текста в Nedit используются привычные комбинации <Ctrl>+<X>, <Ctrl>+<C> и <Ctrl>+<V>. Выделение блока текста осуществляется либо мышкой, либо клавишами-стрелками при удерживаемой клавише <Shift>. Если удерживать <Shift>+<Alt>, будет выделяться прямоугольный столбец. Операции с выделенными блоками можно проделывать не только с помощью клавиатуры, но и через пункты меню **Edit**. Это удобно, если вы не помните комбинацию клавиш для выполнения задуманной операции. А старые привычки не всегда помогают, в частности для отмены последней операции в Nedit используется не <Ctrl>+<U>, а <Ctrl>+<Z>.

В том же меню **Edit** имеются два подпункта (**Lower-case** и **Upper-case**), с помощью которых можно перевести текст в выделенном блоке, соответственно, в нижний и верхний регистры.

Пункт **File** главного меню содержит подпункты, с помощью которых выполняются обычные операции открытия и сохранения редактируемых файлов. Обратите внимание на то, что с помощью подпункта **Open Previous** вы легко можете вернуться к редактированию тех файлов, с которыми работали в предыдущих сеансах. Подпункт **Include file...** (<Alt>+<I>) позволяет вставить содержимое выбранного файла в позицию курсора, подпункт **Print...** служит для вывода редактируемого файла на принтер.

Пункт **Search** главного меню содержит обращения к операциям поиска и замены, а также быстрого перехода к нужной строке или к сделанной ранее закладке (и установки такой закладки-отметки).

Для настройки программы в соответствии с Вашими вкусами и привычками служит пункт меню **Preferences**. Я уже рассказывал, как выбрать шрифт для вывода текста в окне программы. Здесь же вы можете с помощью команды **Language Mode** выбрать язык (например, html или C), по правилам которого будет осуществляться подсветка элементов текста (служебных слов или тэгов), установить размер табуляции (**Tab...**), а также включить или выключить нумерацию строк текста, отображение строки статистики, подсветку синтаксиса, режим сохранения резервной копии файла, режим вставки/замены символа.

Пункт **Shell** используется для запуска команд оболочки и внешних программ. Здесь Вы, в частности, найдете вызов программы проверки правописания **ispell**, только нет возможности через меню программы подключить русский словарь. Но эта трудность легко преодолевается за счет использования файла ресурсов.

nedit в домашнем каталоге пользователя. Достаточно в строке, определяющей вызов программы **ispell**, добавить параметр `-d russian` и проверка русскоязычных текстов заработает (в отдельном окне терминала).

Через тот же файл `~/nedit` можно настроить цвета фона и текста, геометрию окна программы и некоторые другие параметры. То же самое можно сделать и в специальной секции `[nedit]` файла `.Xdefaults` в Вашем домашнем каталоге. Правда, такой секции в этом файле по умолчанию нет, но ее можно создать, как это рекомендуется в книге А.Федорчука.

Но вернемся к пунктам меню. Пункт **Windows** служит для переключения между окнами (которых можно открыть одновременно несколько). Назначение пункта **Help**, я думаю, понятно без дополнительных пояснений. К сожалению, подсказка дается по-английски и отсутствует поиск подсказки по ключевым словам и фразам.

В этом кратком описании я не касался пункта меню **Macro**, а также тех подпунктов главного меню, которые ориентированы на действия, выполняемые при редактировании исходных текстов программ (например, вставка кодов ASCII, подсветка парных скобок или компиляция программного кода). Но даже и на основании того, что было сказано в данном подразделе, можно сделать вывод, что Nedit — это достаточно мощный и удобный редактор для работы с ASCII-файлами в графическом режиме. Если вы хотите узнать о нем побольше, почитайте статьи А.Федорчука в Интернет или его книгу [П1.6].

12.7. Текстовые процессоры

Давайте договоримся, что термином "текстовые процессоры" будем обозначать все программы для редактирования текстовых файлов, работающие в графическом режиме и использующие при сохранении результатов работы в файлах специальные символы или вставки для обозначения элементов форматирования. Эти вставки текстовый процессор не отображает в виде каких-либо символов в процессе редактирования. В этом смысле Nedit не относится к текстовым процессорам, а Netscape Composer — относится, потому что показывает элементы форматирования в графическом виде, а не в виде html-тегов.

12.7.1 Возможности текстовых процессоров

К стандартным средствам форматирования текста относятся:

- возможности выбора различных шрифтов для разных частей текста в одном документе;
- задания ширины полей, величины отступов, интервалов;
- организации текста в виде колонок;
- вставка в текст рисунков, таблиц;
- возможности создания надписей под углом к строке или искривленной формы;
- проверка правописания (для языка пользователя)

и т. д.

Для того чтобы текстовый процессор мог считаться мощным и удобным, необходимо, чтобы в нем имелись возможности обработки файлов, созданных в других распространенных текстовых процессорах (Конечно, в первую очередь имеется в виду MS Word!). Как минимум нужно уметь просматривать документы в форматах других редакторов. Можно, конечно, иметь перекодировщики, но с точки зрения удобства использования, это уже минус.

Имея в виду это неформальное определение, рассмотрим несколько программных продуктов для Linux, претендующих на высокое звание текстового процессора.

Поскольку по своему основному предназначению текстовые редакторы тесно связаны с языком, а мы рассматриваем случай русского языка, при выборе текстового процессора особое внимание приходится уделять тому, имеются ли в нем возможности русификации. Чтобы не повторять одно и то же при описании каждого из рассматриваемых ниже продуктов, замечу сразу, что в моей системе были установлены не только стандартные шрифты TrueType, поставляемые с дистрибутивом Black Cat 6.02, но и шрифты TrueType от Windows (см. гл. 11).

12.7.2 Текстовые процессоры для Linux

Текстовых процессоров для Linux существует множество. Наиболее известны из них StarWriter из пакета StarOffice фирмы Sun и процессор Word Perfect 8. Аналог последнего, разработанный для ОС Windows, долгое время на равных конкурировал с пакетом MS Word, что, конечно, говорит о его высоком качестве. Однако, с русификацией Linux-версии этого процессора какие-то проблемы, поэтому мы его здесь не рассматриваем.

Особое место среди текстовых процессоров занимает издательская система T_EX и основанные на ней продукты типа L_AT_EX. Что касается T_EX, то это скорее язык программирования, чем текстовый редактор. Если вы программируете на двух-трех разных языках, вам, возможно, не составит труда освоить и T_EX. Но простому пользователю, которому редактор нужен для написания деловых писем или диссертации по истории паровых котлов, на мой взгляд, вряд ли стоит браться за его изучение.

12.7.3 Текстовый редактор Ted

Редактор Ted задуман автором (Mark de Does) как простой текстовый редактор, работающий под X Window в Unix/Linux системах, и играющий примерно ту же роль, что и WordPad под MS-Windows, только имеющий больше возможностей. Основное преимущество Ted по сравнению с редакторами, которые рассматривались в предыдущих разделах, — это возможность не только вводить и редактировать текст, но и обеспечивать простейшие возможности форматирования текста, изменять шрифты, вставлять в текст таблицы и рисунки. Именно в силу наличия этих возможностей я причислил этот редактор к текстовым процессорам, хотя он и не обладает всеми возможностями развитых текстовых процессоров.

