

## Основные команды текстового редактора VI / VIM

Текстовый редактор **vim**, созданный на основе более старого **vi**. Один из мощнейших текстовых редакторов с полной свободой настройки и автоматизации, возможными благодаря расширениям и надстройкам. По умолчанию входит в состав любого дистрибутива Linux.

Открываем файл с помощью **vi / vim**:

```
vim test.txt
```

```
vi test.txt
```

### Основные режимы работы.

«**Обычный режим**» — перемещение по файлу, стирание текста и другие редактирующие функции. Это - основной режим, только из него можно сразу перейти в другие режимы. Для возврата в основной режим из любого другого режима:

<ESC>, иногда 2 раза;

<Ctrl-[>

«**Режим ввода**» - ввод текста. Как только завершается ввод текста, принято сразу возвращаться в обычный режим. Заметьте, что стирание и ввод текста происходит в двух разных режимах.

Переход в него из обычного режима:

```
i
```

```
<Insert>
```

«**Командный режим**» - Команды (операции с файлом, поиск и замена, настройка редактора...).

Переход в него из обычного режима:

```
:
```

«**Режим поиска**» - ввод поискового запроса. Переход в него из обычного режима

/, поиск от курсора до конца документа

?, поиск от курсора до начала документа

«**Визуальный режим**» - режим выделения текста:

"v" и влево или вправо стрелками;

Shift+v вся строка целиком;

Ctrl+v прямоугольник, часть текста.

### Перемещение по файлу

После загрузки Vim, на экране вы увидите часть загруженного вами текстового файла. Загрузившись, Vim находится в «командном режиме» - один из основных режимов. Это значит, что если вы нажмете клавишу <L> (строчная L), вместо появления «l» на месте курсора вы увидите, что курсор сдвинулся на один символ вправо. В командном режиме знаки, набираемые на клавиатуре, используются как команды для **Vim**, а не как помещаемые в текст символы. Команды перемещения - один из наиболее важных типов команд. Вот некоторые из них:

Для перемещения курсора нажмите клавиши h,j,k,l так, как показано ниже:

```
^
```

```
k      Советы: Клавиша h находится слева и перемещает влево.
```

```
< h    l >      Клавиша l находится справа и перемещает вправо.
```

```
j      Клавиша j похожа на стрелку "вниз".
```

```
v
```

<Ctrl-f> - на страницу (экран) вниз;

<Ctrl-b> - на страницу (экран) вверх;

<Ctrl-d> - на пол страницы (экрана) вниз;

<Ctrl-u> - на пол страницы (экрана) вверх;

<Ctrl-y> - на строку вниз, без изменения положения курсора;

<Ctrl-e> - на строку верх, без изменения положения курсора;  
0 («ноль») - в начало текущей строки;  
^ - в начало текущей строки (к первому непробельному символу);  
\$ - в конец текущей строки;  
w - на слово вправо;  
b - на слово влево;  
W - до пробела вправо;  
B - до пробела влево;  
} - абзац вниз;  
{ - абзац вверх;  
gg - перейти в начало файла;  
G - перейти в конец файла;  
<number>G - перейти на конкретную строку <number>;  
/<text><CR> - перейти к <text>;  
?<text><CR> - то же самое, но искать назад;  
n - повторить поиск;  
N - повторить поиск назад;  
[[ - в начало функции;  
" - к месту выполнения команды [[

### **Ввод текста**

Следующие команды переводят редактор в режим ввода:

i — перейти в режим ввода с текущей позиции

a — перейти в режим ввода после курсора

I — переместиться в начало строки и перейти в режим ввода

A — переместиться в конец строки и перейти в режим ввода

o — перейти в режим ввода с новой строки под курсором

O — перейти в режим ввода с новой строки над курсором

s - заменяет указанное количество символов (удаляет указанное число символов и переходит в режим ввода). В отличии от команды s которая может удалить кусок текста размером не меньше слова (sw), командой s можно удалить любое число символов. Например 4s удалит четыре символа начиная с того который находится под курсором. Эта команда применяется для замены одного или нескольких символов другими символами.

