

Текстовый редактор Vi

Главный текстовый редактор для Linux и UNIX - это программа, которая называется vi. Хотя для Linux доступно множество редакторов, от крошечного редактора nano до массивного редактора emacs, у редактора vi есть несколько преимуществ:

Редактор vi доступен во всех дистрибутивах Linux в мире. Это не относится к любому другому редактору.

Редактор vi может быть выполнен как в CLI, так и в GUI. Хотя графические редакторы, такие как gedit из среды рабочего стола Gnome или kedit из среды рабочего стола K, проще в использовании, они требуют графического интерфейса, который не всегда будет работать на серверах.

В то время как новые функции были добавлены в редактор vi, основные функции существуют уже несколько десятилетий. Это означает, что если бы кто-то изучил редактор vi в 1970-х годах, он мог бы без проблем использовать современную версию. В то время как это кажется тривиальным, это может показаться не таким уж тривиальным через двадцать лет.

Оригинальный редактор vi был написан Биллом Джой, соучредителем Sun Microsystems. Поскольку vi является частью Единой спецификации UNIX (SUS), требуется, чтобы соответствующие системы на основе UNIX имели ее. Поскольку база стандартов Linux (LSB) отражает требования SUS, системы Linux, соответствующие LSB, также должны включать редактор vi.

В действительности, большинство систем Linux не включают в себя оригинальный vi, а улучшенную версию, известную как vim. Этот факт может быть скрыт большинством дистрибутивов Linux. В некоторых файлах vi будет ссылка на vim, в то время как в других существует псевдоним, который будет выполнять vim при запуске команды vi:

which vi

```
[root@R-FW ~]# which vi
/bin/vi
```

По большей части vim работает так же, как vi, но имеет дополнительные функции.

Чтобы начать использовать vi, просто введите команду, а затем путь к файлу для редактирования или создания:

vi newfile

Перемещение в командном режиме

В vi используются три режима: **командный режим, режим вставки и режим ex**. Изначально программа запускается в **командном режиме**. Командный режим используется для ввода команд, например команд, используемых для перемещения по документу, манипулирования текстом и доступа к двум другим режимам. Чтобы в любой момент вернуться в командный режим, нажмите клавишу ESC.

После добавления текста в документ для выполнения таких действий, как перемещение курсора, сначала необходимо нажать клавишу ESC, чтобы вернуться в командный режим. Это кажется большой работой, но помните, что vi работает в терминальной среде, где мышь бесполезна.

Команды перемещения в vi имеют два аспекта: **движение и необязательный префикс номера**, который указывает, сколько раз повторить это движение. Общий формат выглядит следующим образом:

[count] motion

В следующей таблице приведены доступные клавиши перемещения:

<code>h</code>	Left one character
<code>j</code>	Down one line
<code>k</code>	Up one line
<code>l</code>	Right one character
<code>w</code>	One word forward
<code>b</code>	One word back
<code>^</code>	Beginning of line
<code>\$</code>	End of the line

Этим движениям может предшествовать число, указывающее, сколько раз выполнить движение. Например, **5h** переместит курсор на пять символов влево, **3w** переместит курсор на три слова вправо.

Чтобы переместить курсор к определенному номеру строки, введите этот номер строки, за которым следует символ **G**. Например, чтобы добраться до пятой строки типа файла **5G**. **1G** или **gg** можно использовать для перехода к первой строке файла, в то время как одинокая буква **G** приведет вас к последней строке. Чтобы узнать, на какой строке находится курсор в данный момент, используйте **CTRL-G**.

Стандартное соглашение для редактирования контента с помощью текстовых процессоров заключается в использовании копирования, вырезания и вставки. Программа **vi** не имеет ни одного из них, вместо этого **vi** использует следующие три команды:

<code>cut</code>	<code>d</code>	<code>delete</code>
<code>copy</code>	<code>y</code>	<code>yank</code>
<code>paste</code>	<code>P p</code>	<code>put</code>

Движения, извлеченные из предыдущей страницы, используются для указания того, где должно происходить действие, всегда начиная с текущего местоположения курсора. Любой из следующих общих форматов для команд действия приемлем:

action [count] motion
[count] action motion

Удаление

Delete удаляет указанный текст со страницы и сохраняет его в буфере, который является эквивалентом «буфера обмена», используемого в Windows или Mac OSX. В следующей таблице приведены некоторые распространенные примеры использования:

Action	Result
<code>dd</code>	Delete current line
<code>3dd</code>	Delete the next three lines
<code>dw</code>	Delete the current word
<code>d3w</code>	Delete the next three words
<code>d4h</code>	Delete four characters to the left

Изменить

Change очень похоже на удаление, текст удаляется и сохраняется в буфере, однако программа переключается в режим вставки для немедленного изменения текста. В следующей таблице приведены некоторые распространенные примеры использования:

Action	Result
cc	Change current line
cw	Change current word
c3w	Change the next three words
c5h	Change five characters to the left

Yank

Yank помещает содержимое в буфер, не удаляя его. В следующей таблице приведены некоторые распространенные примеры использования:

Action	Result
yy	Yank current line
3yy	Yank the next three lines
yw	Yank the current word
y\$	Yank to the end of the line

Put

Put помещает текст, сохраненный в буфере, до или после позиции курсора. Обратите внимание, что это единственные две опции, **put** не использует движения, как предыдущие команды действий.