Дистрибутив пакета можно скачать с сайта <ftp://ftp.nluug.nl/pub/editors/ted> или с сервера metalab: [\[http://metalab.unc.edu/pub/packages/editors/ted\]](http://metalab.unc.edu/pub/packages/editors/ted).

Дистрибутив поставляется либо в виде сжатого tar-архива, либо в виде RPM-пакета. Русифицированный вариант редактора можно найти на сервере <ftp://ftp.logic.ru/> (конечно, русификация по версиям чуть отстает).

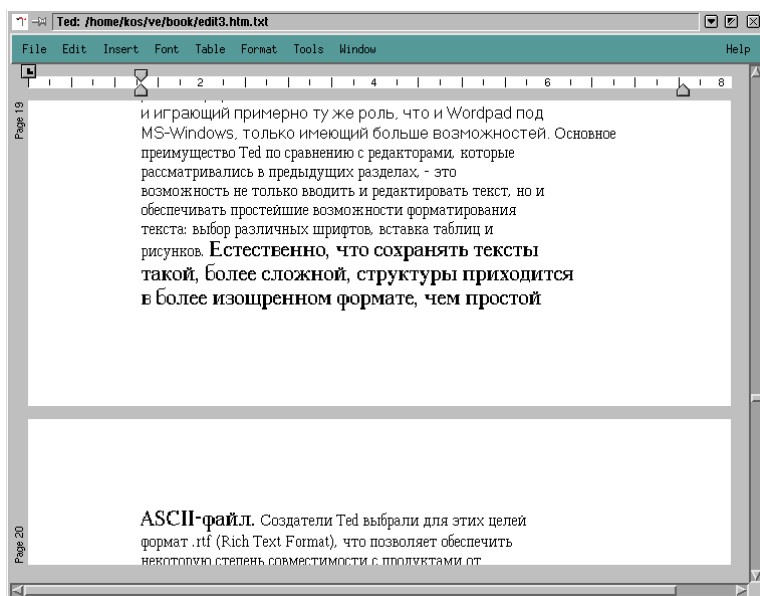
Я устанавливал и русифицированную версию 2.6 редактора Ted, однако после некоторого периода испытаний вернулся к англоязычной версии 2.7, которая датирована 31 декабря 1999 г., и в которой устранены некоторые недостатки предыдущей версии.

После ее запуска появляется окно, в котором изображена гравюра в стиле Альбрехта Дюрера, указана версия программы, ссылка на сайт разработчиков и имеется всего три пункта меню: **Fi le**, **Window** и **Help**.

Очевидно, что для того, чтобы открыть существующий файл для редактирования, нужно выбрать подпункт **Open** меню **File** и выбрать в появившемся окне нужный файл. Для создания нового файла соответственно выбираем подпункт **New** меню **File** или набираем <Ctrl>+<N>. После открытия файла окно программы приобретает вид, изображенный на рис. 12.14.

Ted поддерживает формат файлов .rtf (Rich Text Format), что позволяет обеспечить некоторую степень совместимости с продуктами от Microsoft. А именно, любой файл, созданный в Ted, и сохраненный в формате rtf, будет корректно прочитан программами MS Word под Windows. Обратная совместимость (т. е. возможность открыть любой rtf-файл в Ted) не достигнута, поскольку некоторые опции форматирования, используемые в rtf-файлах, программой Ted не воспринимаются. Тем не менее текст открываемого файла будет выведен на экран полностью, и может быть сохранен в другом (или том же самом) файле, но уже без той информации о форматировании, которая игнорируется при открытии файла.

Рис. 12.14. Основное окно редактора Ted



Приведу краткий список основных возможностей Ted.

- Редактирование файлов в режиме WYSIWIG. При этом вы можете использовать все шрифты, для которых у вас имеются .afm-файлы и которые доступны для X11. Ted поставляется с .afm-файлами для шрифтов Adobe, которые имеются для систем, основанных на Motif, и могут использоваться со всеми postscript-принтерами: Times, Helvetica, Courier и Symbol. Другие шрифты могут быть добавлены с помощью обычных процедур, используемых для этого в X11. Поддерживаются такие свойства шрифтов как выделение (bold), курсив (italic), подчеркивание, верхние и нижние индексы.
- Вставка рисунков в формате bitmap.
- Печать на PostScript-принтерах.
- Работа с файлами в формате Acrobat PDF.
- Проверка правописания для 12 языков, основанных на латинском алфавите.
- Отправка документов по e-mail не выходя из Ted.
- Операции Cut/Copy/Paste, включая обмен с другими приложениями.
- Поиск/Замена.
- Форматирование абзацев: отступ на первой строке, смещение всего абзаца, использование линеек.
- Разбиение текста на страницы.
- Вставка таблиц и операции с таблицами: операции с колонками и строками таблиц, изменение размеров с помощью линеек.
- Поддерживается вставка символов из разных кодовых таблиц.
- Гиперссылки и закладки (bookmarks).
- Сохранение документа в текстовом и HTML-формате.

Ted можно использовать для чтения сообщений электронной почты, посланных с Windows-машин. Можно также сконфигурировать Ted как просмотрщик RTF-файлов для Netscape.

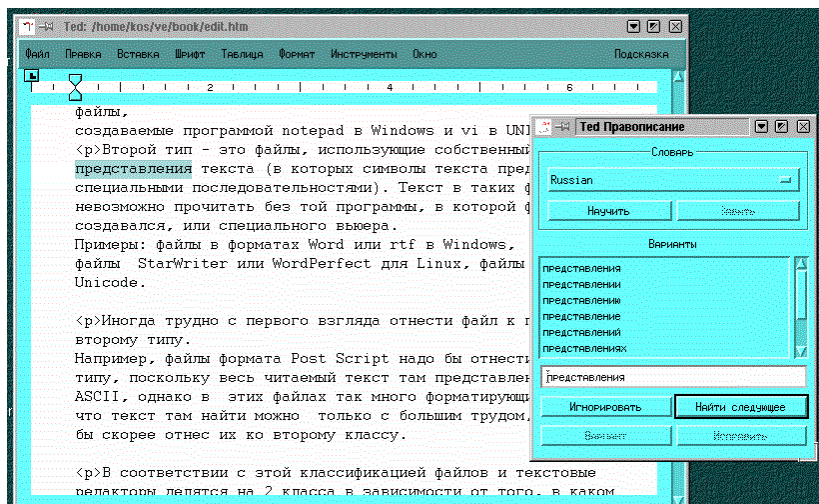


Рис. 12.15. Проверка правописания в редакторе Ted

О том, что Ted может быть полностью русифицирован, я узнал из статьи Виктора Вагнера [П14.19]. Самая интересная возможность русифицированной версии — наличие проверки правописания для русскоязычного текста. Вызов этой функции производится с помощью пункта меню **Инструменты | Правописание**. При этом появляется дополнительное окно (см. рис. 12.15), в котором надо щелкать по кнопке "Найти следующее" для поиска очередной ошибки.

