

Редактор VI

<https://losst.ru/kak-polzovatsya-tekstovym-redaktorom-vim>

Как пользоваться текстовым редактором vim

Опытные пользователи Linux часто используют терминал, потому что так можно намного быстрее выполнить необходимые действия. Во время настройки системы нам довольно часто приходится редактировать различные файлы. Это могут быть настройки программ, какие-нибудь данные или обычные текстовые файлы.

В операционной системе Linux есть несколько текстовых редакторов, которые работают в терминале. Чаще всего новички используют редактор nano, но если вы заметили на нашем сайте во всех статьях используется текстовый редактор vi. Nano неудобный, и недостаточно функционален. Я сознательно не пишу в своих статьях о nano. Есть намного лучший текстовый редактор, это редактор vi. Здесь поддерживается быстрое перемещение по тексту, удобное редактирование, команды для изменения настроек работы, выполнение команд терминала из редактора, а также плагины для расширения функциональности. Но он немного сложный для новичков и очень непривычный.

В этой статье мы рассмотрим как пользоваться vim, рассмотрим основы работы с этим редактором, а также его основные команды.

Минимальные основы

На данный момент существует две версии редактора - vi и vim. Vim расшифровывается как Vi Improved, улучшенный vi. Это новая версия, которая принесла очень много улучшений. В большинстве современных дистрибутивов используется именно она. Поэтому если я буду писать vi, это значит, что я предполагаю использование vim.

Текстовый редактор Vim может работать в двух режимах. Это и есть его главная особенность. Первый режим, который используется по умолчанию при открытии редактора - это командный. В этом режиме вы можете вводить команды vi, а также использовать символьные клавиши для управления редактором. Второй режим - обычное редактирование текста, он работает так же как и редактирование текста в nano. Для переключения в командный режим используется клавиша Esc. Для переключения в режим редактирования - клавиша i. Если вас интересует только как в редакторе vi сохранить и выйти, листайте вниз, но если вы хотите узнать как пользоваться текстовым редактором vim, эта статья для вас.

Перед тем как идти дальше я бы посоветовал вам пройти курс обучения встроенный в редактор. Выполнение всех обучающих заданий займет 25-30 минут. Но после того как вы освоите все что там написано, эта статья поможет вам закрепить материал. Дело в том, что команд и сочетаний клавиш у vim очень много и запомнить их все без практики невозможно. Для запуска обучения наберите:

```
$ vimtutor
```

Но делать это сейчас необязательно, в этой статье есть вся необходимая базовая информация и после ее прочтения вы уже сможете уверенно пользоваться vim, а обучение пройти чуть позже.

Как использовать редактор Vim

Начнем мы, как обычно с запуска программы, а также опций, которые ей можно передать. Синтаксис Vim очень прост:

```
$ vim опции имя_файла
```

Или:

```
$ vi опции имя_файла
```

Простой запуск vim без указания имени файла приведет к созданию пустого файла. А теперь давайте рассмотрим основные опции запуска:

- +номер - переместить курсор к указной строке после запуска.
- +/шаблон - выполнить поиск по шаблону и переместить курсор к первому вхождению
- "+"команда" - выполнить команду после запуска программы
- b - двоичный режим, для редактирования исполняемых файлов.
- d - режим поиска различий в файлах, нужно указать несколько файлов для открытия.
- g - графический режим.
- n - не использовать автосохранение для восстановления файла при сбое.
- R - режим только для чтения.
- w - сохранить все действия в файл.
- x - шифровать файл при записи.
- C - режим совместимости с Vi.

Круто, правда? Но это только начало. Опции ничего по сравнению с командами редактора.

Командный режим Vim

Чтобы перейти в режим редактирования нужно нажать A

Чтобы выйти из режима редактирования нужно нажать Esc

Чтобы перейти в режим командной строки нужно нажать Shift-:

В командном режиме вы можете перемещаться по редактируемому тексту и выполнять действия над ним с помощью буквенных клавиш. Именно этот режим открывается по умолчанию при старте редактора. Здесь вы будете использовать краткие команды, перед которыми может устанавливаться номер, чтобы повторить команду несколько раз. Для начинающих может быть поначалу очень запутанно то, что в командном режиме символы интерпретируются как команды.

Для перемещения используются такие команды:

- h - на один символ влево;
- l - на один символ вправо;
- j - на одну строку вниз;

- k - на одну строку вверх;
- w - на слово вправо;
- b - на слово влево;
- H - перейти в низ экрана;
- G - перейти в конец файла;

Можете запустить редактор и поэкспериментировать, чтобы было легче понять как это работает. Если перед тем как нажать кнопку буквы нажать цифру, то эта команда будет повторена несколько раз. Например, 3j переведет курсор на три строки вверх.

Для переключения в режим редактирования используются такие команды:

- i - вставить текст с позиции курсора, символ под курсором будет заменен;
- I - вставить текст в начало строки;
- a - добавить текст начиная от позиции курсора;
- o - вставить новую строку после этой и начать редактирование;
- O - вставить новую строку перед этой и начать редактирование;
- r - заменить текущий символ;
- R - заменить несколько символов.

