Introduction to Linux
Part 2a: the vi editor

Brett Milash and Wim Cardoen
Center for High Performance Computing

Vi editor

- Editor provided with the OS
  - There are other choices such as nano (easy) and emacs (hard)
  - It is better to edit in a linux shell than moving file to your laptop for editing
- To start – command is **`vi`** (usage: `vi filename`)
  - If filename exists, vi will open it up in the editor
  - If filename does not exist, vi will create it and put you in insert mode
- Use arrow keys to move cursor to location
- Two modes – command, input
  - in command mode – the characters typed are interpreted as commands
    - There is also an external command mode – access with ':'
  - In insert mode – the characters typed are added to the file
    - The esc key exists the insert mode
- Many commands, we are just mentioning some basic ones to get you started

Exiting the vi editor

- **ZZ** – will exit and save as same filename
- **:q** – quit; will let you know if you have unsaved changes
- **:q!** - quit discarding changes
- **:wq** – write and quit; same as doing **:w** followed by **:q**
  - saves as same filename
- To change name add new name after the **:w** or **:wq**
- Note – if you **cntl Z** to suspend – vi saves a swap filename to save current status of the edits. File is named .filename.swp. Next time you try to edit this same file it will let you know the swp file exists and asks you how to proceed. If you do not want to keep changes, delete this file

Moving around a file

- **down arrow** or **j** moves down one line
- **up arrow** or **k** moves up one line
- **right arrow** or **l** moves right by one character
- **left arrow** or **h** moves left by one character
- **0** or **^** moves to start of current line
- **$** moves to end of current line
- **w** moves to beginning of next word
- **b** moves to beginning of previous word
- **:0** or **:1** or **1G** moves to start of first line
- **:n** or **nG** moves to start of nth line
- **:$** or **G** moves to start of last line
- **cntl-f (^f)** moves forward one screen (cntl-d moves forward ½ screen)
- **cntrl-b (^b)** moves back one screen (cntrl-u moves back ½ screen)

Inserting or Adding Text

- Remember – **esc** to exit command mode

- **i** – insert at position of cursor
- **I** – insert at beginning of line
- **a** – append after position of cursor
- **A** – append at end of line
- **o** – new line below current line
- **O** – new line above current line

Changing and Deleting Text

- **r** – replace single character
- **R** – replace characters, starting at cursor position, until **esc** hit

- **x** – delete character that cursor is on (**nx** for n characters starting with one cursor is on)
- **dd** – delete current line (can do n lines with ndd)
- **D** – delete from cursor to end of line
- **dw** – delete word

Cutting and Pasting Text

- **Y** – “yanks” current line into buffer (can use **nY** for n lines, current plus following lines)
- **p** – paste lines in buffer after current line (**P** – paste lines before current line)

Other Useful Commands

- /pattern – searches for next occurrence of pattern, then n goes to next occurrence; can also use / and ? To move to previous and next occurrence
- ?pattern – searches for previous occurrence of pattern
- **u** – to undo results of last command (use multiple times to revert back through multiple commands

External commands

- already mentioned some of these in the moving around and exiting vi
- **:s/old_text/new_text/** – replaces next occurrence of old_text in current line
- **:1,$s/old_text/new_text/g** – replaces all occurrences of old_text in file

Exercise – Practice these commands!

- Try the “vimtutor” command
- Try writing some source code (C, python, fortran, etc)
- Try writing your grocery list