Правда, судя по первому впечатлению, в той версии, которую я видел (версия 2.6), реализация данной функции имеет много недостатков. Во-первых, все слова, набранные латинским шрифтом, обозначаются как ошибки (хотя даже **ispell** без проблем игнорирует их). Во-вторых, воспринимаются как ошибки слова, начинающиеся с заглавной буквы (программа предлагает такое слово набрать полностью заглавными буквами). И, что самое непонятное, очень много правильно набранных слов воспринимаются как ошибочные, причем первым же вариантом исправления предлагается в точности то же самое слово. Впрочем, много и других ошибок. Именно поэтому я вернулся к версии 2.7 и, следуя рекомендациям В.Вагнера, попытался русифицировать ее.

Первым делом надо научить Ted вводить и отображать русские буквы. Для того, чтобы Ted правильно отображал шрифты в кодировке koi8-r на экране, надо создать файл `xfonts.dir` в директории `/usr/local/afm`, где Ted держит метрики шрифтов (*.afm), поместив в него следующий текст

```
Courier-Bold *-courier-bold-r-normal-----*
Courier-BoldOblique *-courier-bold-o-normal-----*
Courier-Oblique *-courier-medium-o-normal-----*
Courier *-courier-medium-r-normal-----*
Helvetica-Bold *-helvetica-bold-r-normal-----*
Helvetica-BoldOblique *-helvetica-bold-o-normal-----*
Helvetica-Oblique *-helvetica-medium-o-normal-----*
Helvetica *-helvetica-medium-r-normal-----*
Times-Bold *-times-bold-r-normal-----*
Times-BoldItalic *-times-bold-i-normal-----*
Times-Italic *-times-medium-i-normal-----*
Times-Roman *-times-medium-r-normal-----*
```

После этого русские буквы вводятся и отображаются нормально. Однако проблема решена еще не полностью: после того как поработаешь с **Format tool** или любым диалогом, содержащим строку ввода, Ted снова перестает вводить русские буквы. Решение этой проблемы, предложенное Иваном Паскалем, состоит в том, что создается пустой файл `/usr/X11R6/lib/X11/locale/koi8-r/Compose` и в файл `/usr/X11R6/lib/X11/locale/Compose.dir` добавляется строка:

```
koi8-r/Compose ru_RU.KOI8-R
```

Это решение у меня вполне сработало (за исключением того, что клавиша <Delete> перестала удалять символы и стала вставлять пробелы, так что для исправления ошибок приходится пользоваться клавишей <BackSpace>).

Теперь рассмотрим вопрос о добавлении новых шрифтов. Вместе со стандартным дистрибутивом Ted поставляются только 4 шрифта: Courier, Helvetica, Symbol и Times. Если вы хотите оформлять свои документы более разнообразно, нужно добавить другие шрифты. В Ted можно использовать любой шрифт из числа тех, которые видны у вас в графической оболочке, однако для этого нужно кое-что сделать. Чтобы

узнать, какие шрифты у вас имеются, запустите программу `xfontsel` и выберите тот шрифт, который хотите добавить. Предположим, что это Bookman Light.

Для того, чтобы Ted увидел новый шрифт, необходимо поместить метрику этого шрифта (метрика — это файл формата `.afm`) в директорию `/usr/local/afm`. Метрику можно либо найти готовую, либо создать самому. Ted весьма привередлив к метрикам шрифта. Как пишет В.Вагнер, Ted долгое время отказывался грузить шрифты от kapella, поскольку в `afm`-файле отсутствуют упоминания нескольких символов, фигурирующих в Adobe Standard Encoding. Вместе с тем, работа с метриками шрифта — обязательное свойство более-менее продвинутой программы форматирования, поскольку той информации о шрифте, которую предоставляет X сервер, недостаточно для качественной печати. Если вы попытаетесь работать с Ted, используя метрики, не соответствующие Вашим шрифтам, результат будет безобразный — слова будут наезжать друг на друга. Готовые метрики шрифтов можно найти в коллекции Adobe base35, которая находится на FTP-сервере <ftp://ftp.adobe.com>, а инструкцию по их установке — в файле подсказки по редактору Ted.

В заключение замечу, что Ted позволяет осуществить русификацию интерфейса. Достаточно перевести на русский язык файл ресурсов, который находится в каталоге `/usr/X11R6/lib/X11/app-defaults/ru` и называется Ted. Если такого каталога нет, его нужно создать и складывать туда все русифицированные ресурсы. После этого, будучи запущенным с русской локалью, Ted будет выводить все меню и большую часть сообщений в диалоговых окнах по-русски.

12.7.4 Текстовый процессор AbiWord

AbiWord — это маленький, быстрый, бесплатный и легкий в освоении и использовании текстовый процессор для X. Этот пакет можно скачать из Интернет с сайта www.abisource.com. После запуска программы вы увидите основное окно программы (рис. 12.16), которое сильно напоминает знакомое всем окно MS Word. Процессор AbiWord обладает всеми особенностями и функциями, которые присущи текстовым процессорам, включая проверку правописания, причем работающую непосредственно в процессе ввода текста.

AbiWord позволяет открывать для редактирования и просмотра файлы формата Microsoft Word (как MS Word 95, так и MS Word 97). Однако, поскольку для преобразования форматов в нем используется вызов процедур пакета `wv`, который мы рассматривали выше, сказываются все недостатки последнего.

После редактирования файл можно сохранить в одном из следующих форматов:

- в собственном формате AbiWord (`.abw`);
- в текстовом формате (`.txt`);
- в формате RTF (`.rtf`);
- в формате HTML;
- в формате UTF8;
- в формате LaTeX.

Таким образом, в принципе, через формат RTF можно организовать обмен файлами с пользователями MS Word.

