

Linux, vim, and git Workshop

A Sequence of (Very) Brief Introductions

Slides: github.com/jleightcap/LinuxVimWorkshop

Wireless Club – Jack Leightcap

January 27, 2020

Workshop Structure

- 15 Minutes – Linux
- 15 Minutes – vim
- 30 Minutes – git
- We have the room booked until 8:30, if you want to delve into anything in more depth, we'd love to stick around!

What is Linux?

- An open source, cross-platform Operating System
 - Linux distributions are completely free to download and use indefinitely
 - Culture of making source code publicly available, if something is broken you almost always have the ability to debug it
- Could be described as an ecosystem of software
 - If you don't like the way some component works, someone probably has already written an alternative
 - Or you have the complete power to write your own implementation
 - With most problems there is probably already a thread somewhere online from 2006 about the exact issue you're having

What is Linux used for, and why should you learn it?

- A lot here at Northeastern! COE/CCS (sorry, *Khoury*) Linux servers for storage space and remote execution, used heavily in Embedded Design...
- A large part of software development takes place using Linux
- Embedded Systems programming heavily relies on familiarity with Linux
- Devices ranging from Android phones to PS4s all run some modified versions of Unix
- Generally most non-desktop processors you can think of run some form of Linux
 - Of all processors (micro or not), any guesses what percentage are used in personal computers?
- The skills learned in one Linux environment are usually extremely transferable to a wide range of environments.



Figure: Embedded Linux on business card for \$1.40



Figure: 1/3 of Microsoft Azure servers run Linux!

Linux Basics - The Terminal

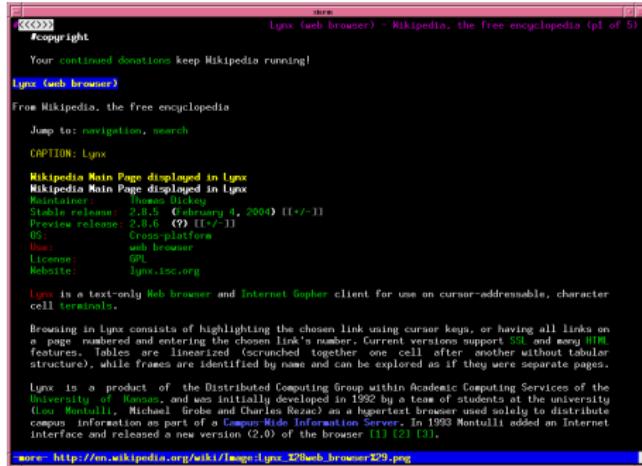


Figure: Lynx - a terminal-based browser

- A method of "communicating with your computer" through text-based commands rather than a point-and-click GUI
- Can be used for
 - Navigating your filesystem
 - Creating, deleting, or modifying files
 - Browsing the internet
 - Checking email
 - Basically, anything you would use a computer for
- You can "live in the terminal"

Linux Basics – File Organization 1

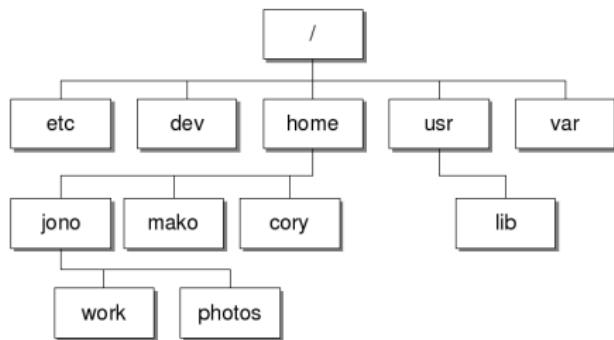


Figure: Basic Linux Directory Structure

- `cd [directory]`
move your current location to directory
- `mv [source] [destination]`
move a file from source to destination
- `pwd`
print working directory
- `ls`
list the files in the current directory

Linux Basics – File Organization 2

ABSOLUTE PATH

- `/home/user`
- The current directory
- The parent directory

RELATIVE PATH

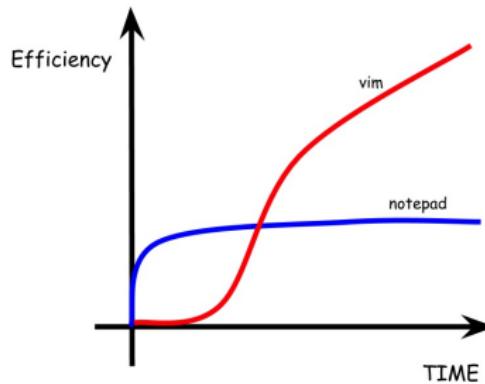
- `~`
- `.`
- `..`

FILE MANIPULATION

- `touch [file]` – create an empty file named `file`
- `rm [file]` – remove `file` (forcibly!)
- `mkdir [directory]` – make an empty directory named `directory`
- `rmdir [directory]` – remove a directory (checks if empty)
- `vim [file]` – edit a file in vim (foreshadowing?)

