ГЛАВА



В этой главе...

Редактор vim Редактор emacs Семейство редакторов KDE Редактор GNOME Резюме

Работа с редакторами

Прежде чем приступать к освоению работы по написанию сценариев для командного интерпретатора, необходимо узнать, как использовать по крайней мере один текстовый редактор в Linux. Чем больше вам будет известно о нюансах применения всех привлекательных средств редакторов, таких как поиск, вырезка и вставка текста, тем в большей степени вы будете подготовлены к решению задач разработки сценариев командного интерпретатора. В настоящей главе рассматриваются основные текстовые редакторы, которые могут встретиться в мире Linux.

Редактор vim

При работе в режиме командной строки часто возникает необходимость применения средств хотя бы одного текстового редактора, который работает на консоли Linux. Одним из редакторов, который применялся в системах Unix с самых первых версий, является редактор vi. В нем используется графический режим консоли для эмуляции окна редактирования текста, что позволяет просматривать строки файла, переходить из одного места файла в другое, а также вставлять, редактировать и заменять текст.

Возможно, правы те, кто считает vi самым сложным редактором в мире (по крайней мере, такого мнения придерживаются люди, которые его буквально ненавидят), но этот редактор предоставляет так много средств, что заслуженно играет роль основного инструмента для администраторов Unix в течение многих десятилетий.

Разработчики проекта GNU, перенося редактор vi в мир программного обеспечения с открытым исходным кодом, решили внести в него некоторые усовершенствования. В связи с этим в новой версии появились значительные отличия по сравнению с исходным редактором vi, применяющимся в мире Unix, поэтому разработчики переименовали свою версию в "vi improved", или сокращенно *vim*.

Для упрощения работы почти во всех дистрибутивах Linux создаются псевдонимы (см. главу 5) для этого редактора, обозначаемые как vi, которые указывают на программу vim:

```
$ alias vi
alias vi='vim'
s
```

В настоящем разделе приведены основные сведения об использовании редактора vim для внесения изменений в текстовые файлы сценариев командного интерпретатора.

Основные сведения о редакторе vim

Редактор vim работает с данными в буфере, находящемся в памяти. Для запуска редактора vim достаточно ввести команду vim (или vi, если определен псевдоним) и имя файла, в который необходимо внести изменения:

\$ vim myprog.c

Если запуск vim происходит без указания имени файла или если указанный файл не существует, то vim открывает новую буферную область для редактирования. Если же в командной строке указан существующий файл, то vim считывает все содержимое файла в буферную область и подготавливает файл для редактирования, как показано на рис. 9.1.

```
    rich@rich-desktop: ~
File Edit View Terminal Help

include <stdio.h>

int main() red
{
    int i;
    int factorial = 1;
    int number = 5;
    for(i = 1; i <= number; i++)
    {
        factorial = factorial * i;
    }
    printf("The factorial of %d is %d\n", number, factorial);
    return 0;
}
"myprog.c" 16 lines, 237 characters
</pre>
```

Рис. 9.1. Главное окно vim

Редактор vim распознает тип терминала, используемый для проведения сеанса (см. главу 2), и переходит в полноэкранный режим, что позволяет работать во всем окне терминала как в области редактирования.

В начальном окне редактирования vim отображается содержимое файла (если таковое имеется) наряду со строкой сообщения, расположенной вдоль нижнего края окна. Если содер-

Глава 9. Работа с редакторами

жимое файла занимает не весь экран, редактор vim обозначает знаками тильды (~) те строки в окне, которые не относятся к файлу (что можно видеть на рис. 9.1).

В строке сообщения в нижней части окна приведены сведения о редактируемом файле, которые изменяются в зависимости от статуса файла, и показаны параметры конфигурации, применяемые по умолчанию в данной конкретной инсталляции vim. Если файл новый, появляется сообщение [New File].

Редактор vim имеет два режима работы:

- нормальный режим;
- режим вставки.

Сразу после открытия файла для редактирования (или после начала работы с новым файлом) редактор vim переходит в *нормальный* режим, в котором он интерпретирует коды введенных комбинаций клавиш как команды (более подробные сведения об этом приведены ниже).

А в режиме *вставки* редактор vim вставляет символ, соответствующий каждой нажатой клавише, в текущем местоположении курсора в буфере, показанном на экране. Для перехода в режим вставки следует нажать клавишу <i>. Чтобы выйти из режима вставки и возвратиться в нормальный режим, необходимо нажать клавишу <Esc> на клавиатуре.

В нормальном режиме можно перемещать курсор в текстовой области с использованием клавиш со стрелками (такая возможность предоставляется, только если применяемый тип терминала распознан редактором vim должным образом). Если же оказалось, что установленное терминальное соединение не поддерживает все необходимые функции и клавиши со стрелками не определены, то это не означает, что все потеряно. В число команд vim входят команды перемещения курсора, включая следующие:

- команда h, предназначенная для перемещения влево на один символ;
- команда ј для перемещения вниз на одну строку (на следующую строку в тексте);
- команда k, позволяющая перейти вверх на одну строку (на предыдущую строку в тексте);
- команда 1, обеспечивающая переход вправо на один символ.

Безусловно, построчное перемещение курсора в больших текстовых файлах является слишком непродуктивным. К счастью, в редакторе vim предусмотрено несколько команд, позволяющих ускорить переход по содержимому файла:

- команда PageDown (или <Ctrl+F>), которая указывает, что должен быть выполнен переход вперед на один экран с данными;
- команда PageUp (или <Ctrl+B>), предназначенная для перехода назад на один экран с данными;
- команда G для перемещения к последней строке в буфере;
- команда *num* G, позволяющая перейти к строке с номером *num* в буфере;
- команда gg, обеспечивающая переход к первой строке в буфере.

Кроме того, в нормальном режиме редактор vim предоставляет доступ к специальной функции, называемой *режимом командной строки*. В режиме командной строки пользователю предоставляется интерактивная командная строка, в которой он может вводить дополнительные команды для управления другими действиями в редакторе vim. Для перехода в режим командной строки следует нажать клавишу со знаком двоеточия в нормальном режиме. Курсор перемещается в строку сообщения, и появляется двоеточие как приглашение к вводу команды. В число команд командной строки входят следующие несколько команд, предназначенных для сохранения буфера в файл и выхода из программы vim:

- команда q, позволяющая сразу же завершить работу, если не было внесено никаких изменений в данные в буфере;
- команда q!, с помощью которой можно завершить работу и отказаться от каких-либо изменений, внесенных в данные буфера;
- команда w filename, предназначенная для сохранения файла под другим именем файла;
- команда wq, обеспечивающая сохранение данных буфера в файле и завершение работы.

После ознакомления только с этими несколькими основными командами vim вполне можно понять, почему некоторые люди испытывают к редактору vim непримиримую ненависть. Дело в том, что для полноценного использования vim необходимо изучить большое количество непонятных, трудно запоминающихся команд. Однако, хорошо освоив даже небольшое количество основных команд vim, вы получите возможность быстро редактировать файлы непосредственно из командной строки, независимо от того, в какой среде работаете. Кроме того, после получения устойчивых навыков ввода команд с клавиатуры привычка набирать и данные, и команды редактирования становится второй натурой, поэтому программисты даже чувствуют себя немного не в своей тарелке, когда снова приходится прибегать к использованию мыши!

Редактирование данных

В процессе работы в режиме вставки можно вставлять данные в буфер, но иногда приходится также добавлять или удалять данные после того, как эти данные уже были введены в буфер. В нормальном режиме редактора vim предусмотрено несколько команд редактирования данных в буфере. В табл. 9.1 перечислены некоторые общие команды редактирования для vim.