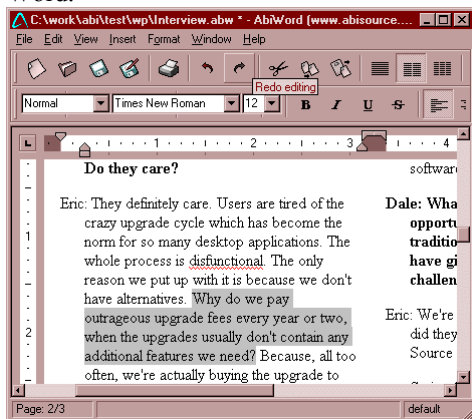


Рис. 12.16. Основное окно редактора AbiWord

К сожалению, пока что этот редактор еще находится в состоянии разработки (я смотрел версии 0.7.8 и 0.7.14). При обращении к некоторым пунктам меню вместо указанного (или ожидаемого) результата появляется сообщение о том, что данный пункт пока не работает. В версии 0.7.8 это относилось к таким функциям, как создание нумерованных списков, стили, нумерация страниц и вставка символов. Вместо выполнения соответствующей функции появлялось сообщение: "пока не реализовано" ("not implemented yet"). И приглашение-просьба: если вы программист, присоединяйтесь к разработке, а если не программист,

то будьте терпеливы. В версии 0.7.14 некоторые из этих недоработок уже устранены, однако часть ошибок все еще не исправлена. В частности, если в диалоговом окне открытия файла просто подсветить какой-то каталог и попытаться перейти в него нажатием клавиши <Enter> , происходит вызов нового экземпляра программы. Правда, текущим каталогом для этого нового экземпляра оказывается тот каталог, куда мы и хотели попасть, но напрасное открытие новых окон раздражает (да и ресурсы зря расходуются). Если же осуществлять переход в каталог двойным щелчком мыши, то нового окна не открывается. Есть и другие ошибки, причем при некоторых редактор просто закрывается, без предупреждений и сохранения результатов предыдущей работы. Так что считать это продукт готовым к применению пока что рановато. Однако в целом он производит очень приятное впечатление и версия 0.7.14 кажется гораздо более работоспособной, чем 0.7.8. Это свидетельствует о том, что работа над программой продолжается. Поэтому давайте наберемся терпения в надежде, что появление полноценного продукта, реализующего все заявленные в меню возможности, не за горами.

Между прочим, по адресу <http://www.hippo.ru/~hv/abiword/> можно найти русифицированную версию AbiWord.

12.7.5 Текстовый процессор KWord

Текстовый процессор KWord входит в состав офисного пакета KOffice. Однако, в отличие от StarOffice, для его запуска не требуется запуск какой-то общей оболочки или общего офисного desktop. После запуска программы (проще всего это сделать из главного меню оболочки KDE, выбрав пункт **Офис | Текстовый процессор**), появляется непритязательное серое окошко (см. рис. 12.17), на фоне которого тут же возникает окно выбора типа открываемого документа (рис. 12.18).

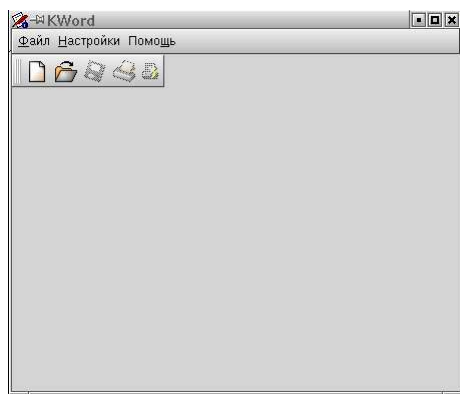


Рис.12.17. Стартовое окно программы KWord

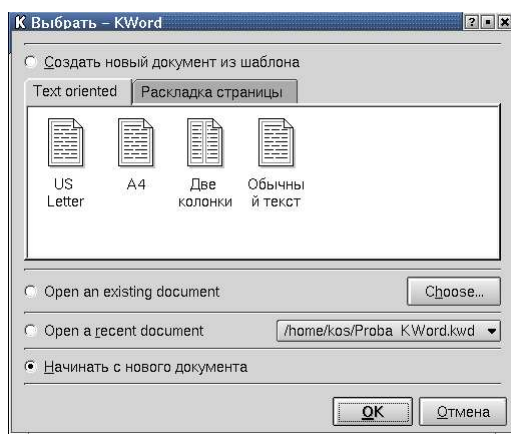


Рис. 12.18. Окно выбора типа открываемого документа

В этом окне предлагается либо создать новый документ на основе одного из шаблонов, либо открыть существующий документ. Здесь же имеется возможность просмотреть список тех документов, с которыми вы работали в предыдущих сеансах и быстро вызвать один из них.

Как только вы вызвали на редактирование какой-то документ, или открыли новый, вид окна кардинальным образом изменяется (рис. 12.19). Появляются и новые пункты меню и панели инструментов и линейки форматирования страницы.

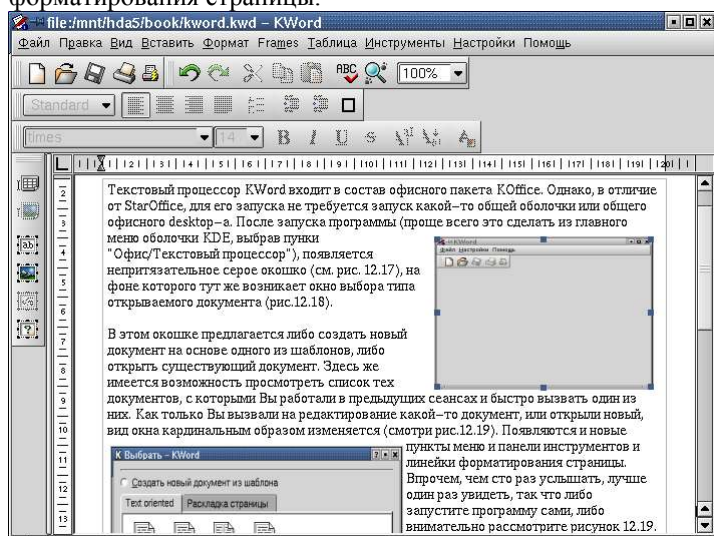
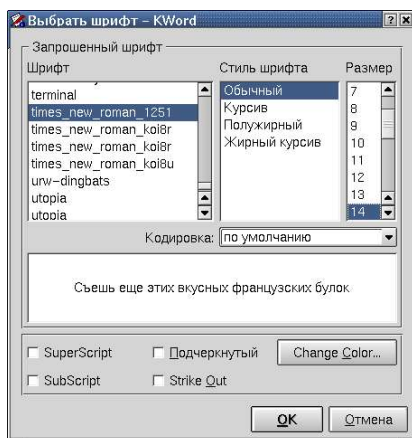


Рис. 12.19. Основное окно программы KWord

Работа с новым текстовым редактором обычно начинается с создания нового документа, поэтому и мы давайте создадим новый документ и попытаемся вводить текст. Для начала надо выбрать шрифт, естественно, русифицированный. В KWord это делается через пункт меню **Формат | Шрифт**. После выбора этого пункта меню появится окошко выбора шрифта (рис. 12.20), в котором этот самый выбор и осуществляем.

В русифицированной версии Linux среди этих шрифтов найдутся и шрифты в кодировках KOI8-R и cp1251. Выбираем один из них и начинаем вводить текст. Кстати, основные характеристики шрифта можно выбрать и, не заглядывая в меню, поскольку на панели инструментов (рис. 12.20) имеются выпадающие списки для наименования шрифта, размера символов, значки для задания жирного, подчеркнутого, зачеркнутого шрифта, а также для изменения цвета символов и превращения части текста в верхние и нижние индексы. Впрочем, значки на панели инструментов можно добавлять или, наоборот, убирать. Но о настройках панелей



инструментов поговорим позже, а пока вернемся к набору текста.