S - удаляет всю текущую строку и переходит в режим ввода. Число перед командой показывает сколько нужно удалить строк начиная с текущей. Например 4S удалит четыре строки включая текущую.

R — перейти в режим ввода с заменой текста (аналог insert). Символы под курсором заменяются на вводимые. Команда применяется когда неизвестно сколько придётся изменить символов на другие (иначе можно было бы использовать команду s с указанием числа заменяемых символов, например, 7s). При удалении вводимых символов возвращаются те которые были до ввода команды. Такой режим сохраняется до конца строки. При вводе новой строки (по нажатию Enter), происходит не переход на другую строку с тем же режимом замены текста, а создание новой строки.

r — заменить один символ. Заменяет символ находящийся под курсором на символ который следует за командой. При этом не происходит выхода из командного режима (не надо нажимать ESC после изменения текста). Например, команда ru - символ под курсором меняется на "y". Числовой показатель указывает сколько символов необходимо заменить на данный. Например, 3ru вставляет три символа "y".

## Удаление и вставка

Ниже перечислены основные команды удаления и вставки текста:

x — удалить символ под курсором (<число>x удаляет указанное число символов начиная с того который находится под курсором);

X — удалить символ влево (удалить символ перед курсором);

d — используется совместно с командами перемещения. Удаляет символы с текущего положения курсора до положения после ввода команды перемещения. Пример:

dw - удаляет символы с текущего до конца слова, включая пробел после слова, чтобы удалить символы до конца слова, но оставить пробел используется команда de,

dE - удаляет символы с текущего до конца слова, включая символы пунктуации, но оставляет пробел после слова,

diw - удаляет слово под курсором,

dd — удалить текущую строку (вырезать);

d<число>d или <число>dd — стирание числа строк начиная с текущей;

db - удаляет символы с текущего до начала слова (удаление в обратном направлении);

d0 - удаление символов с начала строки до текущего положения курсора;

d\$ или D - удаление символов с текущего положения курсора до конца строки;

c — команда аналогичная d, но после удаление переходит в режим ввода;

cc - команда удаляет текущую строку и переходит в режим ввода;

C - удаляет текст с текущего положения курсора до конца строки, аналогична команде c\$ (где \$ - символ конца строки);

yy (также Y) — копирование текущей строки в буфер;

y<число>y — копирование числа строк начиная с текущей в буфер;

p — вставка содержимого буфера под курсором. Поскольку в vim девять ячеек буфера удаления. Можно вставить не только последнее удаление, но и удаления сделанные ранее. Например "4p" вставит под курсор содержимое четвертого удаления начиная с последнего. Также чтобы поменять местами два символа можно использовать комбинацию команд "удалить" -x (удаление в буфер) и "вставить" - p (вставить из буфера). Таким образом, поставив курсор на первую букву из двух которые необходимо поменять местами и нажав комбинацию клавиш xp мы совершим необходимые действия;

P — вставка содержимого буфера перед курсором;

J — слияние текущей строки со следующей. Числовой аргумент перед командой показывает сколько следующих линий необходимо объединить с текущей. Например 2J объединить две следующие строки с текущей, на которой расположен курсор.

## Отмена изменений

u — отмена последней команды;

U — отмена всех последних изменений в строке, если строка удалена, то применить эту команду к данной строке будет невозможно.

## Поиск

Перейти на строку:

/фраза - поиск "фразы" во всем документе.

n - следующее найденное (вниз) N - предыдущее (вверх).

## Выход

Есть еще пара команд, которые вам необходимо знать:

:q! - выйти без сохранения;

:wq - записать файл и выйти;

ZZ - записать файл и выйти (Если файл не изменяли, то записываться он не будет)

В этой статье описаны основные возможности, необходимые для работы. Практически во все дистрибутивы Linux входит учебник по работе с редактором, - просто введите в командной строке команду *vimtutor*.

### **Справка**

Для вызова справки (помощи) о редакторе введите в терминале команду:

```
man vim
```