Команда	Описание
х	Удалить символ в текущей позиции курсора
dd	Удалить строку в текущей позиции курсора
dw	Удалить слово в текущей позиции курсора
d\$	Провести удаление до конца строки от текущей позиции курсора
J	Удалить символ обозначения конца строки, находящийся в конце строки в текущей позиции курсора (что приводит к соединению строк)
u	Отменить предыдущую команду редактирования
a	Добавить данные после текущей позиции курсора
A	Добавить данные до конца строки, начиная с текущей позиции курсора
r char	Заменить отдельный символ в текущей позиции курсора символом $char$
R text	Перезаписывать данные в текущей позиции курсора, заменяя текстом $text$, пока не будет нажата клавиша <escape></escape>

Таб	лина	9.1	. Ka	оманлі	ыре	лакти	рования	vim
140	JIJIGU			отланд	יסקיום	цикти	pobullin	

Некоторые команды редактирования позволяют также использовать числовой модификатор, который указывает кратность выполнения команды. Например, применение команды 2x приводит к удалению двух символов, начиная с текущей позиции курсора, а команда 5dd удаляет пять строк, начиная со строки, в которой в настоящее время находится курсор.

Глава 9. Работа с редакторами



Прибегая к использованию клавиши <Backspace> или <Delete> клавиатуры персонального компьютера в редакторе vim, необходимо соблюдать осторожность. Редактор vim обычно распознает нажатие клавиши <Delete> как команду, функционально эквивалентную команде x, которая удаляет символ в текущем местоположении курсора. С другой стороны, обычно редактор vim не распознает нажатие клавиши <Backspace>.

Копирование и вставка

Стандартным средством современных редакторов является возможность вырезать или копировать данные, а затем вставлять их в другое место в документе. Редактор vim предоставляет способ выполнения этой операции.

Задача вырезки и вставки решается относительно просто. В табл. 9.1 уже были показаны команды, которые позволяют удалять данные из буфера. Однако редактор vim при удалении данных фактически не уничтожает эти данные, а передает на хранение в отдельный регистр. Затем удаленные данные можно получить с помощью команды р.

Например, можно удалить строку текста с помощью команды dd, затем переместить курсор в то место в буфере, в котором снова должна появиться эта строка, после чего применить команду р. Команда р вставляет текст после строки в текущей позиции курсора. Аналогичную операцию можно проделать с помощью любой команды, которая удаляет текст.

Задача копирования текста немного сложнее. Командой копирования в редакторе vim является у (сокращение от yank — копирование в память). С командой у можно использовать такой же второй символ, как и с командой d (например, вариант команды уw означает, что в память должно быть скопировано слово, а команда у\$ указывает, что в память должен быть скопирован текст до конца строки). После копирования текста в память можно перевести курсор к тому месту, где должен быть помещен текст, после чего задать команду p. Теперь скопированный в память текст появится в новом месте.

Сложность применения копирования в память заключается в том, что нельзя следить за происходящим, поскольку копирование текста не затрагивает сам текст, скопированный в память. Вы не знаете наверняка, что было скопировано в память, пока не вставите полученную копию где-нибудь в другом месте. Однако в редакторе vim предусмотрено еще одно средство, которое упрощает применение операции копирования в память.

Это — визуальный режим, предусматривающий выделение текста подсветкой при перемещении курсора. Визуальный режим можно использовать, выбирая текст, который должен быть скопирован в память для последующей вставки. Чтобы перейти в визуальный режим, переместите курсор в то место, где должно начаться копирование в память, и введите команду v. Вы заметите, что текст, начиная с текущей позиции курсора, теперь выделяется подсветкой. После этого перемещайте курсор, чтобы охватить текст, который должен быть скопирован в память (можно даже переходить на нижние строки, чтобы скопировать в память больше чем одну строку текста). По мере перемещения курсора редактор vim выделяет подсветкой текст в области копирования в память. Вслед за окончанием выделения подсветкой текста, который должен быть скопирован, нажмите клавишу <y> для вызова команды копирования в память. Теперь, когда текст отправлен в регистр, достаточно переместить курсор в то место, где он должен быть вставлен, и ввести команду p.

Поиск и замена

Команда поиска vim позволяет легко проводить поиск данных в буфере. Чтобы ввести строку для поиска, нажмите клавишу косой черты (</>). Курсор перейдет в строку сообщения,

и редактор vim выведет косую черту. Наберите текст, который должен быть найден, и нажмите клавишу <ВВОД>. Редактор vim в ответ выполнит одно из трех действий.

- Если искомое слово находится дальше в буфере по сравнению с текущим местоположением курсора, то происходит переход к первому ближайшему местоположению, где находится такой же текст.
- Если искомое слово не расположено вслед за текущим местоположением курсора, то после достижения конца файла происходит переход к первому местоположению в файле, где имеется текст (на что указывает также сообщение).
- Если поиск до конца файла или с начала файла оканчивается неудачей, то выводится сообщение об ошибке с указанием, что текст не найден в файле.

Чтобы продолжить поиск того же слова, нажмите клавишу косой черты, а затем — клавишу <ВВОД>; для продолжения поиска можно также ввести команду n (сокращение от next — далее).

Команда замены позволяет быстро заменять одно слово другим в тексте (подставлять вместо одного слова другое). Чтобы воспользоваться командой замены, необходимо перейти в режим командной строки. Команда замены имеет следующий формат:

:s/old/new/

Редактор vim переходит к первому вхождению текста old и заменяет его текстом new. Команда замены имеет несколько показанных ниже модификаций, которые позволяют заменять больше одного вхождения текста.

- Команда :s/old/new/g обеспечивает замену всех вхождений old в строке.
- Команда : n, ms/old/new/g указывает, что должны быть заменены все вхождения old в строках с номерами от n до m.
- Команда :%s/old/new/g позволяет заменить все вхождения old во всем файле.
- Команда :%s/old/new/gc требует заменить все вхождения old во всем файле, но выводить приглашения для подтверждения каждой замены.

Вполне очевидно, что vim, в отличие от других текстовых редакторов с командной строкой, содержит довольно много дополнительных функций. Редактор vim включен в каждый дистрибутив Linux, поэтому рекомендуется овладеть по меньшей мере основами работы с ним, чтобы всегда иметь возможность вносить изменения в сценарии, независимо от того, какие задачи вы решаете или какие еще редакторы имеете в своем распоряжении.

Редактор emacs

Редактор emacs — чрезвычайно популярный текстовый редактор, который был разработан еще до создания операционной системы Unix. Программисты в свое время настолько привыкли к этому редактору, что перенесли его в среду Unix, а теперь он перенесен также в среду Linux. Редактор emacs начал свое существование как редактор текста на терминале, во многом аналогично vi, но уже довольно давно был заменен графической версией.

Редактор етасs все еще обеспечивает работу в таком же режиме, как и его предшественник, предназначенный для терминала, но в настоящее время предоставляет также возможность использовать графическое окно X Window для редактирования текста в графической среде. Как правило, после запуска редактора етасs из командной строки программа редактора определяет, имеется ли доступный сеанс X Window, и в случае положительного ответа запускается

Глава 9. Работа с редакторами

в графическом режиме. Если таковой сеанс не существует, редактор запускается в терминальном режиме.

В настоящем разделе приведено описание редакторов emacs обоих типов, и для терминального, и для графического режима, чтобы читатель мог узнать, как использовать тот и другой, если есть такое желание (или необходимость).

Работа с редактором emacs на терминале

Версию редактора етасs для терминального режима можно рассматривать как еще один редактор, в котором предусмотрено большое количество клавиатурных команд, позволяющих выполнять функции редактирования. В редакторе emacs используются комбинации клавиш, которые включают клавишу <Control> (клавиша <Ctrl> на клавиатуре персонального компьютера) и клавишу <Meta>. В большинстве пакетов эмуляции терминала персонального компьютера клавиша <Meta> отображается на клавишу <Alt> персонального компьютера. В официально утвержденной документации по emacs клавиша <Ctrl> сокращенно обозначается как <C->, а клавиша <Meta> — как <M->. Таким образом, для указания на то, что нужно ввести комбинацию клавиш <Ctrl-х>, в документах приведено сокращение <C-х>. В данной главе применяется тот же принцип, чтобы не возникала путаница.