Рис. 12.20. Выбор шрифта в KWord

Как только введен первый абзац, возникает желание придать ему благообразный вид. Что и сделаем с помощью пункта меню **Формат | Абзац**. Появится окно с пятью вкладками, изображенное на рис. 12.21. На первой вкладке можно задать отступы для строк абзаца (отдельно для первой строки и остальных), межстрочные интервалы и интервалы между двумя последовательными абзацами. Величина всех интервалов и отступов задается в миллиметрах. На второй вкладке задается положение строк данного абзаца на странице: смещение их влево, вправо, по центру или равномерное растяжение по ширине страницы.

Вкладка **Рамка** позволяет заключить данный абзац в рамку (например, чтобы как-то выделить данный абзац из общего текста). Две последних вкладки служат для задания формата списков и установки позиций табуляции, но об этом поговорим позже (если выдастся случай).

Следующий пункт в меню **Формат** позволяет задать формат целой страницы. После выбора пункта **Формат / Страница** появляется окно с тремя вкладками. Первая из них служит для задания ориентации страницы (книга или альбом), размера страницы (например, A4), ширины полей. На второй вкладке можно задать размещение текста в несколько колонок (столбцов). И третья вкладка определяет свойства верхнего и нижнего колонтитулов. Но нам пока еще не до таких тонкостей, давайте закончим набор первой страницы текста и посмотрим, как же готовая страница будет выглядеть на экране.

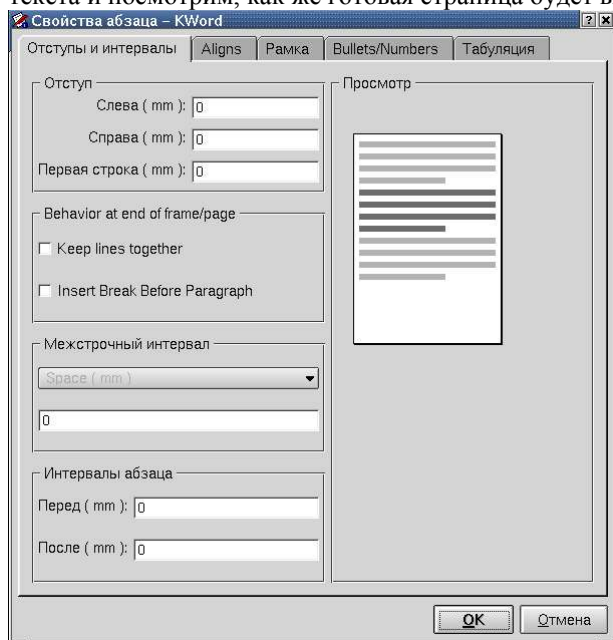


Рис. 12.21. Окно форматирования абзаца

Только вначале отметим, что так же, как и для выбора шрифта, для форматирования абзаца не обязательно пользоваться соответствующим пунктом меню. Во-первых, можно использовать панель инструментов "Формат", на которой можно выбрать один из стандартных стилей абзаца (правда, поначалу список выбора стиля не богат, но можно создать собственный стиль через пункт меню **Формат | Менеджер стилей**), положение строк на странице и сдвинуть абзац вправо. Кроме панели инструментов для форматирования абзаца можно воспользоваться линейками слева и сверху окна ввода текста. Эти линейки позволяют задать поля страницы и отступы абзаца.

Итак, если вы закончили ввод текста или хотя бы набрали целую страницу, можно посмотреть, как эта страница будет выглядеть отпечатанной. Для этого надо воспользоваться пунктом меню **Файл | Предварительный просмотр** или соответствующим значком на панели инструментов "Файл". В отличие от MS Word для предварительного просмотра страницы вызывается отдельная программа — "Просмотрщик PS/PDF" (о ней было сказано несколько слов выше). Если что-то в изображении страницы вам не понравится, можно вернуться к форматированию страницы и абзацев, чтобы после внесения изменений снова перейти к предварительному просмотру. Только, чтобы увидеть изменения, просмотрщик придется открыть снова, так что текущий экземпляр лучше закрывать.

Чтобы не потерять нечаянно результаты работы, давайте сохраним файл на диске. Для сохранения файла можно воспользоваться пунктом меню **Файл | Сохранить** или **Файл | Сохранить как**. В последнем случае можно задать новое имя файла и выбрать каталог, в котором окажется новый файл.

Раз уж мы заглянули в меню **Файл**, давайте посмотрим, какие в нем еще есть пункты. В общем, ничего неожиданного здесь не обнаруживается. Имеются пункты **Новый**, **Открыть**, **Открыть недавние**, **Сохранить**, **Сохранить как**, **Создать шаблон из документа**, **Статистика** (в этом пункте можно получить сведения о числе символов, слов и предложений в набранном тексте), **Печать**, **Просмотр перед печатью**, **Сведения о документе** (здесь вы можете ввести некоторые данные о себе, как авторе документа), **Заккрыть** (текущий документ) и **Выход**.

Меню **Правка** тоже не содержит ничего неожиданного: в нем есть пункты **Отменить** (предыдущее действие), **Повторить** (т. е. заново выполнить отмененное действие), **Вырезать**, **Копировать**, **Вставить**, **Выбрать всё** (т. е. выделить все), **Поиск** и **Замена**.

А вот следующее меню для пользователей MS Word окажется непривычным. Оно называется **Вид**. В MS Word тоже имеется пункт меню с таким названием, но он содержит совершенно другие подпункты. А в KWord здесь обнаруживаются пункты:

Новый вид — выбор этого пункта приводит к открытию нового окна редактора KWord, в котором выведен тот же документ, который вы редактировали в первом окне. В отличие от запуска новой копии программы (нового окна с тем же файлом) при открытии нового вида все изменения, вносимые Вами в текст, будут отображаться в обоих окнах.

Закрыть все виды — приводит к выходу из программы.

Разделить вид — окно редактирования делится на две или большее число панелей, в каждой из которых выводится редактируемый файл. То же самое, что и **Новый вид**, только без открытия нового окна. В каждом виде можно перемещаться по тексту независимо.

Удалить вид — закрывает текущий вид, т. е. ту панель или окно, в котором в данный момент находится курсор.

Расположение разделителя — позволяет расположить панели видов вертикально или горизонтально.

Далее идут два пункта-переключателя: **Page mode** и **Preview mode**. В режиме **Page mode** в панели редактирования отображается одна страница текста, как вы видели на рис. 12.19. В режиме **Preview mode** на экран выводятся сразу все страницы документа. Линейки прокрутки позволяют перемещаться по страницам. Это позволяет посмотреть, как будут располагаться рисунки, таблицы и т. д.

Далее идет неактивизированный пока (я работал с версией 1.1 post-beta программы KWord) пункт **Непечатаемые символы**, после которого следуют еще три переключателя: **Рамка врезки**, **Верхний колонтитул** и **Footer** (в смысле "нижний колонтитул"). Эти три переключателя включают (или, соответственно, выключают) отображение соответствующих элементов оформления страницы (впрочем, **Рамка врезки** не является элементом оформления страницы, она видна только на экране).

