

Использование текстового редактора vim

Vim - очень мощный редактор с огромным набором функций. Ниже указаны основные команды, которых достаточно для того, чтобы начать работу с редактором.

Два режима

Есть два режима работы редактора: режим редактирования и командный режим. Переход в режим редактирования осуществляется различными командами. Выход из режима редактирования - кнопка **<Esc>**

Команды перемещения курсора

Команды перемещения работают в командном режиме. Нажмите **<Esc>**, чтобы убедиться что находитесь в нём.

Курсор можно перемещать стрелками на клавиатуре или, что гораздо быстрее, клавишами **h, j, k, l**:

h - влево
j - вниз
k - вверх
l — вправо

Открытие/создание файлов:

:new - создание нового документа в новом окне
:e <filename> - открытие файла для редактирования
:split - разделить на два экрана
:vsplit - разделить, если указан файл то открыть его (по вертикали)

Другие команды перемещения:

|, 0, home — перейти в начало строки
\$/, end — перейти в конец строки
m — перейти на половину ширины экрана
H — перейти на первую строку экрана
M — перейти на среднюю строку экрана
L — перейти на последнюю строку экрана
w — перейти на слово вперед
b — перейти на слово назад
gg - перейти в начало документа
G - перейти в конец документа
<номер> G - перейти на строку с номером **<номер>**
Можно использовать связку **<номер><команда перемещения>**
3w - перейти к началу четвёртого слова, считая с текущей позиции
4e - перейти к концу четвёртого слова, считая с текущей позиции
(— перейти на предложение назад (до точки)
) — перейти на предложение вперед (до точки)
{ — перейти на абзац назад (до пустой строки)
} — перейти на абзац вперед (до пустой строки)
:ju — список переходов.

Команды удаления

x - удалить букву под курсором
<число> x - удалить **<число>** букв, начиная с той что под курсором
dd - удалить строку
число dd или **d число d** - удалить **<число>** строк, начиная с той где находится курсор
Можно комбинировать команды удаления и перемещения:
dw - удалить 3 слова начиная с текущей позиции
d5w - удалить 5 слов
d\$ - удалить строку с текущего положения курсора до конца строки
d0 - удалить строку с текущего положения курсора до начала строки

Сохранение изменений

Для того чтобы сохранить отредактированный файл нужно:
- перейти в командный режим, нажав **<Esc>**
- ввести команду **:write <имя_файла> <ENTER>**,
- или её сокращённый вариант **:w <имя_файла> <ENTER>**
Если **имя_файла** отсутствует, то сохраняется текущий файл
Чтобы выйти из редактора введите **:quit <ENTER>** или просто **:q <ENTER>**
Выйти не сохранив изменения - **:q! <ENTER>**
:wq - сохранить изменения в текущем файле и выйти

Вставка текста

Для перехода в режим вставки нажмите **<i>**.
A - переход к концу строки и начало режима вставки
a - переход к следующему символу и начало режима вставки
o - начать вставку под текущей строкой
O - начать вставку над текущей строкой
<Esc> - переход к командному режиму

Отмена последних изменений

u - отмена последнего изменения
U - отмена изменений во всей строке

<CTRL-R> - отмена отмены

Копирование и вставка

yy - скопировать всю строку в буфер обмена

p - вставить содержимое буфера обмена после курсора

y5u или **5yy** - скопировать 5 строк

yw - скопировать слово

В буфер обмена попадают и только что удалённый текст. Например если удалить строку **dd**, то сразу же после этого её можно вставить с помощью **p**

Замена

r - заменить символ под курсором

R - заменять символы, начиная с текущего положения

c <число> <команда перемещения> - ещё одна команда замены.

cw - поменять слово.

c\$ - поменять строку, начиная с положения курсора и до конца.

cc - поменять всю строку.

Поиск

/ - поиск в прямом направлении

? - поиск в обратном направлении

n - перейти к следующему найденному элементу

p - перейти к предыдущему найденному элементу

Поиск соответствующей скобки [,), }

Наведите курсор в командном режиме на открывающуюся/закрывающуюся скобку и нажмите %. Курсор перейдёт к соответствующей закрывающейся/открывающейся скобке, если таковая есть

Определение положения в файле

Чтобы определить текущее положение нажмите <CTRL-G> в командном режиме. В нижней части терминала появится строка, содержащая информацию о количестве строк, текущем положении, имя файла.

Автоматическая замена

:s/old/new/g - заменить все сочетания 'old' в текущей строке на 'new'

:10,20s/old/new/g - тоже самое, но для строк с 10й по 20ю

Вызов справки

:help <команда> <ENTER> - вызов справки по команде редактора.

Выполнение shell-команд

!<команда>

Например:

!:ls - исполнить команду **ls** в текущей директории

Чтение файла

:r <filename> - вставить содержимое filename после курсора

Команда set

Некоторые настройки редактора можно менять не выходя из самого редактора с помощью команды **set**.

Например:

:set ic - ignore case - игнорировать регистр букв (например при поиске)

:set hls - highlighting search - подсвечивать найденное при поиске совпадение

:set nohls - не подсвечивать найденное при поиске совпадение

:set wrap - перенос слов

:set tw=80 - установка длины строки - 80 символов

:set ts=4 - установка размера табуляции

:set ai - в начале новой строки отступ копируется из предыдущей

:set nu - включаем нумерацию строк

:set nonu - выключаем нумерацию строк

:set t_Co=256 - влючаем 256 цветов

:colorscheme delek - меняем цветовую схему

:set bg=dark - темный фон

:syntax on - подсветка синтаксиса

:set incsearch - показывать первое совпадение при наборе шаблона

:set nohlsearch - подсветка найденного

:set smartindent - автоматические отступы

Файл настроек

Скрипт, который выполняется при запуске редактора vim находится в домашней директории пользователя и называется **.vimrc**.

В нём можно задать некоторые дополнительные настройки, например:

colo evening

filetype plugin on

set grepprg="grep -nH \$*"

filetype indent on

Примеры этого файла поставляются вместе с дистрибутивом редактора.

Дополнительная информация

Подробную информацию можно получить из встроенной справки с помощью команды **:help** или на страницах мануала **man vim**