Основы emacs

Чтобы приступить к редактированию файла с помощью emacs, введите в командной строке следующую команду:

```
$ emacs myprog.c
```

Откроется окно emacs для работы в терминальном режиме с кратким введением и экраном справки. Не следует беспокоиться по поводу того, что текст выбранного вами файла не появился на экране; как только вы нажмете любую клавишу, редактор emacs загрузит файл в активный буфер и отобразит текст, как показано на рис. 9.2.



Рис. 9.2. Редактирование файла с использованием редактора етася в терминальном режиме

Часть І. Командная строка Linux

Можно видеть, что в верхней части окна терминального режима отображается типичная строка меню. К сожалению, эта строка меню предназначена для использования не в терминальном режиме, а лишь в графическом.



Если у вас имеется графический рабочий стол, но вы предпочитаете использовать emacs в терминальном режиме, а не в режиме X Window, то дополнительно задайте параметр – nw в командной строке.

В отличие от редактора vim, в котором приходится входить и выходить из режима вставки для переключения между вводом команд и вставкой текста, в редакторе emacs имеется только один режим. После ввода любого символа, который может быть выведен на печать, редактор emacs вставляет его в текущей позиции курсора. А после ввода команды emacs выполняет эту команду.

Для перемещения курсора в области буфера можно использовать клавиши со стрелками, а также клавиши <PageUp> и <PageDown>. Эта возможность предоставляется лишь при том условии, что программа редактора emacs правильно распознает применяемый эмулятор терминала. В противном случае для перемещения курсора можно воспользоваться командами, приведенными ниже.

- Команда С-р, предназначенная для продвижения вверх на одну строку (предыдущую строку в тексте).
- Команда С-b, позволяющая перейти влево (назад) на один символ.
- Команда C-f, с помощью которой можно перейти вправо (вперед) на один символ.
- Команда С-п, обеспечивающая перемещение вниз на одну строку (следующую строку в тексте).

Предусмотрены также следующие команды, предназначенные для перемещения курсора в тексте на более значительные расстояния.

- Команда M-f позволяет перейти вправо (вперед) к следующему слову.
- Команда M-b перемещает курсор влево (назад) к предыдущему слову.
- Команда С-а перемещает курсор в начало текущей строки.
- Команда С-е позволяет перейти в конец текущей строки.
- Команда М-а обеспечивает перемещение в начало текущего предложения.
- Команда М-е перемещает курсор в конец текущего предложения.
- Команда M-v обеспечивает переход назад на один экран данных.
- Команда C-v позволяет перейти вперед на один экран данных.
- Команда М-< указывает, что должен быть выполнено перемещение на первую строку текста.
- Команда M-> дает возможность перейти в последнюю строку текста.

Необходимо также освоить несколько следующих команд, с помощью которых можно снова сохранить буфер редактора в файле и выйти из программы emacs.

- Команда C-х C-s служит для сохранения текущего содержимого буфера в файл.
- Команда С-z позволяет выйти из редактора emacs, но оставить его работающим в текущем ceance, чтобы можно было снова к нему вернуться.
- Команда С-х С-с указывает, что нужно выйти из emacs и остановить эту программу.

Глава 9. Работа с редакторами

Заслуживает внимания то, что для двух из этих средств требуются по две клавиатурные команды. Команда С-х называется *расширенной командой*. Применение этой команды лежит в основе создания еще одного крупного набора команд, предназначенных для работы.

Редактирование данных

Редактор етася довольно удачно справляется с операциями вставки и удаления текста в буфере. Чтобы вставить текст, переместите курсор в то место, где должен быть вставлен текст, и приступите к вводу текста. Для удаления текста в редакторе етася предназначены клавиша <Backspace>, позволяющая удалить символ перед текущей позицией курсора, и клавиша <Delete>, которая удаляет символ в текущем местоположении курсора.

В редакторе emacs предусмотрены также команды уничтожения текста. Разница между удалением и уничтожением текста состоит в том, что текст после уничтожения помещается редактором emacs во временной области, из которой его можно извлечь (см. раздел "Копирование и вставка"). А удаленный текст исчезает навсегда.

Для уничтожения текста в буфере предусмотрено несколько команд.

- Команда М-Васкspace позволяет уничтожить слово перед текущей позицией курсора.
- Команда М-d служит для уничтожения слова после текущей позиции курсора.
- Команда С-к применяется для уничтожения текста от текущей позиции курсора до конца строки.
- Команда М-к предназначена для уничтожения текста от текущей позиции курсора до конца предложения.

Кроме того, в редакторе emacs предусмотрен привлекательный способ уничтожения большого объема текста. Для этого достаточно переместить курсор в начало области, в которой должен быть уничтожен текст, и нажать клавиши <C-@> или <C-ПРОБЕЛ>. После этого необходимо переместить курсор в конец области, где находится уничтожаемый текст, и нажать командные клавиши <C-w>. Весь текст между этими двумя местоположениями будет уничтожен.

Если окажется так, что уничтожение текста произошло по ошибке, то можно воспользоваться командой С-и, которая отменяет действие команды уничтожения и возвращает данные в то состояние, в котором они находились до уничтожения.

Копирование и вставка

Выше было показано, как вырезать данные в области буфера emacs, а в данном разделе речь пойдет о том, как вставить эти данные в другом месте. К сожалению, у тех, кто привык работать с редактором vim, могут возникать недоразумения после перехода к выполнению этого процесса с использованием редактора emacs.

Этому способствует такое неудачное совпадение, что в редакторе emacs *копированием в па-мять* (yanking) называется вставка данных. А в редакторе vim термином *yanking* обозначается непосредственно копирование в память, и об этом приходится постоянно помнить, если в ходе работы используются оба редактора.

После уничтожения данных с применением одной из команд уничтожения переместите курсор в то местоположение, где должны быть вставлены данные, и задайте команду С-у. Это приводит к копированию в память текста из временной области и вставке его в текущей позиции курсора. Команда С-у копирует в память текст, сохраненный во временной области после выполнения последней команды уничтожения. Если же было выполнено несколько команд уничтожения, то можно циклически переходить по полученным с их помощью данным с использованием команды М-у.

Часть I. Командная строка Linux

Чтобы выполнить копирование и вставку текста, достаточно лишь скопировать его назад в память в том же местоположении, в котором он был уничтожен, а затем перейти в новое местоположение и снова воспользоваться командой С-у. Допускается выполнять копирование текста назад в память столько раз, сколько потребуется.

Поиск и замена

Поиск текста в редакторе emacs выполняется с помощью команд C-r и C-s. Команда C-s осуществляет прямой поиск в области буфера от текущей позиции курсора до конца буфера, а команда C-r выполняет обратный поиск в области буфера от текущей позиции курсора до начала буфера.

После ввода команды C-s или C-r в нижней строке экрана появляется приглашение к вводу информации с запросом ввести текст для поиска. Редактор етасs может выполнять операции поиска двух типов.

При инкрементном поиске редактор етасѕ выполняет поиск текста в оперативном режиме одновременно с тем, как происходит ввод искомого слова. После ввода первой буквы редактор выделяет подсветкой все вхождения этой буквы в буфере. А вслед за вводом второй буквы выделяются подсветкой все вхождения двухбуквенной комбинации в тексте, и такой ввод может продолжаться до тех пор, пока не будет найден искомый текст.

При *неинкрементном* поиске требуется нажать клавишу <BBOД> после вызова команды С-г или C-s. Это приводит к блокированию вывода запроса поиска в нижней строке текстовой области, что позволяет полностью ввести искомый текст перед началом поиска.