Action	Result
p	Put (paste) after cursor
P	Put before cursor

Поиск в vi

Другая стандартная функция, которую предлагают текстовые процессоры - это **find**. Программа vi использует поиск. Поиск более мощный, чем поиск, потому что он поддерживает как текстовые шаблоны, так и регулярные выражения.

Для поиска вперед от текущей позиции курсора используйте /, чтобы начать поиск, введите поисковый термин, а затем нажмите клавишу Enter, чтобы начать поиск. Курсор переместится к первому найденному совпадению.

Чтобы перейти к следующему совпадению, используя тот же шаблон, нажмите клавишу n. Чтобы вернуться к предыдущему совпадению, нажмите клавишу N. Если достигнут конец или начало документа, он автоматически переместится на другую сторону документа.

Чтобы начать поиск назад с позиции курсора, начните с ввода ?, Затем введите образец для поиска совпадений и нажмите клавишу Enter.

Режим вставки

Режим вставки используется для добавления текста в документ. Есть несколько способов войти в режим вставки из командного режима, каждый из которых отличается тем, где начнется вставка текста. Следующая таблица охватывает наиболее распространенные:

Input	Purpose
<code>a</code>	Enter insert mode right after the cursor
<code>A</code>	Enter insert mode at the end of the line
<code>i</code>	Enter insert mode right before the cursor
<code>I</code>	Enter insert mode at the beginning of the line
<code>o</code>	Enter insert mode on a blank line after the cursor
<code>O</code>	Enter insert mode on a blank line before the cursor

Ex Mode

Первоначально редактор `vi` назывался **ex editor**. Название **vi** было аббревиатурой визуальной команды в бывшем редакторе, которая переключала редактор в «визуальный» режим.

В исходном обычном режиме бывший редактор позволял пользователям видеть и изменять только одну строку за раз. В визуальном режиме пользователи могли видеть столько документов, сколько поместится на экране. Поскольку большинство пользователей предпочитали визуальный режим, а не режим редактирования строки, файл программы `ex` был связан с файлом `vi`, так что пользователи могли запускать `ex` непосредственно в режиме визуализации при запуске ссылки `vi`.

В конце концов, фактический файл программы был переименован в `vi`, а бывший редактор стал ссылкой, указывающей на редактор `vi`.

Когда используется `ex`-режим редактора `vi`, можно просматривать или изменять настройки, а также выполнять команды, связанные с файлами, такие как открытие, сохранение или отмена изменений в файле. Чтобы перейти в режим `ex`, введите символ “:” в командном режиме. В следующей таблице перечислены некоторые общие действия, выполняемые в режиме `ex`:

Input	Purpose
<code>:w</code>	Write the current file to the filesystem
<code>:w filename</code>	Save a copy of the current file as filename
<code>:w!</code>	Force writing to the current file
<code>:1</code>	Go to line number 1 or whatever number is given
<code>:e filename</code>	Open filename
<code>:q</code>	Quit if no changes made to file
<code>:q!</code>	Quit without saving changes to file

Быстрый анализ таблицы выше показывает, если восклицательный знак! добавляется в команду, затем пытается форсировать операцию. Например, представьте, что вы вносите изменения в файл в редакторе `vi`, а затем пытаетесь выйти с помощью: `q`, только чтобы обнаружить, что команда “quit” завершается неудачно. Редактор `vi` не хочет выходить без сохранения изменений, внесенных в файл, но вы можете заставить его выйти с помощью команды `ex: q!`.

Редактор **vi** может сохранять изменения в файле только для чтения. Команда: **w!** попытается записать в файл, даже если он доступен только для чтения, при попытке изменить разрешения для файла, выполнить запись в файл и затем изменить разрешения обратно на те, которые были изначально.

Это означает, что пользователь **root** может вносить изменения практически в любой файл в редакторе **vi** независимо от прав доступа к файлу. Тем не менее, обычные пользователи смогут принудительно записывать только те файлы, которые им принадлежат. Использование **vi** не меняет того факта, что обычные пользователи не могут изменять права доступа к файлу, который им не принадлежит.

Хотя **ex**-режим предлагает несколько способов сохранить и выйти, также есть **ZZ**, который доступен в командном режиме; это эквивалент: **wq**. Есть много других перекрывающихся функций между режимом **ex** и командным режимом. Например, **ex mode** можно использовать для перехода к любой строке в документе, набрав: с последующим номером строки, а **G** можно использовать в командном режиме, как было показано ранее.