

Редактор vim

Редактор vim

[Общие сведения, команды выхода и сохранения](#)

[Редактирование текста](#)

[Копирование и вставка](#)

[Поиск и замена](#)

[Литература](#)

Общие сведения, команды выхода и сохранения

Для запуска редактора vim достаточно ввести команду vim (или vi, если определен псевдоним) и имя файла, в который необходимо внести изменения:

```
1 | user@linux-pc:~/bin$ vim user_info
```

Если файл не существует, на экране появится следующее:

```
1 | ~
2 | ~
3 | ~
4 | ~
5 | ~
6 | ~
7 | ~
8 | ~
9 | ~
10 | ~
11 | "new_file.txt" [New File]                                0,0-1      All
```

Начальные символы тильды (~) сообщают об отсутствии текста в соответствующих строках. Таким способом vim сообщает, что файл пуст. Если же в командной строке указан существующий файл, то vim считывает все содержимое файла в память и подготавливает файл для редактирования:

```
1 | #!/bin/bash
2 | # Отображение информации о пользователе
3 | echo "User info for $USER"
4 | echo "UID : $UID"
5 | echo "HOME: $HOME"
6 | ~
```

7	~		
8	~		
9	~		
10	~		
11	~		
12	~		
13	~		
14	~		
15	"user_info" 5L, 151C	1,1	All

В нижней части окна расположена строка сообщений. В строке сообщений приводятся сведения о редактируемом файле, которые изменяются в зависимости от статуса файла, и показываются параметры конфигурации, применяемые по умолчанию. Если файл новый, появляется сообщение [New File].

Редактор vim имеет три режима работы:

- нормальный режим;
- режим вставки;
- режим выделения.

Сразу после начала работы редактор vim переходит в нормальный режим, в котором он интерпретирует коды нажатых клавиш как команды. В режиме вставки редактор vim вставляет символ, соответствующий каждой нажатой клавише, в текущем местоположении курсора, 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.

Находясь в нормальном режиме, vim предлагает большое количество команд управления курсором (часть этих команд используется программой less). В таблице приведены некоторые из этих команд.

Клавиша	Перемещает курсор
l (маленькая L) или стрелка вправо	Вправо на один символ
h или стрелка влево	Влево на один символ
j или стрелка вниз	Вниз на одну строку
k или стрелка вверх	Вверх на одну строку
CTRL+f или PgDn	Вниз на одну страницу
CTRL+b или PgUp	Вверх на одну страницу
Число G	К строке с указанным номером (например, команда 1 G выполнит переход к первой строке файла)
gg	К первой строке файла
GG	К последней строке файла
0 (ноль)	В начало текущей строки
SHIFT+6 (^)	К первому непробельному символу в текущей строке

Клавиша	Перемещает курсор
SHIFT+4 (\$)	В конец текущей строки

Многие команды в vim могут начинаться с числа. Добавляя число в команду, можно указать, сколько раз она должна быть выполнена. Например, команда 5j переместит курсор на пять строк вниз.

Замечание

Почему для перемещения курсора были выбраны клавиши h, j, k и l? Потому что, когда был написан редактор vim, не все видеотерминалы имели кнопки со стрелками на клавиатуре.

Кроме того, в нормальном режиме редактор vim предоставляет доступ к специальной функции, называемой режимом командной строки. В режиме командной строки пользователю предоставляется 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 :. После этого курсор перемещается в строку сообщений, появляется символ двоеточия как приглашение к вводу команды.

Рассмотрим некоторые из команд командной строки:

- 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 имя файла, предназначенная для сохранения файла под другим именем;
- команда wq, обеспечивающая сохранение данных буфера в файле и завершение работы.

Редактирование текста

В процессе работы над текстом приходится вставлять и удалять данные в файл. В нормальном режиме предусмотрено несколько команд редактирования текста.

Команда	Описание
x	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
u	Отменить предыдущую команду редактирования
a	Добавить данные после текущей позиции курсора
A	Добавить данные в конец текущей строки
г символ	Заменить отдельный символ в текущей позиции курсора символом char

Команда	Описание
R текст	Перезаписывать данные в текущей позиции курсора, заменяя текстом <code>text</code> , пока не будет нажата клавиша ESC (режим замены)
o	Вставить пустую строку ниже текущей
O	Вставить пустую строку выше текущей

Эти команды также могут начинаться с числа. Например, команда `2x` удалит два символа, начиная с текущей позиции курсора, а команда `5dd` удалит пять строк, начиная со строки, в которой находится курсор.

Копирование и вставка

В таблице из подраздела 6.1 были показаны команды, которые позволяют удалять текст. Однако редактор `vim` не просто удаляет текст, в копирует его в буфер вставки (своего рода буфер обмена), откуда удаленный текст можно получить с помощью команды `p` (он будет вставлен правее позиции курсора) или `P` (он будет вставлен левее позиции курсора).

Например, можно удалить строку текста с помощью команды `dd`, затем переместить курсор в то место в файле, в котором она должна быть, после чего применить команду `p`. Аналогичную операцию можно проделать с помощью любой команды, которая удаляет текст.

Команда `y` выполняет копирование данных. С командой `y` можно использовать такой же второй символ, как и с командой `d` (например, вариант команды `yw` означает, что в память должно быть скопировано слово, а команда `y$` указывает, что в память должен быть скопирован текст до конца строки). После копирования текста в буфер его можно вставить в нужное место в файле с помощью команды `p`.

Сложность такого копирования заключается в том, что иногда трудно отследить что именно было скопировано в буфер. Однако в редакторе `vim` предусмотрено средство, которое упрощает применение операции копирования. Это, так называемый, визуальный режим, предусматривающий выделение текста подсветкой при перемещении курсора. Чтобы перейти в визуальный режим, переместите курсор в то место, где должно начаться копирование, и введите команду `v`. После этого перемещайте курсор, чтобы выделить текст, который должен быть скопирован в буфер. Когда вы закончите выделять текст, нажмите клавишу `u` для вызова команды копирования в память.

Поиск и замена

Чтобы ввести строку для поиска, используйте команду `/` (прямой слеш). Курсор перейдет в строку сообщений и редактор `vim` выведет косую черту. Наберите текст, который должен быть найден, и нажмите клавишу ENTER. После этого курсор переместится к следующему вхождению искомой строки. Поиск следующего вхождения той же строки можно повторить командой `n`.

Команда замены позволяет быстро заменять одно слово другим в тексте. Чтобы воспользоваться командой замены, необходимо перейти в режим командной строки. Команда замены имеет следующий формат:

```
1 | :s/old/new/
```

При выполнении этой команды редактор vim переходит к первому вхождению текста old и заменяет его текстом new.

Команда замены имеет несколько 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 выводит приглашения для подтверждения каждой замены.

Литература

1. Шоттс У. “Командная строка Linux.”, глава 12
2. Блум Р., Бреснахэн К. “Командная строка Linux и сценарии оболочки.”, глава 9, подраздел “Редактор vim”