Чтобы заменить существующую текстовую строку новой текстовой строкой, необходимо воспользоваться командой М-х. Для выполнения этой команды требуется также задать текстовую команду с параметрами.

Текстовой командой является команда replace-string. После ввода этой команды нажмите клавишу <BBOД>, и редактор emacs выведет запрос для указания существующей текстовой строки, подлежащей замене. Вслед за вводом этой строки снова нажмите клавишу <BBOД>, и редактор emacs выведет запрос для указания новой строки, которая должна заменить существующий текст.

Использование буферов в редакторе emacs

Если в редакторе emacs развернуто несколько областей буферов, то он позволяет редактировать сразу несколько файлов. Предусмотрена возможность загружать в каждый буфер по одному файлу и переключаться с одного буфера на другой в ходе редактирования.

Чтобы загрузить новый файл в буфер, работая в редакторе emacs, воспользуйтесь последовательностью комбинаций клавиш <C-x C-f>. Это — режим поиска файла в редакторе emacs. В этом режиме открывается нижняя строка в окне и пользователю предоставляется возможность ввести имя файла, к редактированию которого он желает приступить. Если имя или местоположение файла не известно, достаточно нажать клавишу <BBOД>. Это приводит к вызову средства просмотра файла в окне редактирования, как показано на рис. 9.3.

С помощью этого окна можно перейти к файлу, который необходимо отредактировать. Чтобы перейти на более высокий уровень в каталоге, переведите курсор на запись с двойной точкой и нажмите клавишу <ВВОД>. Для перехода вниз по каталогу переведите курсор на запись каталога и нажмите клавишу <ВВОД>. После того как будет найден файл, предназначенный для редактирования, остается только нажать клавишу <ВВОД>, и редактор emacs загрузит его в новую область буфера.

🛯 🛇 🔗 rich@rich-deskto	p: ~		
File Edit View Terminal Help			
File Edit Options Buffers	Tools Operate Mar	k Regexp Immediate Subdir Help	
/home/rich:			
total used in directory	544 available 563	10608	
drwxr-xr-x 37 rich rich	4096 2010-10-07	19:38 .	
drwxr-xr-x 4 root root	4096 2010-09-23	19:01	
drwx 2 rich rich	4096 2010-09-22	12:10 .aptitude	
-rw 1 rich rich	2831 2010-09-27	20:32 .bash_history	
-rw-rr 1 rich rich	220 2010-05-10	18:36 .bash_logout	
-rw-rr 1 rich rich	3103 2010-05-10	18:36 .bashrc	
drwx 5 rich rich	4096 2010-10-07	19:27 .cache	
drwx 3 rich rich	4096 2010-09-11	18:34 .compiz	
drwxr-xr-x 10 rich rich	4096 2010-09-11	18:32 .config	
drwx 3 rich rich	4096 2010-05-10	18:47 .dbus	
drwxr-xr-x 2 rich rich	4096 2010-09-22	12:10 .debtags	
drwxr-xr-x 2 rich rich	4096 2010-09-11	18:31 Desktop	
-rw-rr 1 rich rich	41 2010-10-07	19:27 .dmrc	
drwxr-xr-x 2 rich rich	4096 2010-05-10	18:47 Documents	
drwxr-xr-x 2 rich rich	4096 2010-09-22	13:46 Downloads	
drwxr-xr-x 3 rich rich	4096 2010-10-07	19:38 .emacs.d	
-rw 1 rich rich	16 2010-05-10	18:47 .esd auth	
-rw-rr 1 rich rich	179 2010-05-10	18:36 examples.desktop	
drwxr-xr-x 2 rich rich	4096 2010-06-28	20:09 .fontconfig	
-UUU:%%F1 ~	Top L5	(Dired by name)	
			∇

Рис. 9.3. Браузер режима поиска файла, применяемый в редакторе етася

Предусмотрена возможность просматривать список активных областей буферов, нажимая комбинацию расширенной команды С-х С-b. Редактор emacs разбивает окно редактора на две части и отображает список буферов в нижней части окна. В дополнение к главному буферу редактирования редактор emacs всегда предоставляет еще два таких буфера:

- рабочая область, обозначаемая как *scratch*;
- область сообщений с обозначением *Messages*.

Рабочая область позволяет вводить команды языка программирования LISP, а также оставлять для себя заметки. В области сообщений отображаются сообщения, формируемые редактором emacs в процессе работы. Если в ходе выполнения программы emacs возникают какиелибо ошибки, они отображаются в области сообщений.

Предусмотрены следующие два способа переключения на другую область буфера в окне.

- Команда С-х о позволяет переключиться на окно со списком буферов. После этого можно переместиться к требуемой области буфера с помощью клавиш со стрелками и клавишу <ВВОД>.
- Команда С-х b предусматривает ввод имени области буфера, на которую необходимо переключиться.

Если будет выбран вариант с переключением с помощью окна со списком буферов, то редактор emacs откроет область буфера в новой области окна. Редактор emacs позволяет открывать несколько окон в одном ceance. В следующем разделе показано, как управлять несколькими окнами в редакторе emacs.

Использование окон в редакторе emacs, работающем в терминальном режиме

Редактор етася для работы в терминальном режиме был разработан задолго до того, как появилась идея применения графических окон. Но в то время такое нововведение представ-

ляло собой большой шаг вперед, поскольку обеспечивалась возможность поддерживать одновременно несколько окон редактирования с помощью основного окна emacs.

Предусмотрена возможность разбивать окно редактирования етасѕ на несколько окон с помощью одной из следующих двух команд.

- Команда С-х 2 разбивает окно по горизонтали на два окна.
- Команда С-х 3 разбивает окно по вертикали на два окна.

Для перехода от одного окна к другому служит команда C-х о. Следует отметить, что после создания нового окна редактор етасѕ использует в этом новом окне область буфера из исходного окна. После перехода в новое окно можно воспользоваться командой C-х C-f для загрузки нового файла или вызвать одну из тех команд, которые предназначены для переключения на другую область буфера в новом окне.

Чтобы закрыть окно, необходимо перейти к нему и вызвать на выполнение команду C-х 0 (0 означает нуль). Если потребуется закрыть все окна, кроме того, в котором вы работаете, воспользуйтесь командой C-х 1 (1 — это единица).

Использование редактора emacs в среде X Window

После вызова на выполнение редактора emacs в среде X Window (такой как рабочий стол KDE или GNOME) его запуск происходит в графическом режиме, как показано на рис. 9.4.



Рис. 9.4. Графическое окно редактора emacs

Глава 9. Работа с редакторами

После успешного овладения приемами работы с редактором emacs в терминальном режиме не составляет особого труда перейти к работе в режиме X Window. Все клавиатурные команды становятся доступными как элементы строки меню. Строка меню emacs содержит следующие элементы.

- File (Файл). Этот элемент меню позволяет открывать файлы в окнах, создавать новые окна, закрывать окна, сохранять буфера и выводить содержимое буферов на печать.
- Edit (Редактирование). Позволяет вырезать и копировать выбранный текст в буфер обмена, вставлять данные буфера обмена в текущей позиции курсора, выполнять поиск и замену текста.
- Options (Параметры). Предоставляет настройки для многих других средств emacs, таких как выделение подсветкой, перенос по словам, определение типа курсора и задание шрифтов.
- Buffers (Буфера). Позволяет сформировать списки текущих доступных буферов и легко переключаться между областями буферов.
- Tools (Инструменты). Обеспечивает доступ к таким дополнительным функциям emacs, как доступ к интерфейсу командной строки, проверка орфографии, сравнение текста в разных файлах (что принято называть поиском различий), отправка сообщений электронной почты, работа с календарем и калькулятором.
- Help (Справка). Предоставляет доступ к оперативному руководству по етасs, что позволяет легко получить справку по конкретным функциям етасs.

