

# Знакомство с VIM

Большая часть нашего взаимодействия с Linux — это работа с текстовыми файлами. Настройка программы — это зачастую правка текстового файла.

Вся работа системного администратора, devops-инженера связана с работой с файлами.

Почему именно Vim?

1. Он не требует графического интерфейса, на серверах его обычно нет. По этой же причине, кстати, он требует гораздо меньше системных ресурсов, памяти и процессора, чем другие текстовые редакторы.
2. Он удобен и не требует для работы мышь или трекпад (если вам придётся работать в серверной, подключившись к серверу напрямую, мыши может не оказаться).
3. Он предоставляет много крутых возможностей: нумерация строк, массовая правка текста, подсветка синтаксиса. Различные дополнения позволяют его использовать и как среду разработки тоже.

## Способы просмотра и редактирования файлов

Способ 1.

```
cat myfirstfile
```

Эта утилита выводит на консоль сразу всё содержимое файла. Это удобно, когда файл небольшой и вы, например, хотите скопировать что-то и выполнить в командной строке.

Если файл большой, лучше воспользоваться другим способом.

Способ 2.

```
less myfirstfile
```

Эта программа позволяет искать по файлу, пролистывать его целыми страницами. Это удобно, например, если у нас есть файл с разнообразными событиями. Если этой программы нет, её нужно установить.

Для того чтобы воспользоваться поиском по файлу, просто нажмите слеш (/).

Для выхода — букву q на клавиатуре.

Способ 3.

```
vim -r myfirstfile
```

Это самый простой способ использования редактора Vim. Ключ -г запрещает редактирование файла. Это удобно, если вы работаете с файлом, изменения в котором могут быть опасны.

Как выйти из редактора

Для выхода из редактора нам нужно набрать :q! и нажать Enter.

Это способ для выхода без сохранения файла, но так как мы всё равно не смогли бы файл отредактировать, сохранять нам ничего и не требуется.

## Редактирование файлов при помощи Vim

Vim, скорее всего, не похож на те текстовые редакторы, с которыми вы сталкивались.

У него есть несколько режимов работы.

- Первый и основной – командный режим. Это режим, в который вы попадаете сразу после запуска редактора. В нём мы не можем писать текст, но можем перемещаться по тексту и редактировать его.  
Чтобы перейти из любого другого режима в командный режим, нужно нажать клавишу Esc.
- Режим командной строки. Этот режим вызывается нажатием : из командного режима. За двоеточием следует какая-нибудь команда (например, поиска или замены), которая после ввода выполняется нажатием ↵ Enter. После выполнения команды редактор возвращается в командный режим.
- Режим ввода. Здесь происходят знакомые вещи, мы пишем текст.  
Для перехода в режим ввода используются клавиши i (для начала вставки текста на месте курсора) или a (для начала вставки текста после курсора). В этом режиме нажатые вами клавиши не воспринимаются как команды, а воспринимаются как вводимый текст.
- Визуальный режим. В этом режиме мы сможем выделить текст и применить к нему какую-нибудь команду, например замены. Редактировать в этом режиме проще, чем в командном, потому что сразу видно, к чему вы применяете команду.

Мы уже умеем открывать файл при помощи Vim и выходить из него без сохранения.

## Создание файла при помощи Vim

Зайдём в нашу домашнюю директорию и откроем Vim.

Мы оказались в командном режиме. Чтобы начать ввод текста, нажмём i или I (чтобы начать вставку текста с начала строки). Традиционно первая программа, которую пишут все программисты, выводит строки “Hello World!”. Не будем отступать от традиции, наберём “Hello World!” и нажмём Esc, чтобы выйти в командный режим. Теперь файл нужно сохранить, для этого перейдём в режим командной строки : и наберём write hello\_world

Как мы помним, после нажатия Enter мы снова попали в командный режим. Чтобы закрыть редактор, снова войдём в режим командной строки : и наберём q Enter.

Редактор закрылся, мы видим, что создан файл hello\_world.

```
ls
```

Для того чтобы текстовый файл открывался без проблем, он не должен заканчиваться на .txt

Поэтому с одного взгляда не всегда понятно, что за файл перед вами. Для того чтобы это выяснить, можно использовать команду *file*.

```
file hello_world
```

Это ASCII text. Его безопасно открыть утилитой cat.

Если вы увидите что-то вроде такого:

```
file /bin/less
/bin/less: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV),
dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux
2.6.32, BuildID[sha1]=76f473ce4a656996c6b4f665320df70da4acbf45, stripped
```

Это бинарный файл, в нём нет текста, который может прочитать человек, а данные специально организованы так, чтобы их мог выполнить компьютер. Открывать такой файл cat'ом не опасно, но и не очень полезно.

