VIM – For Text Editing

Introduction

- What is VIM?
  - Terminal-based Text Editor
  - Very powerful Text Editor
- No kidding – Huge learning curve
- Most people find it hard only because it’s different
  - Put pre-conceived notions about text editors away
- Work a little now, be efficient the rest of your life

Introduction

- Why learn and use VIM?
  - Accessibility
    - Every Unix/ Linux machine you ever encounter almost guaranteed to have VI or VIM installed
  - Speed
    - Known to use far fewer keystrokes to achieve the same result compared to other editors
    - Designed to minimize moving your hands around
  - Power
    - Lets you do complicated tasks with a few keystrokes that other text editors might now even have by default

Philosophy of VIM

- By having a separate ‘typing’ mode, all the characters on the keyboard become available for shortcuts on their own
- This is what reduces the keystrokes
- Example: copy line 5 and paste it on line 30
- Regular editor: scroll to line 5 (5 keystrokes), select the line (2 keystrokes), ctrl-c (2 keystrokes), scroll to line 30 (25 keystrokes), ctrl-v (2 keystrokes)
  - (5xdown)(shift+end)(ctrl+c)(25xdown)(ctrl+v)=36 strokes
- VIM:
  - 5Gyy30Gp = 8 strokes

Let’s start

- Launch VIM in a terminal
  - vi or vim will start VIM with a new file
  - vi *filename* or vim *filename* will open file in VIM
- Now, DON’T touch anything!

First Look

- VIM works in one of three modes at any time
  - Normal
  - Insert
  - Command

First Look (cont.)

- VIM works in one of three modes at any time
  - **Normal (enter by pressing ESC)**
    - This is the mode you should always stay in
    - No typing here
    - Browse the file
    - Perform functions like cut, copy, paste, search, replace, etc.
    - Whatever else you do, always get back to this mode
  - **Insert (enter by pressing i or other variants)**
    - Enter this mode when you want to type where your cursor is
    - Do not even delete, or try to move around
    - As soon you finish typing go back to normal mode
  - **Command (enter by pressing :)**
    - For commands such as save, quit, etc.

Some Important Instructions

- Do not use / touch the cursor / arrow keys!!
- Do not use the mouse!!
- Forget that they exist for now!!
- While it may seem like a handicap for now, the promised speed and power of VIM can only be achieved if you adhere to these rules

Normal Mode (Basic Movement)

- Use the h,j,k,l keys to move one character left, down, up, right
    - *Practice till you have it in your muscle memory*
- w,b keys to move one word forward or backward
- 0, $ to move to the beginning and end of the line
- {, } to move to next and previous paragraph
- gg, G to move to the beginning and end of the document

Normal Mode (Basic Movement)

- Now try the same shortcuts but precede them with a number
- For example:
  - 5h takes you 5 characters to the left
  - 2w takes you 2 words forward
  - 7b takes you 7 words backward
  - 3j takes you 3 characters down
  - 5G takes you to the 5th line

Normal Mode (Searching)

- fx and Fx jump to next and previous occurrence of character x in current line
  - For example
    - fa will move you to the next ‘a’
    - Fg will move you to the previous ‘g’
- */searchterm* will find the next occurrence of *searchterm* going forward in the document
- ?searchterm will find the previous occurrence of *searchterm* going backward in the document
- n will keep repeating the previous search

Normal Mode (Cut, Copy, Paste)

- Copy
  - yy yanks (copies) the current line
    - (3yy will yank 3 lines, etc.)
  - y$ yanks from cursor to end of line
  - yw yanks the current word
- Cut
  - dd deletes (cuts) the current line
    - (3dd will cut 3 lines, etc.)
  - d$ deletes (cuts) from cursor to end of line
  - dw deletes (cuts) current word
  - x deletes (cuts) current character
- Paste - p

Normal Mode (Editing)

- rx replaces the current character with x
  - Example: rz will replace current character with ‘z’
- cw*NewWord*(esc) replaces the current word with *NewWord*
- cc*NewLine*(esc) replaces the entire line with *NewLine*
- *sNewText*(esc) replaces the current character with *NewText*
- J join the next line to this one with a space
- u will undo, Ctrl-r will redo
- . will repeat the last command

Normal Mode (Visual Mode)

- Ctrl-v to start visual mode
- Use movement to select text
- Then use one of the following:
  - y to yank (copy) selected text
  - d to delete selected text
  - ~ to switch case
  - >, < to shift text right or left

Insert Mode

- Press i from normal mode
  - (More ways to enter insert mode we’ll discuss later)
- Start typing text
- Remember you cannot move cursor around, or delete anything
- To do anything else, except typing press (esc) to go back to normal mode
- As soon as you are done typing press (esc) quickly and always stay in normal mode

Command Mode

- : followed by a command lets you run commands to alter the file and other VIM related options
- Examples:

:w	Writes (saves) file to disk
:w *filename*	Writes file to disk as *filename*
:wq OR :x	Writes and quits VIM
:q!	Quits without saving
:q	Quits only if there are no unsaved changes

Command Mode

- Set command is very useful to set various parameters of VIM
- Examples

:set number :set nonumber	Show line numbers Hide line numbers
:set wrap :set nowrap	Wrap long lines Do not wrap
:set ignorecase :set noignorecase	Ignore case in search Match case in search