К этим командам тоже применимы символы повторения. Поэкспериментируйте, можно получить интересный и не совсем ожидаемый результат.

Более сложны команды редактирования текста. Вы можете править текст не только в обычном режиме, но и в командном с помощью команд. Для этого применяются такие команды:

- d - удалить символ;
- dd - удалить всю строку;
- D - удалить символы начиная от курсора и до конца строки;
- y - копировать символ;
- yy или Y - скопировать всю строку;
- v - выделить текст;

Эти команды редактора vim работают немного по-другому после нажатия одной из них ничего не произойдет. Мы еще можем задать количество символов, к которым будет применена команда и направление, с помощью кнопок перемещения курсора. Например, чтобы удалить два символа справа от курсора нажмите d3l, а чтобы удалить три строки вниз - d3j. Команды yy, dd, Y - не что иное, как сокращения.

Кроме этих команд, есть еще несколько полезных, которые мы не можем не рассмотреть:

- p - вставить после позиции курсора;
- P - вставить перед позицией курсора;
- u - отменить последнее действие;
- . - повторить еще раз последнее действие;
- U - отменить последнее действие в текущей строке;
- /шаблон - искать вхождение;

%s/шаблон/заменить - заменить первое слово на второе;

n - продолжить поиск вперед;

N - продолжить поиск назад;

С основными командами разобрались. Но у нас есть еще командная строка Vim, которая сама по себе тоже представляет огромный интерес.

Командная строка Vim

Командная строка Vim запускается в командном режиме нажатием двоеточия - ":". Здесь доступны команды для сохранения файла и выхода из редактора, настройки внешнего вида и взаимодействия с внешней оболочкой. Рассмотрим наиболее часто используемые команды редактора vim:

:w - сохранить файл;

:q - закрыть редактор;

:q! - закрыть редактор без сохранения;

:e файл - прочитать содержимое файла в позицию курсора;

:r файл - вставить в содержимое файла в следующую строку;

:r! - выполнить команду оболочки и вставить ответ в редактор;

:set переменная=значение - установить значение переменной, например, tabstop=4, или set number, с помощью этой команды можно управлять многими аспектами работы vim.

:buffers - посмотреть открытые файлы.

Со всеми основами разобрались, и вы теперь использование vim не будет казаться вам таким сложным. Но это еще далеко не все, этот мощный редактор может еще очень многое. Дальше мы рассмотрим несколько примеров использования vim, чтобы вам было легче справиться с новой программой.

Редактирование файла в Vim

Несмотря на то, что из всего вышесказанного можно понять как это делается рассмотрим еще раз. Чтобы открыть файл выполните:

\$ vim имя_файла

Затем, если вы не хотите пока использовать возможности командного режима просто нажмите i, чтобы перейти в режим редактирования. Здесь вы можете редактировать файл так же, как и в nano. После того как завершите нажмите Esc, чтобы перейти в командный режим и наберите :wq. Записать и выйти. Все, готово.

Поиск и замена в Vim

Довольно часто нам нужно найти определенную последовательность в тексте. Текстовый редактор Vim умеет это делать.

Во-первых, если нужно найти символ в строке, нажмите f и наберите нужный символ, курсор будет перемещен к его позиции.

Для поиска по всему файлу используйте команду /. После нее нужно ввести слово, которое нужно найти. Для поиска следующего вхождения используйте n, для предыдущего - N.

Для замены будет использоваться немного другая конструкция:
:%s/искать/заменить/g

Двоеточие запускает командную оболочку с командой s для замены. Символ % означает что обрабатывать нужно весь файл, а g значит, что нужно обработать все найденные строки, а не только первую. Чтобы программа спрашивала перед каждой заменой можно добавить в конец строки опцию. Одновременное редактирование нескольких файлов

Чтобы открыть несколько файлов, просто передайте их в параметры при запуске программы:
\$ vim файл1 файл2 файл3

Редактор vim linux откроет первый файл, для переключения ко второму используйте команду :n, чтобы вернуться назад :N.

С помощью команды :buffers вы можете посмотреть все открытые файлы, а командой :buffer 3 переключится на третий файл.
Буфер обмена Vim

Текстовый редактор Vim имеет свой буфер обмена. Например, вам нужно скопировать в четыре строки и вставить их в другое место программы, для этого выполните такую последовательность действий:

- Нажмите Esc, чтобы перейти в командный режим;
- Наберите 4yy чтобы скопировать четыре строки;
- Переместите курсор в место где нужно вставить эти строки;
- Нажмите p для вставки.

Также можно использовать выделение vim, чтобы скопировать строки. Выделите текст с помощью v, а затем нажмите у, чтобы скопировать.
Кириллица в Vim

Кириллица в Vim работает превосходно. Но есть одно но, когда включена кириллица в системе, все команды vim не работают, им и не нужно работать, они же не приспособлены для кириллицы.

Но переключать каждый раз раскладку, когда работаете в командном режиме тоже не очень удобно, поэтому открываем файл ~/.vimrc и добавляем туда такие строки:
set keymap=russian-jcukewin
set iminsert=0
set imsearch=0

Теперь раскладка клавиатуры в командном режиме переключается по Ctrl+^ и все команды работают.