Вернёмся в Vim.

Мы будем пользоваться заранее подготовленным файлом с текстом (ссылка в описании к видео).

Откройте браузер внутри вашей виртуальной машины, перейдите по этой ссылке и загрузите его. Он вам, скорее всего, предложит сохранить файл в папку Downloads. Она находится в вашей домашней директории, вы можете перейти в неё из консоли командой cd, например вот так:

```
cd Downloads
```

Чтобы открыть при помощи Vim файл, можно сразу указать его при запуске.

Проверим, что мой файл хороший:

```
file algebraist
```

Написано, что это текст.

```
vim algebraist
```

Теперь у нас есть некоторое количество текста, на котором можно вам показать функции Vim.

## Функции Vim

Для перемещения по файлу в командном режиме можно пользоваться стрелками. Но это не очень удобно, ведь для этого нужно переносить руку вправо. Поэтому для перемещения можно пользоваться клавишами:

h — влево.

l — вправо.

j — вниз.

k — вверх.

Клавишей w можно перемещаться в начало другого слова.

Если нужно добавить текст перед первым предложением или начать писать с конца последнего предложения в строке, то клавишами I и A можно переместиться в начало или в конец строки соответственно. Вы попадёте в режим редактирования, и вы можете начать писать текст.

Если вы не хотите начинать писать текст, а, например, планируете выделить, то для перехода в начало либо в конец строки, оставаясь в командном режиме, используйте Shift + ^ и Shift + \$ соответственно.

Предположим, мы хотим переместиться не на следующую строчку, а через одну. 2j перенесёт нас на две строки вниз. Может показаться, что мы перепрыгнули больше, но строкой мы считаем всё до переноса строки (то есть пока не нажали Enter).

Включить нумерацию строк. Перейдём в режим командной строки, нажав :, затем наберём set number и нажмём Enter. Чтобы это отключить, наберём set nonumber.

Если мы хотим перемещаться по словам не подряд, а через два или через три, с использованием 2w мы можем прыгать через слово. Или через два с 3w. Это же верно и для символов — 2l будет перемещать нас через символ.

Для клавиш Page Up (вверх) и Page Down (вниз) тоже есть замена:

gg перемещает нас в начало документа. G — в конец (очень полезно, если у вас Mac и клавиш Page Up / Page Down нет). Но если набрать номер строки и G — мы переместимся на строку с этим номером.

## Визуальный режим

Здесь есть:

v — для посимвольного выделения.

V — для построчного выделения.

А при помощи Ctrl + V мы можем выделить блок текста. Для выхода из этого режима также используется клавиша Esc.

Скопировать пятую строчку и вставить её два раза. Команда будет выглядеть так: 5G — чтобы перейти на пятую строку.

V — для того чтобы выделить строку.  
y — чтобы скопировать.  
2p — чтобы вставить.

Чтобы удалить текущую строку, используется команда dd, а чтобы удалить две строки, d2d.

Кстати, у нас есть возможность отменить действие, это делается командой u.

Найти слово moon в документе. Для этого наберём /moon и нажмём Enter.  
Далее клавишей n или N мы можем перемещаться между найденными словами.

Заменить слово moon во всём тексте на Moon.

Способ 1. Для того чтобы заменить символ, на котором стоит курсор, на другой, используется команда r и сразу после неё символ, на который мы хотим заменить.  
Так что, в принципе, можно найти слово /moon, затем нажать rM esc и после этого n, чтобы перейти к следующему слову.

Способ 2. Для этого переходим в режим командной строки.  
Далее значок процента (%), для того чтобы наши изменения применились ко всему файлу. Та же команда без значка процента заменит слово только в текущей строке, на которой установлен курсор.

`:%s/moon/Moon/g` и нажимаем Enter.

## Добавить Vim в консоль

Это нам потребуется в домашней работе и в работе в дальнейшем.

У sudo есть конфигурация. Но, как и в случае с файлами /etc/passwd, мы не редактируем её напрямую, а используем специальную утилиту, которая не позволит нам создать плохую конфигурацию или где-нибудь опечататься. Команда для редактирования sudo называется visudo.

К сожалению, если вы наберёте visudo, она откроет нам не Vim (несмотря на название!). Давайте это исправим. Наберите в консоли:

```
sudo update-alternatives -config editor
```

Затем нажмите 3 и нажмите Enter.