Vim! What is it?

- Vim is a terminal-based and standalone text editor
- Entirely based on keyboard interaction rather than point and click
- Hands stay close to the home row of the keyboard throughout text editing
- Frequently voted among the most popular text editors among Linux users



Vim Basics – Modes

Vim has different editing modes for different parts of the typing process,

- *Normal Mode* – the default mode, used for editor commands.
- *Visual Mode* – used to highlight text. Commands executed apply to all highlighted text.
- *Insert Mode* – actually write text from keyboard.
- *Command-line Mode* – execute commands.

To move between modes,

- Normal mode is the default, at any point press `esc`.
- Visual mode is enabled by pressing `v` or `V`.
- Insert mode is enabled by pressing `i` (or other shortcut keys)
- Command-line Mode is enabled by pressing `:`

Vim Basics – Normal Mode

- Typing in this mode doesn't actually insert any text!
- Some basic movements:
 - {h, j, k, l} → {left, down, up, right}
 - w/W – move forward by a word (punctuation, whitespace)
 - b/B – move backward by same rule as w/W
 - e/E – move forward by same rule as w/W, but to the end of word
 - ^/\$ – the start/end of the current line
 - gg/GG – the first/last line of the current file
 - *ngg* – go to the *n*th line of the current file
 - all commands can be prefixed by a number, so 4k would move up 4 lines
- Some basic editing:
 - x – delete character under cursor
 - diw - delete in word, delete word under cursor
 - dd – delete the current line

Vim Basics – Visual Mode

Much simpler than insert mode! Define selection start and end, then execute a command over that selection.

- v - start highlighting at current cursor position by character
- V - start highlighting at the current line by line

Visual mode is generally useful for commands repeated over multiple lines.

- <>/>> – indent back/forward
- d - delete, equivalent to "cut" – move into "clipboard" (buffer in vim)
- y – yank, equivalent to "copy"
- p – paste text in buffer after the cursor

To repeat a command, prefix with a number as mentioned before, or use .

Vim Basics – Insert Mode

Actually input text! There's not much to say about this, so some shortcuts that enter insert mode:

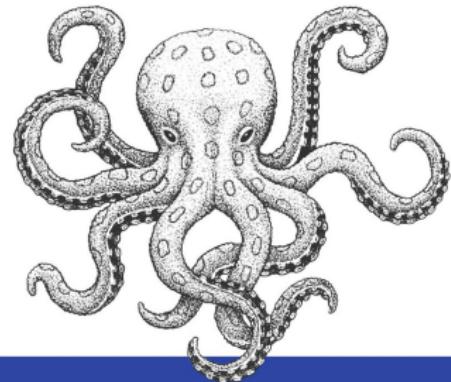
- `a/A` – append, at end of word/line
- `i/I` – insert, at current cursor position, or at beginning of line
- `ct{x}` – change to `x`, delete text from cursor position to `x`, and enter insert mode
- `c^/c$` – change to beginning of line, change to end of line

Vim Basics – Command-line Mode

Enter commands after :,

- `:w`, `:q` – write, quit
- `:! [command]` – run terminal command from inside vim!
- `/[string]` – search for string in current file, `n/N` to navigate instances

Just memorize these fourteen contextually dependant instructions



Exiting Vim

Eventually

O RLY?

@ThePracticalDev

Actually Trying Vim and Linux!

A lecture about Vim is pretty dry, and is something that is learned through practice.

If we have time, some good resources:

- VIM Adventures (vim-adventures.com) – a browser game that visualizes vim commands very well
- `vimtutor` – frequently comes installed alongside vim.
- Vim Tips Wiki (vim.fandom.com) – if you end up going down this rabbit hole, as plenty examples of surprisingly efficient commands.
- Over the Wire (overthewire.org/wargames/bandit/) – SSH based game based on finding keys, targeted at Linux beginners.