Назначение последнего пункта в меню **Вид** очевидно из его названия — **Масштаб**. Впрочем, вместо этого пункта удобнее пользоваться выпадающим списком на панели инструментов. Тем более, что задать произвольное значение масштаба невозможно ни тем, ни другим способом, можно только выбрать одно из предустановленных значений.

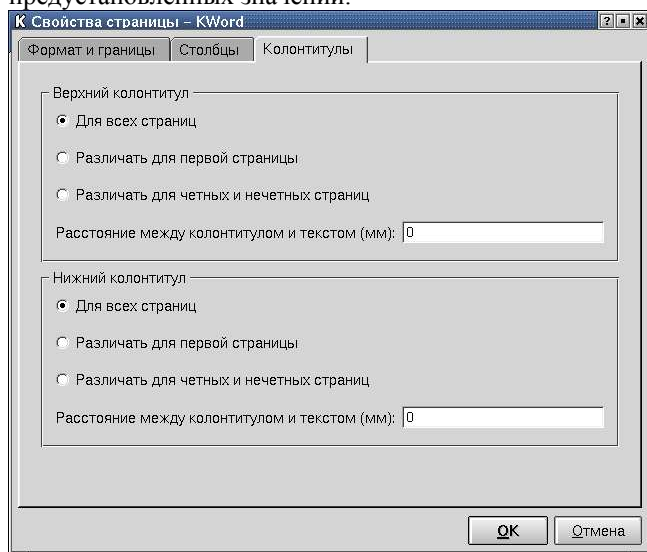


Рис. 12.22. Параметры верхнего и нижнего колонтитулов

Раз уж речь зашла о колонтитулах, стоит упомянуть, что на вкладке **Колонтитулы**, которая становится доступной при выборе пункта меню **Формат | Страница**, можно задать по три параметра для верхнего и нижнего колонтитулов (смотри рис. 12.22).

В одном из колонтитулов обычно выводят номер текущей страницы. В KWord это можно сделать из следующего меню — **Вставить**. Надо выбрать пункт **Переменная** и из появившегося списка выбрать пункт **Page Number**.

А вообще через меню **Вставить** можно вставить картинку, таблицу, формулу, специальный символ, разрыв страницы, содержание, переменную, выражение (Expression), текстовый фрейм, фрейм с картинкой, фрейм с объектом (Object Frame). Правда с картинками работать достаточно сложно. Щелчок мышкой по картинке не приводит к появлению меню или каких-то символов, служащих для изменения размера. Надо умудриться щелкнуть по рамке картинки в левом верхнем углу, где имеется маленькая область, щелчок по которой

приводит к появлению маленьких квадратиков в углах изображения и центрах сторон, уцепившись за которые, можно изменять размер картинки.

Несколько успешнее удастся вставлять фрейм с картинкой. Вызвать меню управления фреймом тоже нелегко — надо попасть в ту же маленькую область на границе фрейма правой кнопкой мыши, чтобы вызвать это меню, но все же уже можно что-то сделать. После вызова меню появляется окно, изображенное на рис. 12.23.

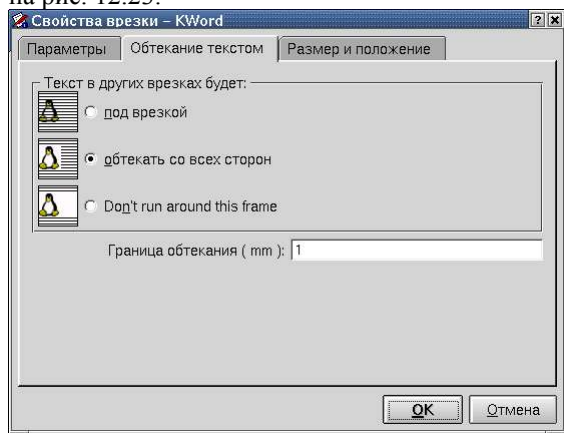


Рис. 12.23. Окно настройки фрейма (врезки)

В этом окне можно задать взаимоотношения основного текста и врезки (см. рис. 12.23) и размер врезки, который для фреймов с картинками определяет и размеры изображения. Однако попытки изменения размеров картинок очень часто приводят к тому, что программа "сваливается".

Вставка таблиц работает, однако таблицы получаются пока не слишком красивые. Вставка формул — это пока, вообще, что-то неприменимое. Подпункт "Expression" позволяет вставить в текст несколько фиксированных фраз (привет, прощай, адрес e-mail и т. п.). В качестве вставляемых переменных могут использоваться номер страницы, число страниц в документе, имя файла, имя текущего каталога, текущие дата и время (причем, как я понял, либо время вставки этого параметра, либо время последнего редактирования документа), те данные, которые вы ввели в пункте **Файл | Сведения о документе**, а также любые определенные Вами значения. По поводу вставки переменных тоже можно высказать некоторые претензии. В частности, после первого сохранения файла вставка времени и даты сохраняется, а после второго сохранения — пропадает.

В меню **Инструменты** мы видим три пункта: **Проверка правописания**, **Пользовательские переменные** и **Автозамена**. Проверка правописания представляет собой графическую оболочку к встроенному вызову программы `ispell`. При обнаружении незнакомого слова оно выделяется и выводится окно, изображенное на рис. 12.24. Можно заменить слово одним из предлагаемых вариантов или пропустить его ("игнорировать" ошибку). Можно также напрямую поправить слово в строке предлагаемой замены, после чего нажать кнопку **Заменить**.

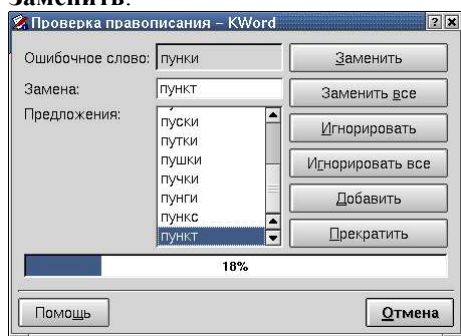


Рис. 12.24. Возможные реакции на ошибочные слова

Пункт **Пользовательские переменные** служит для определения собственных переменных, а пункт **Автозамена** — для задания правил автоматического исправления наиболее характерных и часто встречающихся ошибок, типа двух заглавных букв в начале слова. Можно изменить правила замены или задать собственные.

Через пункт меню **Настройки** можно включить или отключить отдельные панели инструментов, настроить вид этих панелей и привязки "горячих" клавиш, а также произвести некоторые настройки самой программы.

Ну, и, наконец, последний пункт главного меню, имеющий традиционную для KDE структуру: помощь по программе, сведения о Kword, и сведения о KDE. Помощь по программе пока (в версии 1.1) отсутствует. На этом описание текстового процессора KWord можно считать завершенным. К сожалению, пока что и его нельзя признать полностью функциональным тестовым процессором. Главные претензии — работа с рисунками и таблицами. Но с русским языком процессор работает вполне нормально и даже позволяет выбрать кодировку символов. Как это сделать в MS Word (если это вообще возможно), мне не известно. Так что для создания не очень сложных документов KWord вполне пригоден.

