

Vim Cheat Sheet

Борис Липин <lonely.ruyk@mail.ru>

Сыктывкарский Государственный Университет
Лаборатория Программирования и Прикладной Математики
<http://amplab.syktu.ru>

1. Режимы VIM

Vim поддерживает несколько режимов работы. Это означает что действие клавиш и их комбинаций различается в зависимости от того в каком режиме вы сейчас находитесь.

1.1. Командный режим

Режим по умолчанию. Из него осуществляется переход в другие режимы и выполнение команд. Клавиша перехода - **ESC**. Все команды из разделов 2 - 10 выполняются из этого режима

1.2. Режим ввода

Клавиши перехода:

i - ввод с текущего символа

a - ввод со следующего символа

o - ввод со следующей строки

1.3. Режим выделения

Клавиша перехода - **v**.

1.4. Режим замены

Клавиша перехода - **R**.

2. Работа с файлами и сессиями

:mksession - создать файл сессии (сохраняется информация о настройках и всех открытых файлах и буферах).

:source Session.vim - загрузить файл сессии (все настройки и открытые в прошлый раз окна восстанавливаются).

:eFILE - открывает FILE на редактирование.

:w[FILE] - записывает сделанные изменения в открытый файл или в FILE если такой указан.

:q - выходит из vim. При **:q!** выходит без записи сделанных изменений.

Ремарка: можно миксовать команды. Например **wq!**.

3. Перемещение

Влево/вниз/вверх/вправо

h / j / k / l

Ремарка: Плюс в том что при такой раскладке пальцы всегда лежат на большинстве нужных клавиш (j/k/l) и не нужно таскать руку от букв до стрелок

Перемещение на начало слова

следующего / прошлого

w / b

\$ - перемещение в конец строки.

^ - перемещение в начало строки.

{NUM}gg - переход на {NUM} строку (например **55gg** переход на 55 строку).

G - переход в конец файла.

4. Отмена действий:

u - отменяет предыдущее действие. Может применяться несколько раз. Аналог комбинации **ctrl + z**

ctrl + r - отмена отмены предыдущего действия. Аналог комбинации **ctrl+shift + z** во многих редакторах

5. Поиск

/[word] - поиск слова {word} в текущем файле от курсора к концу файла.

?[word] - поиск слова {word} от курсора к началу файла.

n - найти следующее слово.

N - Найти предыдущее слово

f{letter} или **F{letter}** - перемещает курсор на следующую (или предыдущую если F) букву {letter} в текущей строке

t{letter} или **T{letter}** - перемещает курсор за один символ до следующей (или предыдущей если T) букву {letter} в текущей строке

6. Редактирование

{NUM}dd - удаление {NUM} строк

{NUM}cc - аналог dd но после удаления переводит редактор в режим ввода.

{NUM}yy - копирование {NUM} строк в буфер обмена.

p - вставка из буфера в то место где стоит курсор

7. Ещё пара слов о командном режиме

Можно относиться к командам в vim как к предложениям.

Есть глаголы например (d, f, g). А так же есть и существительные, например (w, буквы, числа).

Примеры составления предложений:

- Delete till find 'b' (удалить всё до символа b) и соответствующая команда - **dtfb**
- Mark from there to 22 line (выделить всё начиная с текущего символа и до 22 строки). Команда, это реализующая **vg22g**.

8. Вызов внешних команд

!:cmd вызывает shell команду cmd для обработки переданных данных. Если данных не передано показывает вывод команды на экран.

Примеры:

!ls - показывает файлы в текущей директории.

:%!wc -w - считает количество слов в файле.

Делаем vim hex редактором:

:%!xxd - конвертирует файл в hex представление.

:%!xxd -r - конвертирует файл обратно.

9. Прочие полезности

:E - открывает меню выбора файла.

:setlocal - показывает текущие локальные переменные.

10. Путешествие по тегам

Файл тегов можно создать с помощью команды **ctags -R**.

ctrl +] - идти на определение тега.

g] - то же самое но показывает меню выбора если определений несколько.

ctrl + t - вернуться обратно.

11. Файл настроек vimrc.

При запуске редактор считывает настройки, сохранённые в файле **~/.vimrc** (можно сменить место поиска файла настроек запуская vim с ключём **-u path_to_conf.**) Те же команды можно установить для текущей сессии vim в командном режиме после двоеточия (Пример **:set nu**).

Для установки опций используется ключевое слово **set**. причём если мы хотим выключить опцию то перед её именем мы пишем **no**.

Пример:

set nu - включает нумерацию строк vim.

set nonu - выключает нумерацию.

Пара других полезных опций:

set hls (highlight search) - подсвечивает искомые слова.

set ic (ignore case) - игнорирует регистр букв при поиске.

syntax on - включает подсветку синтаксиса.

12. Справка

vim содержит довольно полную документацию, которой можно воспользоваться в любой момент времени. Для просмотра полной справки нужно набрать **:help**. Для получения справки по интересующей команде достаточно набрать **:help commandname** в командном режиме. По оглавлению можно перемещаться с помощью клавиш из раздела "путешествие по тегам".

13. Полезные дополнения vimrc

- работа с файлами в виндовой кодировке

```
set fileencodings=cp1251,utf-8
```

```
if has("x11") == 0
```

```
    "set termencoding=utf-8
```

```
    set enc=cp1251
```

```
endif
```

```
set enc=cp1251
```

- Window-изация vim. Добавление привычных комбинаций **ctrl + c** (режим вставки), **ctrl + v** (командный режим), **ctrl + a** (командный режим):

```
function! CopyAll()
```

```
    let ln = line('.')
```

```
    normal gg"+yG
```

```
    exec ln
```

```
endfunction
```

```
vmap <C-c> "+y
```

```
map <C-v> <esc>"+gP
```

```
map <C-a> <esc>:call CopyAll(<cr>
```

- Маппинг русских клавиш чтобы не приходилось переключать раскладку для командного режима.

```
set langmap=йq ,цw ,уe ,кr ,et ,ну ,гу ,ші ,що ,зр ,фа ,ьs ,bd ,af ,pg ,ph ,oj ,лк ,дl ,яз ,чх ,сс ,mv ,иб ,тп ,ьm ,ЙQ ,ЦW ,УE ,KR ,ET ,НУ ,ГУ ,ШІ ,ЩО ,ЗР ,ФА ,ЫS ,BD ,AF ,PG ,PH ,OJ ,ЛK ,ДL ,ЯZ ,ЧX ,CC ,MV ,ИБ ,TN ,ЬM ,x[ ,X{ ,ь\ ,ж\; ,Ж: ,э' ,Э\" ,б\ ,Б< ,ю. ,Ю>
```

14. Плагины vim

Для упрощения работы в VIM в интернете можно найти множество плагинов. Установка их довольно проста (обычно нужно просто поместить файлы в папку **~/.vim/**).

Из них можно выделить

- **omnicppcomplete**. Плагин для автодополнения имён структур, функций. Информацию о которых получается из файла тегов, который можно сгенерировать с помощью утилиты **ctags**.
- **snipmate**. Плагин содержащий в себе снипеты (короткие но часто используемые куски программ, например определение функции main) для многих языков программирования. Добавить свои снипеты или изменить существующие можно в файлах **.vim/snippets/***.
- **NERDtree**. Плагин отображает дерево каталогов начиная с выбранного корня. Довольно удобно для просмотра чужих исходников.

15. Используемые источники

```
vim -c ":help"
```

<http://ru.wikibooks.org/wiki/Vim>

Steve Oualline. Vi iMproved (VIM)

<http://www.youtube.com/watch?v=SkdrYWhh-8s>