Skill Pill: Editor

Lecture 2: Vim a serious text editor

Could have been worse, could have been emacs

Valentin Churavy

Okinawa Institute of Science and Technology
valentin.churavy@oist.jp

March 17, 2016

Schedule

1. Introduction

2. Vim basics
   - Modes
   - Movement
   - Editing
   - Advanced movement

3. Configuration and plugins

Why?!

As a programmer and a scientist I spend my day typing. Think about how much you are typing each day.

Why?!

As a programmer and a scientist I spend my day typing. Think about how much you are typing each day.

Vi was writing at a time where the only interface to the computer was a text terminal and as such efficient text manipulation was quite important.

A short history of computation

The competition

There are a few decent editors out there.

The competition

There are a few decent editors out there.

- Graphical Editors
  - Sublime Text
  - Atom
  - gedit, kedit, notepad++
- Terminal based editors
  - Vi family, vi, vim, neovim
  - Emacs
  - nano

The competition

There are a few decent editors out there.

- Graphical Editors
  - Sublime Text
  - Atom
  - gedit, kedit, notepad++
- Terminal based editors
  - Vi family, vi, vim, neovim
  - Emacs
  - nano

All of these editors will get the job done. After all we are just writing text, but only a few will increase your efficiency and productivity.

The ~~Browser~~ Editor wars

There once was a holy fight between the Church of Emacs with St IGNU-cius (Richard Stallman) as its leader and the following of the beast (vi-vi-vi 666).

Nowadays the war is over and Emacs has lost.

Nano: The easy way out.

If you want a simple editor for the terminal without a lot of bells and whistles use nano. [ctrl] $= \wedge$

Nano: The easy way out.

If you want a simple editor for the terminal without a lot of bells and whistles use nano. [ctrl] $= \wedge$

Example

Time for our first excursion.

Getting started

vim	Open a terminal instance of vim
gvim	Open a graphical instance of vim
:help	The invaluable help menu

Modes

Vim has several modes that are optimized for different task:

- **normal** For navigation and manipulation of text.
- **insert** For inserting new text.
- **command-line** For entering editor commands
- **visual** Fast large style text manipulation

Normal is the mode in which you start, when Vim is opening up, and you can always go back there, via ESC.

Writing text

$$i \rightarrow [\text{text}] \rightarrow [\text{ESC}]$$

Saving text and leaving vim

:w Writing the current buffer to a file.

:q Leave vim

:q! Forcefully leave vim

Vim modes in slow motion

```
vim example.tex
  Normal mode
i -> Insert mode
[ESC] -> Normal mode
: -> command-line mode
w [Enter] -> Normal mode
:q
```

Movement

In normal mode we use h, j, k, l to move up, left, right, down.

Movement: words

word and WORD

A word consists of a sequence of letters, digits and underscores separated with white space.
A WORD consists of a sequence of non-blank characters, separated with white space. An empty line is also considered to be a WORD.

w|W	move words—WORDS forward [exclusive]
e|E	move to the end of word—WORD forward [inclusive]
b|B	move words—WORDS backwards [exclusive]
ge|gE	move to the end of word—WORD backwards [inclusive]

Trying out the differences

0 Move to the first character of the line.

$ Move to the end of the line

x- Move x lines upwards

gg goto line, defaults to first line

G goto line, defaults to last line

Counts

Most commands in normal mode take a count:

$3w$ & $15j$

Editing I

- i Insert text before the cursor
- a Append text after the cursor
- o Begin a new line below the cursor
- I Insert text at the beginning of the line
- A Append text to the end of the line
- O Begin a new line above the cursor

Editing II

dx	Delete text that motion x moves over
cx	Delete text that motion x moves over and start insert
dw	Delete word
d$	Delete till end of line
dd	Delete line
3dd	Delete 3 lines

Editing III

u	Undo last change
U	Undo all changes to current line
Y	Yank lines
y*x*	Yank over movement
yy	Yank current line
p	Paste after cursor
P	Paste before cursor

Advanced movement

(|) Move sentence wise

{|} Move paragraph wise

[[|]] Move section wise

ci" change inside "

f|F move to occurrence of character to the right|left

t|T move till before occurrence of character to the right|left

Search and replace

/ search forwards

? search backwards

:%s/foo/bar/g Search foo and replace with bar in the whole file

Vimgolf

Vimgolf

Try solving the example below with what you learned today, while using as little keystrokes as possible.
For more examples see `vimgolf`

Start file:

```
name=www-data, groups=developer
```

Target:

```
name=developer, groups=www-data
```

Vimgolf

Vimgolf

Try solving the example below with what you learned today, while using as little keystrokes as possible.
For more examples see `vimgolf`

Start file:

```
name=www-data, groups=developer
```

Target:

```
name=developer, groups=www-data
```

Solution:

```
fwdt,fdPldwF,PZZ
```

Configuration

Configuration

The configuration file lives at ∼/.vimrc.
Comments "

```
set nocompatible  " No backwards compatibility
set relativenumber  " hybrid line numbers
set number
set ruler
syntax on
```

The case for soft-stops

```
set smartindent
set tabstop=2
set shiftwidth=2
set expandtab
```

Don't use tabs

Tabs are rendered differently between users and consistently formatting of source code is best done with spaces (soft-tabs).

Colorschemes

```
set background=dark
colorscheme solarized
```

Plugins

Vundle.vim	plug-in manager for Vim
vim-sensible	Sensible defaults
fugitive.vim	All the Git
syntastic	Syntax checker
vim-gitgutter	Git status

Neovim

The `neovim.io` project aims to rewrite vim for modern times. It aims to be a drop-in replacement, while improving the underlying architecture. As an example it uses asynchronous IO enabling plugins to be run in the background.
Everything else is kept the way it is and as such switching between neovim and vim should be quite painless.

Final words

You can use vim like a normal text editor and only ever use *i*, but the more you learn the more productive you are. Don't be scared and go and try vimtutor.