12.7.6 Текстовые процессоры StarWriter и OpenOffice.org Writer

Текстовый процессор StarWriter входит в состав офисного пакета StarOffice фирмы Sun. Домашняя страница: <http://www.sun.com/products/staroffice>. Скачать с домашней страницы можно бесплатно, CD-ROM стоит \$39.95 (в США).

Первоначально пакет StarOffice был разработан немецкой фирмой StarDivision, которая была основана в середине 1980-х годов. Летом 1999 года фирма StarDivision была куплена вместе с ее разработкой фирмой Sun. Бета-версия StarOffice 5.1a, выпущенная уже от имени Sun, имела мало принципиальных отличий от тех версий, которые выпускались первоначальным разработчиком. Конечно, появилась эмблема Sun и была значительно (на мой взгляд) облегчена процедура установки, которая стала очень напоминать процедуры установки Windows-продуктов: ответь на несколько вопросов, а дальше сиди и читай сменяющие друг друга рекламные сообщения. Возможно, это явилось следствием того, что пакет StarOffice выпущен также и в варианте для Windows.

В июне 2000 г. Sun выпустила версию 5.2 пакета, причем StarWriter позиционировался как альтернатива Microsoft Word, удовлетворяющая самым придирчивым требованиям. И действительно, StarWriter имеет довольно большой набор файловых фильтров и обладает довольно большим сходством с Microsoft Word по набору основных операций, так что переход от MS Word к StarWriter должен происходить для пользователя достаточно просто. Более того, StarWriter обеспечивает большинство возможностей, которые имеет MS Word. С его помощью можно создавать (в режиме WYSIWYG) документы высокого качества. Легко можно получить текст, разбитый на несколько колонок, вставить в текст рисунки и таблицы, врезки, задать формат абзацев, ширину полей и т.п. Судя по описаниям, имеются возможности, присущие профессиональным издательским системам, такие как управление величиной расстояния между соседними буквами (кернинг), задание стилей для отдельных фрагментов текста (заголовков, цитат и т.п.). С помощью StarWriter можно создавать конверты, наклейки, письма.

Если говорить о том, в каких случаях стоит использовать StarWriter в качестве текстового процессора, то в первую очередь надо упомянуть тот факт, что он неотделим от остальных частей пакета StarOffice. Пакет StarOffice сам по себе представляет собой интегрированную графическую среду, из которой можно запускать все другие приложения, так что он достаточно требователен к ресурсам. Поэтому вряд ли целесообразно использовать StarWriter в таких случаях, когда надо создать небольшую служебную записку или коротенькое электронное письмо (если, конечно, вы не используете весь пакет StarOffice в качестве своей рабочей среды).

К сожалению, даже версия 5.2 StarWriter не является полностью русифицированной и недостаточно корректно обрабатывает документы, созданные с помощью MS Word, поэтому после недолгих экспериментов с этой программой я отказался от ее использования. Подробнее о пакете StarOffice в целом вы можете прочитать в книге А.Федорчука "Офис, графика, Web в Linux" [П1.6].

13 октября 2000 года Sun открыла исходные коды пакета StarOffice. Тем самым появился проект OpenOffice.org (обратите внимание на приставку .org – как говорится на Web-странице проекта, именно так следует правильно именовать данный проект, поскольку "OpenOffice" является зарегистрированной торговой маркой какого-то другого продукта).

Исходные коды OpenOffice.org основаны на технологии, которая первоначально была разработана Sun Microsystems для будущих версий пакета StarOffice(TM). В отличие от StarOffice проект OpenOffice.org не содержит интегрирующей оболочки - архитектура пакета предполагает отдельное использование входящих в пакет приложений. Пакет написан на языке C++, и включает в себя все основные офисные приложения, такие как текстовый процессор, электронную таблицу, программу управления презентациями, программу для работы с графикой, средства создания диаграмм и редактирования формул. Но OpenOffice.org не включает в себя клиента электронной почты, календаря и браузера.

Среди других новшеств можно отметить использование формата файлов, основанного на стандарте XML, что облегчает взаимодействие с другими продуктами (и, следовательно, с другими пользователями компьютеров), а также обеспечивает ясные перспективы развития продукта в будущем. В состав продукта включены конверторы для других распространенных файловых форматов, включая форматы всех версий Microsoft Office, и улучшена поддержка азиатских языков (китайского, японского, корейского).

В названиях отдельных исполняемых файлов пакета OpenOffice.org прослеживается его происхождение от StarOffice. Весь пакет вызывается командой "soffice." Отдельные программы пакетов вызываются следующими командами:

Таблица 12.3. Названия программ пакета OpenOffice.org.

Программа	Команда
Текстовый процессор	swriter
Менеджер презентаций	simpres
Электронная таблица	scalc
Программа работы с графикой	sdraw
Программа обработки формул	smath

В настоящее время локализация пакета OpenOffice.org выполнена для 27 языков. Пакет стабильно работает в операционных системах Solaris, Linux (включая PPC Linux) и Windows. Перенос его на платформы FreeBSD, IRIX и Mac OS X находится в стадии завершения.

Это все были общие сведения. Пора уже рассказать немного о первых впечатлениях от работы с самой программой. Работал я с версией 641b, которая входит в состав дистрибутива Red Hat Linux 7.2 Cyrillic Edition от "Урбан-софт".

Установка программы производится достаточно просто, – разворачиваем в одном из доступных каталогов архив и запускаем программу установки setup из полученного каталога install. Для начала можно выбрать стандартный вариант установки (есть еще варианты Minimal и Custom). Установка прошла у меня без всяких проблем, после чего в моем домашнем каталоге появился подкаталог OpenOffice.org641. В этом каталоге имеется символическая ссылка с именем soffice, которая позволяет запустить программу. Запускается по этой ссылке программа swriter, т. е. текстовый процессор. Остальные программы надо запускать по их собственным именам.

Внешний вид окна программы, конечно, несколько непривычен для пользователя MS Office. Кроме обычных полосок меню и панелей инструментов в верхней части окна имеется еще вертикальная панель инструментов слева. Но, в общем-то, все элементы окна достаточно понятны.

К сожалению, я познакомился с этой программой на последнем этапе работы над данной книгой, и у меня нет возможности дать подробное описание этого текстового процессора. Отмечу только самое существенное.

Для начала я, естественно попробовал просто вводить произвольный текст на русском языке. Никаких проблем с набором не возникло, но буквы почему-то наезжают друг на друга (проблема, о которой уже сообщалось применительно к StarOffice). Попытки изменить шрифт показали, что наиболее приемлемым является Helvetica. Текст стал в целом читабелен, но полностью эффект наезжающих букв не устранился. Вставка картинки (снимка окна программы) и таблицы прошли без проблем (при этом при создании таблицы сразу сформировалась ее шапка).

