Vim Basics

Jacob Farkas
Cal Poly Linux Users Group

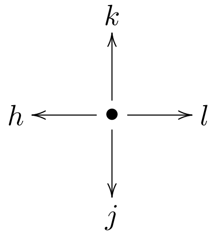
May 25, 2004

1 How to Use Vim Without Hurting Yourself

1.0.1 What's a Modal Editor?

- Modal editors have two modes: **input** and **command**.
- In **input** mode the keys you type are entered in the file.
- In **command** mode the keys you type move you around in the file, change the file, or perform other actions.
- To get out of insert mode, just hit `<esc>`
- To enter insert mode, type an insert command (we'll go over these soon)
- Vim will tell you if you're in insert mode by displaying `-- INSERT --` in the bottom left corner.
- If `-- INSERT --` isn't displayed you're in command mode

1.0.2 How to Move Around



- You can cheat and use the arrow keys, which work during insert mode, but using `hjkl` will help you keep track of modes better.
- You could draw that picture up and put it by your screen or tatoo it on your hand or something, but there's an easier way: Nethack!

1.0.3 How to Edit Text

- `i` **i**nsert before the current character
- `r` **r**eplace the text under the cursor
- `u` **u**ndo the last action
- `^r` **r**edo the last action

1.0.4 How to Save and Exit

- `ZZ` save and exit
- `:w` **w**rite the file to disk
- `:q` **q**uit (`:q` only works if the file has already been saved)
- `:q!` **q**uit without saving**!**
- `:wq` **w**rite the file and **q**uit

1.0.5 Help Yourself

- `:help` bring up the online help
- Move around the help files like any other file
- To follow a `|link|`, put your cursor over the link and press `^]`
- `:q` **q**uits the help window

2 Cool Vim Tricks

2.0.6 Repeated Commands

- Almost any command can be prefixed with a number of times to perform that command.
- Move around: `5j`, `5k`
- You can also give a command along with a movement and a count
- `d5j` Delete the next 5 lines

2.0.7 Registers

- `yy` **y**ank (copy) the current line into a register
- `dd` **d**elete (cut) the current line into a register
- `p` **p**aste the line in the register after the current line
- `P` **p**aste the line in the register before the current line

These can be combined with movement commands also:

- `y`*n*`j` **y**ank the next *n* lines into a register
- `d`*n*`j` **d**elete the next *n* lines into a register
- `:reg` Displays all registers and their contents

2.0.8 Handy Shortcut Keys

- Moving among words
  - `w` : for**w**ard one word
  - `b` : **b**ackward one word
  - Combine them with `d`: `dw` and `db` delete next/previous word
- Getting around in a file faster
  - ˆ`d` **d**own half a page
  - ˆ`u` **u**p half a page
- Working with the beginning and end of a line
  - `$` go to the end of the line,
  - `^` go to the beginning of the line
  - `gm` **g**o to the **m**iddle of the line
  - `I` insert at the end of the line
  - `A` insert at the beginning of the line
- Grab Bag (my favorite commands)
  - `:TOhtml`
  - `>>` indent the current line
  - `.` repeat the last command
  - toggle case
  - `%` find matching paren/comment/brace
  - ˆ`a` increment number under cursor
  - ˆ`x` decrement number under cursor

2.0.9 Recording

- To begin recording, type `q`*n* in command mode. The *n* is the number or letter to assign the macro to
- Do whatever it is you need to do, then type `q` again in command mode to stop recording
- `recording` will show up in the bottom left corner when you're recording
- To play back a macro, type `@`*n* where *n* is the location you recorded the macro to

3 Vim and Multpile Files

3.0.10 Split Windows

- Creating a split window
- Moving between split windows
- Closing a split window
- Using split windows as bookmarks

3.0.11 Opening a New File in a Split Window

4 Getting Friendly With /.vimrc

- .tex file stuff
- Syntax highlighting

5 Basic Regular Expressions