В дополнение к обычным графическим элементам строки меню emacs часто встречается еще один отдельный элемент, характерный для того типа файла, который находится в буфере редактора. На рис. 9.4 показано, что в окне открыта программа на языке С, поэтому в редакторе еmacs предусмотрен элемент меню С, позволяющий применять дополнительные настройки для выделения подсветкой синтаксиса С, а также компилировать, выполнять и отлаживать код из командной строки.

Графическое окно етасs может служить примером того, как происходил перенос более старых приложений для консоли в мир графических средств. В настоящее время, когда во многих дистрибутивах Linux предусмотрены графические рабочие столы (даже на серверах, где они фактически не требуются), графические редакторы все чаще применяются вместо редакторов всех прочих типов. В обеих популярных версиях среды рабочего стола Linux (KDE и GNOME) также предусмотрены графические текстовые редакторы, специально предназначенные для той или иной среды, которые рассматриваются в остальной части этой главы.

Семейство редакторов КDE

Перед пользователями дистрибутивов Linux, в которых применяется рабочий стол KDE (см. главу 1), открывается широкий выбор, если возникает необходимость выполнить работу в текстовом редакторе. В проекте KDE официально поддерживаются два текстовых редактора.

- Kwrite. Пакет редактирования текста с применением одного экрана.
- Каtе. Полнофункциональный, многоэкранный пакет редактирования текста.

Оба эти редактора представляют собой графические текстовые редакторы, которые содержат много дополнительных функций. Редактор Каte предоставляет более развитые функции, а также позволяет воспользоваться дополнительными, привлекательными средствами, которые не часто встречаются в стандартных текстовых редакторах. В настоящем разделе описаны каждый из этих редакторов, а также некоторые средства, которые можно использовать для упрощения работы в области редактирования сценариев командного интерпретатора.

Редактор KWrite

Основным редактором для среды KDE является KWrite. Он предоставляет простые возможности редактирования текста, такие как редактирование с учетом стилей, а также поддерживает выделение цветом синтаксических конструкций в коде и редактирование кода. Применяемое по умолчанию окно редактирования KWrite показано на рис. 9.5.



Рис. 9.5. Применяемое по умолчанию окно KWrite, в котором редактируется программа сценария командного интерпретатора

Об этом нельзя судить на основании рис. 9.5, но редактор KWrite распознает несколько типов языков программирования и позволяет использовать цветовую разметку для проведения различий между константами, функциями, комментариями и т.д. Кроме того, следует отметить, что для оператора цикла for предусмотрен значок, который связывает открывающие и закрывающие фигурные скобки. Этот значок называется *маркером сворачивания*. Щелкнув на подобном значке, можно свернуть целую функцию в одну строку. Это средство становится очень полезным, когда приходится работать с большими приложениями.

Окно редактирования KWrite предоставляет все возможности вырезки и вставки с использованием мыши и клавиш со стрелками. Как и любой текстовый процессор, этот редактор позволяет выделять подсветкой, вырезать (или копировать) текст в любом месте области буфера и вставлять его в любом другом месте.

Чтобы приступить к редактированию файла с использованием KWrite, можно либо выбрать KWrite в системе меню KDE на рабочем столе (некоторые дистрибутивы Linux даже создают для этого редактора значок на панели), либо запустить программу редактора в приглашении командной строки:

Глава 9. Работа с редакторами

\$ kwrite factorial.sh

Команда kwrite имеет несколько описанных ниже параметров командной строки, которые можно использовать для настройки способа запуска редактора.

- Параметр --stdin вынуждает редактор KWrite читать данные со стандартного устройства ввода данных, а не из файла.
- Параметр -- encoding указывает, какой тип кодировки символов используется для файла.
- Параметр --line позволяет указать, с какого номера строки должен начаться вывод файла в окне редактора.
- Параметр --column указывает, с какого номера столбца начинается вывод файла в окне редактора.

В редакторе KWrite предусмотрены и строка меню, и панель инструментов в верхней части окна редактирования, что позволяет выбирать необходимые средства и вносить изменения в параметры конфигурации редактора KWrite.

Строка меню содержит следующие элементы.

- File (Файл). Позволяет загружать, сохранять, печатать и экспортировать текст из файлов.
- Edit (Правка). Дает возможность управлять текстом в области буфера.
- View (Представление). Управляет тем, как выглядит текст в окне редактора.
- Bookmarks (Закладки). Служит для обработки указателей, позволяющих возвращаться к конкретным местоположениям в тексте (может потребоваться разрешить использование этой функции с помощью параметров настройки).
- Tools (Инструменты). Содержит специализированные средства управления текстом.
- Settings (Настройки). Служит для определения способа обработки текста в редакторе.
- Неір (Справка). Предназначен для предоставления сведений о редакторе и его командах.

В элементе строки меню Edit предусмотрены команды, с помощью которых можно решать любые задачи редактирования текста. Поэтому достаточно лишь выбирать нужные элементы в строке меню Edit, как показано в табл. 9.2, не сталкиваясь с необходимостью запоминать загадочные клавиатурные команды (которые, кстати, также поддерживаются редактором KWrite).

Элемент	Описание
Undo (Отмена)	Выполняет по отношению к последнему действию или операции обратное действие или операцию
Redo (Повтор)	Выполняет действие, обратное по отношению к последнему действию отмены
Cut (Вырезка)	Удаляет выбранный текст и помещает его в буфер обмена
Сору (Копировать)	Копирует выбранный текст в буфер обмена
Copy as HTML (Копирование как HTML)	Копирует выбранный текст в буфер обмена как код HTML
Paste (Вставить)	Вставляет текущее содержимое буфера обмена в текущей позиции курсора
Select All (Выделить все)	Выделяет весь текст в редакторе
Deselect (Отменить выделение)	Отменяет выделение всего текста, который выбран в настоящее время

Таблица 9.2. Элементы меню Edit (Правка) редактора KWrite

Часть І. Командная строка Linux

Элемент	Описание
Overwrite Mode (Режим замены)	Переключает режим вставки на режим замены, в котором происходит замена текста вновь введенным текстом, а не вставка нового текста
Find (Найти)	Выводит диалоговое окно Find Text (Поиск текста), которое позволяет на- страивать текстовый поиск
Find Next (Найти следующее)	Повторяет последнюю операцию поиска в прямом направлении в области буфера
Find Previous (Найти предыдущее)	Повторяет последнюю операцию поиска в обратном направлении в области буфера
Replace (Заменить)	Выводит диалоговое окно Replace With (Заменить на), которое позволяет настраивать текстовый поиск и замену
Find Selected (Найти выбранное)	Находит следующее вхождение выбранного текста
Find Selected Backwards (Найти выбранное сзади)	Находит предыдущее вхождение выбранного текста
Go to Line (Перейти к строке)	Выводит диалоговое окно Goto (Перейти), которое позволяет ввести номер строки. Курсор перемещается к указанной строке

Средство Find (Поиск) имеет два режима: обычный режим, который позволяет выполнять простые операции текстового поиска, и дополненный режим поиска и замены, позволяющий в случае необходимости решать более сложные задачи поиска и замены. Для переключения между этими двумя режимами предназначена зеленая стрелка в разделе Find, как показано на рис. 9.6.

