

Source Code Editors. Vim



Vlad Dogaru

ddvlad@rosedu.org

Alex Juncu

alexj@rosedu.org

June 18, 2013

ROSEdu

1 More than Notepad

2 Configuration

3 Multiple files

4 Bigger projects

5 Where to next?

FFS, Why Vim?

FFS, Why Vim?

- because that's what we know and use :-)

FFS, Why Vim?

- because that's what we know and use :-)
- try Emacs, seriously

FFS, Why Vim?

- because that's what we know and use :-)
- try Emacs, seriously
- Vim and Emacs are not that different

FFS, Why Vim?

- because that's what we know and use :-)
- try Emacs, seriously
- Vim and Emacs are not that different
- need to get to the beginning of the word
 - vi: b (be sure to Esc if needed)
 - emacs: Alt-b
- need to execute the same command 10 times
 - vi: 10<command>
 - emacs: C-u10<command>

FFS, Why Vim?

- because that's what we know and use :-)
- try Emacs, seriously
- Vim and Emacs are not that different
- need to get to the beginning of the word
 - vi: b (be sure to Esc if needed)
 - emacs: Alt-b
- need to execute the same command 10 times
 - vi: 10<command>
 - emacs: C-u10<command>

More than Notepad

- hjkl

More than Notepad

- h j k l **Use them**

More than Notepad

- h j k l **Use them**
- Insert mode

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what?*

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what*?
 - it matters what follows after d: d\$, df(, di(, dG, d`a

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what*?
 - it matters what follows after d: d\$, df(, di(, dG, d`a
- copy-pasta – it's bad

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what*?
 - it matters what follows after d: d\$, df(, di(, dG, d`a
- copy-pasta – it's bad **But we have it**

More than Notepad

- hjkl **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what*?
 - it matters what follows after d: d\$, df(, di(, dG, d`a
- copy-pasta – it's bad **But we have it on steroids**

More than Notepad

- h j k l **Use them**
- Insert mode **It's evil**
- advanced movement (^, \$, f, F, G, gg, 10G, :10, % ctrl-o)
- compound commands
 - d – delete
 - delete *what*?
 - it matters what follows after d: d\$, df(, di(, dG, d`a
- copy-pasta – it's bad **But we have it on steroids**
 - "zyi[, "zp
 - :help registers

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point
 - but give credit where it is due

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point
 - but give credit where it is due

$$\frac{\textit{picture}}{\textit{words}} = 1000$$

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point
 - but give credit where it is due

$$\frac{\textit{picture}}{\textit{words}} = 1000$$

But be careful!

- You still need to know your way in a default environment.

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point
 - but give credit where it is due

$$\frac{\textit{picture}}{\textit{words}} = 1000$$

But be careful!

- You still need to know your way in a default environment.
- Bonus points if your config is usable by others.

Configuration

- defaults just won't cut it
- it's hard to write a config from scratch
- use other configs as a starting point
 - but give credit where it is due

$$\frac{\textit{picture}}{\textit{words}} = 1000$$

But be careful!

- You still need to know your way in a default environment.
- Bonus points if your config is usable by others.
 - *Don't* remap i.

Buffers

- buffer == an open file (Vim speak)
- buffers are not always displayed
- `vim a.txt b.txt c.txt` – you see `a.txt`
- use `:next`, `:prev` to change buffers
- or `:buffer N` to skip to buffer N
- `:e filename` creates a new buffer
- `:ls` to list all buffers
- `:help hidden`

Windows

- create: `:split (Ctrl-W s)`, `:vsplit (Ctrl-W v)`
- move between windows: `Ctrl-W [hjkl]`
- resize: `Ctrl-W [+<>]`
- “maximize”: `Ctrl-W _` or `:only`

Windows

- create: `:split (Ctrl-W s)`, `:vsplit (Ctrl-W v)`
- move between windows: `Ctrl-W [hjkl]`
- resize: `Ctrl-W [+<>]`
- “maximize”: `Ctrl-W _` or `:only`

Tabs

- tab == one or more open windows

Tabs

- tab == one or more open windows
 - preferable more than one
 - usually wrong: one file, one tab

Tabs

- `tab ==` one or more open windows
 - preferable more than one
 - usually wrong: one file, one tab
- new tab: `:tabnew`
- next/previous: `gt/gT`
- go to tab N: `NgT`

Sessions

- so I got this setup: 6 tabs, dozens of open files... the works
- `:mksession` \Rightarrow `Session.vim`
- later: `vim -S`

ctags/cscope

- C source code analyzer
- can be integrated with Vim
- needs a little config

ctags/cscope

- C source code analyzer
- can be integrated with Vim
- needs a little config
- **example:** Pidgin source code, `pidgin/gtkblist.c`

Autocomplete

- `Ctrl-N` and `Ctrl-P` in Insert mode
- useful not only for code
- can look inside headers (C, C++)

Autocomplete

- `Ctrl-N` and `Ctrl-P` in Insert mode
- useful not only for code
- can look inside headers (C, C++)
 - but that's about as smart as it gets

Autocomplete

- `Ctrl-N` and `Ctrl-P` in Insert mode
- useful not only for code
- can look inside headers (C, C++)
 - but that's about as smart as it gets
- fancier stuff: OmniComplete, YouCompleteMe

Where to next?

Some teasers:

- editing and moving: you never know enough
- editing inside archives
- modeline
- search and replace – sed in a can
- code folding
- macros
- vimscript
- endless plugins, e.g. the NERD Tree
- <http://vim.wikia.com/>