Далее я попытался сохранить набранный текст в формате MS Word. Для этого выбрал пункт **File / Save as**, и в выпадающем списке File Type – пункт Microsoft Word 97/2000/XP, ввел имя файла и щелкнул по кнопке **Save**. Перезагрузился в Windows и открыл сохраненный файл. Вначале картинка не появилась, однако, после того, как в меню **Вид** я выбрал пункт **Разметка страницы**, все встало на свои места. С таблицей тоже все нормально. Так что сохранение файла в формате Word работает.

Обратное действие – открытие в OpenOffice.org файла, созданного в MS Word – также прошло успешно. Правда, первоначально некоторые строки вывелись целиком квадратами вместо букв. Но, как оказалось, это следствие разницы в наборах шрифтов (или стилей) – после выделения соответствующей части текста и изменения шрифта текст становится вполне читаемым.

Между прочим, приведенное краткое описание пакета OpenOffice.org набрано в программе swriter.

Не имея сейчас возможности заниматься подробным описанием даже программы swriter, не говоря уже об остальных программах пакета, хочу только сформулировать первый вывод. Самое важное, на мой взгляд, то, что впервые появилась возможность под Linux работать с файлами, созданными в Microsoft Office. Тем самым уже нет необходимости запускать эмуляторы для того, чтобы просмотреть файл формата MS Word, полученный как вложение по электронной почте от любого из приверженцев Microsoft. И не беда, что пока есть претензии к качеству вывода изображения на экран. Все открытые продукты постоянно совершенствуются и развиваются и недостатки будут со временем устранены. Но появление OpenOffice.org разрывает последнюю нить, привязывавшую приверженцев Linux к продукции Microsoft, и позволяет им полностью перейти в среду Linux.

12.8. Словари и переводчики

Лично я отрицательно отношусь к программам, которые предназначены для автоматического перевода текста с одного языка на другой. Как бы ни старались разработчики таких программ, но добиться сколь-нибудь приемлемого качества перевода им не удастся, и, думаю, в ближайшем будущем не удастся.

Созданные такими программами переводы могут просто до абсурда изменить смысл написанного. Поэтому сам я такими программами не пользуюсь и вам не советую.

Другое дело словари. Без словаря не могут, как мне кажется, обойтись даже профессиональные переводчики. Даже на родном языке мы не всегда понимаем значение некоторых слов, так что вынуждены пользоваться толковыми словарями и энциклопедиями. Так что иметь электронный словарь просто необходимо. И такие словари существуют, в частности для Linux. Естественно, что большинству пользователей в первую очередь требуется англо-русские словари, чтобы понимать пояснения к программам, которые в абсолютном большинстве случаев даются на английском.

Я не проводил сравнительного анализа электронных словарей, результаты которого мог бы вам здесь предложить. Первой мне попала программа `slovo`, разработанная Д.Анисимовым, к которой я уже привык, и она меня вполне устраивает. Эта программа использует широко известный англо-русский словарь Мюллера (хотя могут быть подключены и другие словари) и дает транскрипцию английских слов, так что в обычной практике ее вполне достаточно.

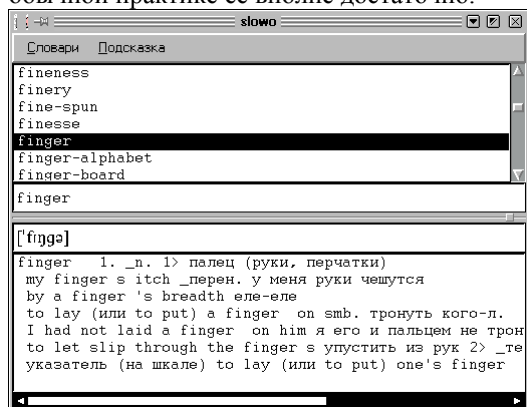


Рис. 12.26. Окно программы `slovo`

Некоторые затруднения у вас может вызвать получение транскрипции, поскольку для корректной работы этой части программы необходимо установить специальные шрифты (это естественно, обычные шрифты не содержат знаков транскрипции). Однако в руководстве к программе, которое прилагается и в русскоязычном варианте, необходимые действия подробно описаны, так что объяснять это здесь нет нужды. По крайней мере, у меня все получилось, думаю, что и у вас тоже все будет нормально, разве что придется приложить небольшие усилия.

После запуска программы появляется окно, изображенное на рис. 12.26, только первоначально три нижних поля пусты.

Вы можете выбрать требующее перевода слово из списка в верхнем поле, либо прямо ввести его в строке ввода (второе сверху поле). В процессе ввода происходит перемещение подсветки в списке слов на слово, ближайшее к введенной Вами комбинации символов. Нажатие на клавишу <Enter> приводит к появлению перевода в нижнем поле окна программы и транскрипции в поле, расположенном над переводом. Пункт меню **Словари** позволяет сменить словарь (не все словари позволяют выводить транскрипцию, в моем случае этим свойством обладал только словарь Мюллера).

Надеюсь, что приведенных пояснений вполне достаточно для того, чтобы вы смогли результативно пользоваться этой программой.

Еще один распространенный англо-русский словарь встретился мне в составе дистрибутива ALT Linux Junior 1.0. Это словарь **nova** Евгения Цимбалюка. Окно графической оболочки для этого словаря изображено на рис. 12.27.

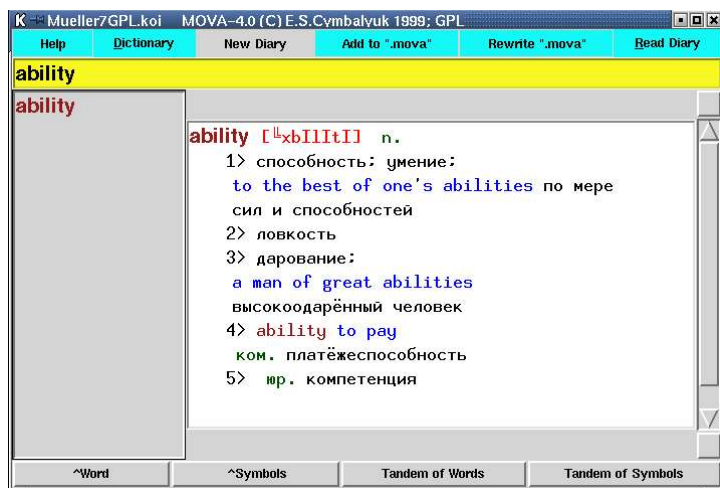


Рис. 12.27. Окно графической оболочки словаря mova

В строке ввода (она выделена желтым цветом) вводим английское слово и нажимаем клавишу "**^Word**". Можно просто скопировать неизвестное слово в строку ввода. Инструкции по использованию программы даны в прилагаемом к ней файле `Readme_mova_koi.txt`, причем на русском языке, так что здесь подробных пояснений давать не требуется.