🗊 📀 factorial.sh – KWrite 💿 👘	\odot	×
<u>File Edit View Tools Settings H</u> elp		
Image: Save As Close Undo Redo		
<pre>#!/bin/bash factorial=1 number=5</pre>		Î
<pre>for ((i=1; i <= \$number; i++)) factorial=`expr \$factorial * \$i` } echo The factorial of \$number is \$factorial.</pre>		
	<>	
🛛 😽 Find:	Previous 🛛 🔻	F
Replace:	olace <u>A</u> ll	
Plain text 🗸 Match case Op	otions 🗸	
Line: 4 Col: 5 INS LINE Bash factorial.sh		

Рис. 9.6. Раздел Find редактора KWrite

Дополненный режим поиска позволяет задавать для поиска не только символьные строки, но и регулярные выражения (о чем речь пойдет в главе 19). Предусмотрены также некоторые другие параметры, которые можно использовать для настройки поиска, позволяющие, напри-

Глава 9. Работа с редакторами

мер, указать, должен ли проводиться поиск с учетом регистра и следует ли рассматривать при поиске лишь целые слова, а не произвольные строки.

Элемент строки меню Tools (Инструменты) предоставляет несколько удобных средств для работы с текстом в области буфера. Инструменты, предусмотренные в редакторе Kwrite, перечислены в табл. 9.3.

иолица э.о. иногрументы киние	
Инструмент	Описание
Read Only Mode (Режим доступа только для чтения)	Блокирует текст, чтобы во время работы в редакторе нельзя было вносить какие-либо изменения
Encoding (Кодировка)	Задает кодировку для набора символов, используемого в тексте
Spelling (Орфография)	Запускает программу проверки орфографии с начала текста
Spelling (from cursor) (Орфография (с позиции курсора))	Запускает программу проверки орфографии с текущей позиции курсора
Spellcheck Selection (Проверка орфо- графии в выбранном тексте)	Запускает программу проверки орфографии только на выбранном участке текста
Indent (Прямой отступ)	Увеличивает отступ абзаца на единицу
Unindent (Обратный отступ)	Уменьшает отступ абзаца на единицу
Clean Indentation (Очистка отступов)	Возвращает все отступы абзацев к исходным настройкам
Align (Выровнять)	Принудительно возвращает текущую строку или выбранные строки к заданным по умолчанию стандартным настройкам
Uppercase (Верхний регистр)	Преобразует символы в выбранном тексте или в текущей позиции курсора в символы верхнего регистра
Lowercase (Нижний регистр)	Преобразует символы в выбранном тексте или в текущей позиции курсора в символы нижнего регистра
Capitalize (Преобразовать в прописные)	Преобразовывает первую букву выбранного текста или слова в текущей позиции курсора в прописную
Join Lines (Соединить строки)	Объединяет в одну строку выбранные строки или строку в текущей позиции курсора и следующую строку
Word Wrap Document (Выполнить перенос по словам)	Обеспечивает перенос на новую строку в тексте. Если длина строки текста превышает ширину окна редактора, вывод строки текста про- должается на следующей строке экрана

Таблица 9.3. Инструменты KWrite

Очевидно, что этот с виду простой текстовый редактор предоставляет весьма широкий набор инструментов!

Меню Settings (Настройки) позволяет получить доступ к диалоговому окну Configure Editor (Настройка редактора), которое показано на рис. 9.7.

В левой части диалогового окна Configuration (Конфигурация) находятся значки, позволяющие пользователю выбрать средство KWrite, подлежащее настройке. После выбора значка в правой части диалогового окна отображаются значения параметров конфигурации для соответствующего средства.

Средство Appearance (Внешний вид) позволяет задавать некоторые функции, которые управляют отображением текста в окне текстового редактора. Здесь можно разрешить перенос по словам, ввести нумерацию строк (что очень удобно для программистов) и обозначить папки маркерами. Средство Fonts & Colors (Шрифты и цвета) позволяет настраивать полную цветовую схему для редактора, определяя, какие цвета должны применяться для обозначения всех категорий текстовых конструкций в коде программы.

🗐 💮 Configure – KW	frite	$? \odot \diamond \otimes \qquad \qquad$
	Appearance	\checkmark
Appearance	Dynamic Word Wrap	
1	Dynamic word wrap indicators (if applicable): Follow	Line Numbers 🗸
Fonts & Colors	Align dynamically wrapped lines to indentation depth:	of View Width
1		
Editing	Borders	
	Show <u>f</u> olding markers (if available)	
Open/Save	Show icon border	
	Show line numbers	
Futureirus	Show <u>s</u> crollbar marks	
Extensions		
	Advanced	
	Enable power user mode (KDE 3 mode)	
	Show indentation lines	
	Highlight range between selected brackets	
Help	C	✓ <u>O</u> K Ø Cancel

Рис. 9.7. Диалоговое окно Configure Editor редактора KWrite

Редактор Kate

Редактор Kate — ведущий редактор для проекта KDE. В нем используется такой же текстовый редактор, как и в приложении KWrite (поэтому аналогичным является и большинство предусмотренных в нем средств), но в тот же отдельный пакет включено много дополнительных средств.

Первое, что бросается в глаза после запуска редактора Kate, — диалоговое окно, показанное на рис. 9.8, а не само окно редактора!

🕐 💽 Session Chooser – Kate 💿 🚃	$\odot \odot \odot$	×
Session Name © Open Documents		:
Default Session 1		
Always use this choice		
Always use this choice		

Рис. 9.8. Диалоговое окно сеанса Каte

Глава 9. Работа с редакторами

Редактор Каte обрабатывает файлы в ceancax. Предусмотрена возможность открывать в ceансах несколько файлов, а также иметь несколько сохраненных ceancoв. После запуска редактор Каte предоставляет возможность выбрать ceanc, к которому следует возвратиться. После закрытия ceanca Kate редактор запоминает документы, которые были открыты, и отображает их при следующем запуске Kate. Это позволяет легко управлять файлами из нескольких проектов путем использования отдельных рабочих пространств для каждого проекта.

После выбора сеанса открывается основное окно редактирования Kate, которое показано на рис. 9.9.



Рис. 9.9. Основное окно редактирования Кате

В области окна слева отображаются документы, открытые в настоящее время в сеансе. Предусмотрена возможность переключаться с одного документа на другой, для чего достаточно щелкнуть на имени документа. Чтобы вызвать на редактирование новый файл, перейдите на вкладку Filesystem Browser (Обозреватель файловой системы), находящуюся слева. После этого в левой области окна открывается полноценный графический обозреватель файловой системы, который позволяет находить нужные файлы с помощью графического интерфейса.

Великолепным средством редактора Каte является встроенное окно терминала, которое показано на рис. 9.10.

Запуск встроенного эмулятора терминала в программе Каte осуществляется после перехода на вкладку с обозначением терминала в нижней части окна текстового редактора (применяется эмулятор терминала Konsole среды KDE). Вызов этого средства на выполнение приводит к разбиению текущего окна редактирования по горизонтали и созданию нового окна, в котором работает терминал Konsole. Это окно позволяет вводить команды командной строки, запускать программы или проверять параметры настройки системы, не выходя из редактора! Чтобы закрыть окно терминала, достаточно ввести команду exit в командной строке.

Как показывает это средство работы с терминалом, редактор Каte обладает также способностью поддерживать несколько окон. Элемент меню Window (Окно) предоставляет следующие возможности.

🗹 💿 Default Session: factorial.sh -	-Kate	\odot	×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>B</u> ookmar	ks Sess <u>i</u> ons <u>T</u> ools <u>S</u> ettings <u>H</u> elp		
New Open Back Forward	Save Save As Close Undo Redo		
factorial.c factorial.sh	<pre>#//bin/bash factorial=1 number=5 for ((i=1; i <= \$number; i++)) { factorial=`expr \$factorial * \$i` } echo The factorial of \$number is \$factorial.</pre>		*
	Line: 1 Col: 1 INS LINE factorial.sh	<	>
	rich@localhost:~/Documents> ./factorial.sh The factorial of 5 is 120. rich@localhost:~/Documents> ∎		
	Terminal 📣 Find in Files		

Рис. 9.10. Встроенное окно терминала Каte

- Создание нового окна Каtе с использованием текущего сеанса.
- Разбиение текущего окна по вертикали для создания нового окна.
- Разбиение текущего окна по горизонтали для создания нового окна.
- Закрытие текущего окна.

