

ИТМО  
Факультет ПИиКТ  
Системное программное обеспечение

Самостоятельная часть тестирования по редакторам vi и  
emacs

Нестеров Дали Константинович  
Группа Р3202

Санкт-Петербург  
2019 год

# Vi и Vim

Vim – это улучшенная версия vi. Он включает в себя такие улучшения, как:

- Визуальный режим
- Подсветка синтаксиса
- Можно подключать плагины
- Встроенный режим сравнения двух файлов (vimdiff)
- Работа в нескольких окнах
- Работа с файлами из архивов
- Можно много раз сделать отмену действий

## Режимы

**Нормальный режим** – открывается при запуске vim, из него можно запускать любые редактирующие команды, а также переходить в другие режимы. Для того, чтобы перейти в него нужно нажать Esc (иногда два раза).

**Режим ввода** – набираемые клавиши воспринимаются как обычные символы и добавляются в текст. Чтобы перейти в него нужно нажать i из нормального режима.

**Командный режим** – позволяет выполнять операции по работе с файлами, настраивать редактор, выполнять поиск и замену, запускать команды внешней оболочки и т.д. Чтобы перейти в него нужно нажать : из нормального режима.

**Визуальный режим** – позволяет выделять текст и выполнять операции по отношению ко всему выделенному тексту. Чтобы перейти в него нужно нажать v из нормального режима.

**Режим поиска** - /<text> чтобы искать <text> в файле. Ищет от курсора и до конца файла.  
?<text> ищет от курсора до начала файла.

## Навигация

Все эти команды нужно выполнять в нормальном режиме.

h – сместить курсор влево

j – курсор вниз

k – курсор вверх

l – курсор вправо

Ctrl-f – на экран вниз

Ctrl-b – на экран вверх

Ctrl-d – на пол экрана вниз

Ctrl-u на пол экрана вверх

Ctrl-y на строку вверх без изменения положения курсора

Ctrl-e на строку вниз без изменения положения курсора

O – к началу строки

^ – к первому не пробельному символу

\$ – к концу строки

w – к следующему слову  
b – к предыдущему слову  
e – к концу слова  
{ – на абзац вверх  
} – абзац вниз  
gg – в начало файла  
G – в конец файла  
<number>G – перейти на строку с номером <number>  
z- – сдвинуть экран так, чтобы строка с курсором оказалась внизу  
z. – сдвинуть экран так, чтобы строка с курсором оказалась посередине  
z[Enter] – сдвинуть экран так, чтобы строка с курсором оказалась вверху  
H – переместить курсор на верхнюю строку экрана  
M – переместить курсор на среднюю строку экрана  
L – переместить курсор на нижнюю строку экрана  
% – если курсор стоит на скобке, перемещает курсор к соответствующей закрывающей/открывающей скобке.

## Ввод текста

Существует множество способов перейти в режим ввода текста:

i – с текущей позиции (перед курсором)  
a – после курсора  
I – с начала строки  
A – с конца строки  
o – с новой строки под курсором  
O – с новой строки над курсором  
R – перейти в режим замены. Набираемый текст будет заменять предыдущий.

## Удаление, копирование, вставка

x – удалить символ под курсором  
X – удалить символ перед курсором  
d – используется совместно с командами перемещения:  
dw – удаляет символы с курсора до начала следующего слова  
de – удаляет символы с курсора до конца текущего слова  
d\$ – удаляет символы с курсора до конца строки  
diw – удаляет слово под курсором  
dd – удаляет текущую строку  
db – удаляет символы с курсора до начала слова  
d0 – удаляет символы с начала строки и до курсора  
c – команда, аналогичная d, но после удаления переходит в режим ввода текста. Команды аналогичны d, за исключением:  
cc – удаляет текущую строку и переходит в режим ввода  
y – команда, аналогичная d, но текст не удаляется а копируется в буфер.

yy – скопировать текущую строку

r<sym> – заменяет символ под курсором на <sym>

p – вставить содержимое неименованного буфера после курсора. В неименованный буфер записывается текст, который был удален или скопирован, такими командами как x, d, c, y.

P – вставить содержимое буфера перед курсором.

J – слияние текущей строки со следующей.

## Повторение действий

Перед командой можно ввести число, тогда команда исполнится заданное количество раз.

Например, 7dd удалит 7 строк. 22ihello[esc] Напишет слово hello 22 раза. 88j перенесет курсор на 88 строк вниз и т.д.

s – удаляет указанное число символов и переходит в режим ввода. Так 6s удалит 6 символов начиная с текущей позиции.

## Поиск и замена

/<text> – ищет <text> начиная с курсора и до конца файла. При достижении конца файла возвращается в начало.

n – следующий результат поиска

N – предыдущий результат поиска

?<text> – как / только ищет в обратном направлении

:%s/text1/text2/g – заменить все text1 на text2 во всем файле

:%s/text1/text2/g – заменить все text1 на text2 во всем файле, но спрашивать подтверждение каждый раз

## Отмена изменений, повторение, изменение регистра

u – отмена последней команды

U – отмена всех изменений в текущей строке

Ctrl-r – вернуть изменение

. – повторить последнюю команду (в том числе ввод текста)

~ – поменять символ под курсором на противоположный регистр.

## Выход, запись

ZZ (или :wq) – сохранить и выйти

ZQ (или :q!) – выйти без сохранения

:w – сохранить файл и не выходить

:w <file> – сохранить файл как <file>

:e! – отменить все изменения в текущей сессии

:qa! – выйти из всех файлов без сохранения

:q – выйти. Выдаст ошибку, если были внесены изменения.

## Исполнение внешних команд

`:!<command>` – исполнит внешнюю команду `command` (Например, `:!ls`)

`:sh` – перейти в shell. Чтобы вернуться в редактор, нужно ввести `exit`.

`:r <file>` – вставить текст из файла `<file>` на место курсора.

`:r!<command>` – вставить результат команды `<command>` на место курсора.

## Изменение параметров редактора

`:set ic` – поиск будет выполняться регистронезависимо

`:set noic` – поиск зависит от регистра

`:set list` – отображать непечатные символы

`:set nolist` – не отображать непечатные символы

`:set number` – нумерация строк

`:set nonumber` – отключить нумерацию строк

`:set wrap` – перенос слов

`:set nowrap` – длинные слова не переносятся

## Работа с несколькими файлами

Окно vim'a можно разбить на несколько окон и работать с разными файлами на одном экране или над разными частями одного файла синхронно.

`:new <file>` – открыть файл `<file>` в новом окне

`Ctrl-w` – переместиться к следующему окну

`Ctrl-w q` – закрыть текущее окно

`:split` – разбить текущее окно по горизонтали

`:vsplit` – разбить текущее окно по вертикали

`:only` – закрыть все окна кроме текущего

Альтернативой являются вкладки

`:tabnew` – открывает файл в новой вкладке

Между вкладками можно перемещаться с помощью:

`gt` – перейти к следующей вкладке

`gT` – перейти к предыдущей вкладке

При запуске Vim можно задать несколько файлов в аргументах. По умолчанию следующий файл будет открываться при закрытии предыдущего, но если ввести `:tab` после запуска, то все файлы откроются в разных вкладках.

`vim -o file1 file2` откроет файлы и разместит их в одном окне разбив по горизонтали. Если задать ключ `-O`, то окно разобьется по вертикали.