Чтобы задать значения параметров конфигурации в Kate, разверните элемент строки меню Settings (Настройки) и выберите элемент Configure Kate (Настройка Kate). Появится диалоговое окно Configuration (Конфигурация), которое показано на рис. 9.11.

Заслуживает внимания то, что область Editor settings (Настройки редактора) полностью совпадает с той же областью, которая относится к редактору KWrite. Это связано с тем, что в обоих этих редакторах применяется одно и то же ядро текстового редактора. Область Application settings (Параметры приложения) позволяет задавать настройки для различных элементов меню Kate, например, относящихся к управлению сеансами (как показано на рис. 9.11), формированию списка документов и применению обозревателя файловой системы. Кроме того, редактор Kate поддерживает внешние сменные приложения, которые могут быть активизированы в этой области.

Редактор GNOME

Пользователи, работающие в системе Linux с применением среды рабочего стола GNOME, могут также воспользоваться графическим текстовым редактором, предусмотренным в этой среде. Это — текстовый редактор gedit, представляющий собой довольно простой текстовый

редактор, в котором разработчики предусмотрели также несколько дополнительных функций, которые увеличивают его привлекательность. В настоящем разделе приведено краткое описание средств gedit и показано, как использовать этот редактор при программировании сценариев командного интерпретатора.



Рис. 9.11. Диалоговое окно со значениями параметров конфигурации Kate

Запуск программы gedit

В большинстве вариантов среды рабочего стола GNOME редактор gedit можно открыть, развернув элемент меню Accessories Panel (Панель вспомогательных программ). Если в этом меню редактор gedit отсутствует, его можно запустить из приглашения командной строки:

```
$ gedit factorial.sh myprog.c
```

После запуска программы gedit для редактирования нескольких файлов происходит загрузка всех этих файлов в отдельные буфера, а имя каждого файла отображается в заголовке одной из вкладок окна с вкладками в главном окне редактора (рис. 9.12).

В левой области главного окна редактора gedit отображаются имена документов, которые в настоящее время загружены в редактор. В правой области показано окно с вкладками, которое содержит текст из буфера. При проведении указателя мыши над каждой вкладкой появляется диалоговое окно, в котором показаны полное имя пути файла, тип MIME и кодировка набора символов, используемого в файле.



Рис. 9.12. Главное окно редактора gedit

Основные средства gedit

В дополнение к окнам редактора в программе gedit используются строка меню и панель инструментов, которые позволяют задавать применяемые средства и определять значения параметров. Панель инструментов предоставляет быстрый доступ к элементам строки меню. Предусмотренные в редакторе элементы строки меню перечислены ниже.

- File (Файл). Обеспечивает обработку новых файлов, сохранение существующих файлов и вывод содержимого файлов на печать.
- Edit (Правка). Дает возможность управлять текстом в активной области буфера и задавать параметры редактора.
- View (Вид). Предназначен для определения функций редактора, которые управляют отображением данных в окне, и задания режима выделения текста подсветкой.
- Search (Поиск). Служит для поиска и замены текста в активной области буфера редактора.
- Tools (Инструменты). Предназначен для получения доступа к сменным инструментам, установленным в редакторе gedit.
- Documents (Документы). Позволяет управлять файлами, открытыми в областях буферов.
- Help (Справка). Предоставляет доступ к полному руководству по программе gedit.

Очевидно, что в этом редакторе предусмотрен примерно такой же набор функций, как и в других редакторах. Меню File предоставляет доступ к параметру Open Location (Открыть доступ), который позволяет открывать файлы, полученные из сети с использованием стандартного формата URI (Uniform Resource Identifier — универсальный идентификатор ресурса), ко-

Глава 9. Работа с редакторами

торый широко применяется в Интернете. Этот формат позволяет указать протокол, используемый для получения доступа к файлу (такой как HTTP или FTP), сервер, на котором находится файл, и полный путь на сервере для получения доступа к файлу.

В меню Edit предусмотрены стандартные функции вырезки, копирования и вставки, а также еще одна удобная функция, которая позволяет легко вставлять в текст значения даты и времени в нескольких форматах. Меню Search предоставляет доступ к стандартной функции поиска, после вызова которой открывается диалоговое окно, в котором можно ввести искомый текст, а также задать параметры поиска (указать, должен ли учитываться регистр, проводиться поиск путем сопоставления с целым словом или строкой, и определить направление поиска). Предусмотрено также средство инкрементного поиска, которое работает в оперативном режиме, отыскивая текст одновременно с тем, как пользователь вводит символы искомой строки.

Задание параметров

Меню Edit содержит элемент Preferences (Параметры), после щелчка на котором открывается диалоговое окно gedit Preferences (Параметры gedit), которое показано на рис. 9.13.

🔍 🔿 ਨੇ facto	rial.sh (~) - gedit		
File Edit View S	gedit Preferences		
Documents factorial.sh myprog.c	 gedit Preferences View Editor Font & Colors Plugins Text Wrapping Enable text wrapping Do not split words over two lines Line Numbers Display line numbers 		k 9e
	Highlight current line		
	Right Margin Display right margin Right margin at column: 80 # Bracket Matching Highlight matching bracket		lal.
	Help	Close	INS

Рис. 9.13. Диалоговое окно gedit Preferences

Диалоговое окно gedit Preferences предназначено для настройки параметров, управляющих работой редактора gedit. Оно содержит пять областей с вкладками, предназначенных для определения средств и функций редактора, которые описаны ниже.

Вкладка View (Вид)

Позволяет определить параметры, управляющие тем, как программа gedit отображает текст в окне редактора.

- Вкладка Text Wrapping (Перенос текста по словам). Управляет отображением длинных строк текста в редакторе. После выбора параметра Enabling text wrapping (Разрешить перенос строк текста по словам) части строк текста, длина которых превышает ширину экрана, переносятся на следующие строки экрана редактора. Параметр Do Not Split Words Over Two Lines (Не выполнять разбивку слов на две строки) отменяет автоматическую вставку дефисов в длинных словах, в результате которой две части одного слова оказываются на двух строках экрана.
- Вкладка Line Numbers (Нумерация строк). С ее помощью можно предусмотреть отображение номеров строк в левой части окна редактора.
- Вкладка Current Line (Текущая строка). Позволяет предусмотреть выделение подсветкой строки, в которой в настоящее время находится курсор, тем самым упрощая поиск позиции курсора в тексте.
- Вкладка Right Margin (Правое поле). Дает возможность определить правое поле и задать количество столбцов в окне редактора. Значение по умолчанию составляет 80 столбцов.
- Вкладка Bracket Matching (Согласование скобок). С ее помощью можно включить функцию выделения подсветкой парных скобок в коде программы, что позволяет легче находить соответствующие друг другу скобки в операторах if-then, в циклах for и while и в других программных конструкциях, в которых применяются парные скобки.

Функции нумерации строк и согласования скобок создают для программистов такую удобную среду отладки кода, которая не часто встречается в текстовых редакторах.

Вкладка Editor (Редактор)

Позволяет задавать параметры, определяющие то, как ведется обработка в редакторе gedit знаков табуляции и отступов, а также указывать, как должно проводиться сохранение файлов.

- Параметр Tab Stops (Табуляторы). Указывает количество позиций, на которое должен сместиться курсор после нажатия клавиши <Tab>. Значение по умолчанию — восемь.
 В определение этой функции также включен флажок, который обеспечивает вставку соответствующего количества пробелов вместо одного знака табуляции.
- Параметр Automatic Indentation (Автоматический отступ). Если этот параметр включен, редактор gedit автоматически обозначает отступом все последующие строки текста в абзацах и программных конструкциях (таких как операторы if-then и циклы).
- Параметр File Saving (Сохранение файла). Позволяет воспользоваться двумя дополнительными возможностями сохранения файлов. Таковыми являются, во-первых, создание резервной копии файла после его открытия в окне редактирования и, во-вторых, автоматическое сохранение файла через заранее установленные промежутки времени.

Средство автоматического сохранения — это великолепный способ обеспечить сохранение вносимых изменений на регулярной основе. Это позволяет предотвратить случайную потерю результатов работы при аварийном отказе или прекращении подачи электроэнергии.

Вкладка Font & Colors (Шрифт и цвет)

Как и следует из ее названия, позволяет настраивать следующие два набора свойств текста.

 Элемент Font (Шрифт). Позволяет выбрать заданный по умолчанию шрифт Monospace 10 или указать определяемые пользователем шрифт и размер шрифта в диалоговом окне.

Глава 9. Работа с редакторами

Элемент Color Scheme (Цветовая схема). Позволяет выбрать цветовую схему, применяемую по умолчанию, для текста, фона, выбранного текста и так далее, или указать определяемый пользователем цвет для каждой категории.

Как правило, цвета, заданные по умолчанию для редактора gedit, согласуются со стандартной темой рабочего стола GNOME, выбранной в настоящее время. Если для рабочего стола будет принята другая тема, изменятся и эти цвета.

Вкладка Plugins (Подключаемые модули)

Предоставляет возможность настройки подключаемых модулей, применяемых в редакторе gedit. Подключаемые модули — это отдельные программы, способные взаимодействовать с программой gedit, предоставляя дополнительные функциональные средства. Вкладка Plugins показана на рис. 9.14.

🗵 🔿 🔿 factori	ial.sh (~) - gedit	
File Edit View S	aedit Preferences	
Open	View Editor Font & Colors Plugins	94
Documents	Active Plugins:	
factorial.sh 📓 myprog.c	Change Case Changes the case of selected text.	
	Document Statistics Analyzes the current document and re	
	External Tools Execute external commands and shell	
	File Browser Pane Easy file access from the side pane	
	Insert Date/Time Inserts current date and time at the cu	ial.
	Modelines Emacs, Kate and Vim-style modelines	
	Python Console Interactive Python console standing in	
	Ouick Open	
	About Plugin Configure Plugin	
	Help	INS

Рис. 9.14. Вкладка Plugins Preferences (Параметры подключаемых модулей) программы gedit

Для редактора gedit предусмотрен целый ряд подключаемых модулей, но не все они устанавливаются по умолчанию. В табл. 9.4 перечислены подключаемые модули, которые в настоящее время предусмотрены в редакторе gedit.

Таблица 9.4.	Подключаемые	модули gedi
--------------	--------------	-------------

Подключаемый модуль	Описание
Change Case (Смена регистра)	Преобразует регистр символов в выбранном тексте на противоположный
Document Statistics (Статистические данные документа)	Позволяет получить сведения о количестве слов, строк, символов и не-пробельных символов

Часть I. Командная строка Linux

Подключаемый модуль	Описание
External Tools (Внешние инструмен- тальные средства)	Предоставляет в редакторе среду командного интерпретатора, позволяющую выполнять команды и сценарии
File Browser Pane (Панель обозрева- теля файлов)	Предоставляет простое средство просмотра файлов, с применением ко- торого легче решается задача выбора файлов для редактирования
Insert Date/Time (Вставка даты и времени)	Позволяет вставить значения текущей даты и времени в нескольких разных форматах в текущей позиции курсора
Modelines (Моделайны — строки управления параметрами)	Предоставляет возможность открывать строки сообщений в стиле emacs в нижней части окна редактора
Python Console (Консоль Python)	Позволяет открыть в нижней части окна редактора интерактивную кон- соль для ввода команд на языке программирования Python
Quick Open (Быстрое открытие)	Открывает файлы непосредственно в окне редактирования gedit
Snippets (Фрагменты)	Обеспечивает возможность сохранения часто используемых фрагментов текста для упрощения их выборки и вставки в любом месте в тексте
Sort (Сортировка)	Быстро сортирует весь файл или выбранный текст
Spell Checker (Программа проверки орфографии)	Предоставляет словарь, с помощью которого можно проверить орфографию в текстовом файле
Tag List (Список тегов)	Предоставляет список часто используемых строк, которые можно легко вставлять в текст

Подключаемые модули, которые разрешены для использования, обозначены флажком рядом с именем. Некоторые подключаемые модули, такие как External Tools (Внешние инструментальные средства), предоставляют также дополнительные средства настройки после их выбора. Это позволяет задать комбинацию клавиш для запуска терминала, в котором gedit отображает вывод, а также команду запуска сеанса командного интерпретатора.

К сожалению, не все подключаемые модули устанавливаются в одном и том же месте в строке меню программы gedit. Одни из них появляются в элементе строки меню Tools (к ним относятся подключаемые модули Spell Checker и External Tools), тогда как другие находятся в элементе строки меню Edit (в частности, подключаемые модули Change Case и Insert Date/ Time).

Резюме

При создании сценариев командного интерпретатора приходится прибегать к использованию текстового редактора того или иного типа. Для среды Linux предусмотрен целый ряд широко применяемых текстовых редакторов. Редактор vi, наиболее широко применяемый в мире Unix, был перенесен в Linux под именем vim. Он позволяет решать простые задачи редактирования текста с помощью консоли, для чего служит полноэкранный графический режим, не предусматривающий применение достаточно развитых графических функций. Редактор vim предоставляет много дополнительных средств редактора, таких как поиск и замена текста.

Из среды Unix в мир Linux пробился еще один популярный редактор — emacs. В Linux предусмотрены две версии emacs, для консоли и для графического режима X Window, поэтому emacs может служить мостом между старыми и новыми технологиями редактирования текста. Редактор emacs позволяет использовать несколько областей буферов, что дает возможность редактировать одновременно несколько файлов.

Глава 9. Работа с редакторами

В рамках проекта KDE созданы два редактора, предназначенные для использования на рабочем столе KDE. Редактор KWrite — это простой редактор, предоставляющий основные средства редактирования текста, наряду с несколькими дополнительными функциями, такими как выделение цветом синтаксической конструкции в коде программы, нумерация строк и сворачивание кода. Редактор Kate предоставляет больше дополнительных функций для программистов. Одним из великолепных средств Kate является встроенное окно терминала. Пользователь может открыть сеанс работы с командной строкой непосредственно в редакторе Kate, не открывая отдельное окно эмулятора терминала. Кроме того, редактор Kate позволяет открывать несколько файлов, выводя содержимое каждого из них в отдельные окна.

В рамках проекта GNOME также предусмотрен простой текстовый редактор для программистов. Редактор gedit обладает основным набором функций текстового редактора, а также предоставляет некоторые дополнительные функции, такие как выделение цветом синтаксических конструкций в коде и нумерация строк, но в целом в задачу его разработчиков входило создание редактора, который сам по себе не отягощен ненужными функциями. Для расширения возможностей редактора gedit разработчики создали подключаемые модули, позволяющие предусмотреть для него дополнительные функции. В число предусмотренных в настоящее время подключаемых модулей входят программа проверки орфографии, эмулятор терминала и средство просмотра файлов.

Эта глава завершает ряд подготовительных глав, посвященных работе с командной строкой в Linux. В следующей части книги основное внимание уделено ознакомлению с возможностями сценариев для командного интерпретатора. Со следующей главы начинается рассмотрение вопросов создания файлов сценариев для командного интерпретатора и описания того, как вызывать эти файлы на выполнение в системе Linux. Кроме того, далее будут показаны основы сценариев для командного интерпретатора, что позволит читателю создавать простые программы, связывая друг с другом отдельные команды в сценарий, который может быть вызван на выполнение.