Learning Linux with Shogo

Saturday October 30, 2021

S.Toyonaga
3rd year EECS Undergraduate Honours Student

Agenda

- General Introduction - [Who am I?] (5 Minutes)
- Linux Familiarization - [Motivations, Elementary Commands, etc.] (15 Minutes)
- Compiling Code on the EECS Server - [Java, C] (10 Minutes)
- Using the Linux Environment on EECS - [ssh] (10 Minutes)
- Remote Labs Guide - [Basic] (10 Minutes)
- Using Web Submit - [GUI] (5 Minutes)
- Concluding Thoughts - [Wrapping the Ideas Up] (5 Minutes)
- Questions and Answers - [OPTIONAL] (10 Minutes)

Who am I?

- **Third Year Computer Science (Honours) Student**
  - Golden Key Society Member
  - Academic Peer Helper Representative
- **Academic Interests**
  - Database Systems
  - Theory of Computation
  - Design and Analysis of Algorithms
  - Discrete Mathematics
- **Interests**
  - Cycling
  - Cooking
  - PC Gaming
  - Philosophy of the Mind

Linux - Motivations

- **Free, Open Source Operating System**
- **Privacy Friendly**
- **Security**
- **Powerful Terminal**
  - Package Managers (apt, pacman, etc...)
- **Performance**
  - Universality (Distro-Hopping)
  - Hardware Independent

Playing the Devil’s Advocate

1. Has a learning curve {Unix Commands}
2. Lack of available (commercialized) software
3. Poor Game Performance (via 2)
4. Can, “Break” If You’re Not Careful

- One should be very careful before running commands...
  - Use **sudo pacman -Sy**

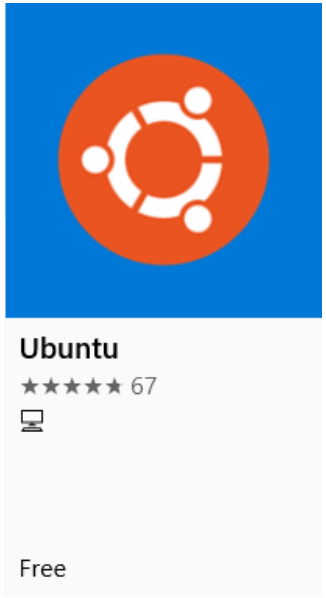
Connecting Linux to EECS (York)

- To use the York University EECS Linux environment, **you must have an EECS account!**
  - https://www.eecs.yorku.ca/activ8
- You will be eligible for an account if you're enrolled in a course that requires EECS privileges:
  - Submissions (Web Submit)
  - EECS Tools (Remote Labs)
- Additionally, you can apply what will be taught in preparation if you install Linux on your local machine!
- Last Resort ⇒ Virtual Machines

Distros - Built for Different People!

Pre-Environmental Setup Considerations

1. Virtualizing Software ⇒ https://www.virtualbox.org/
2. Creating a Bootable Installer ⇒ https://rufus.ie/
3. Installing an Ubuntu terminal on your Computer



Walking Steps - Elementary Commands

Purpose	Instructions
Directory Navigation	pwd, ls, cd
Files	cp, rm, touch, less, cat
Editing	nano, vim
Conversions	convert, ffmpeg
Github	git
Compiling (Java, C)	java, gcc

Manual - Your Source of Learning

- “An interface to the system reference manuals.”
  - Serves as an, **“API”** to any instruction that you need clarification on.
- **Basics:** man instruction_name
- **What will the output of, “man man” be?**

Man Man

Part 1) Directory Navigation

- pwd
  - **“Print name of current/working directory”**
  - Useful for supplying an argument for future commands...
- ls
  - **“List directory contents”**
  - Entries are sorted **alphabetically**
    - -a: Do not ignore entries starting with, “.”
    - -author: List the author(s) of all the files and directories
- cd
  - **“Change the working directory”**
    - .. ⇒ Go back by one directory
    - / ⇒ Go directly to the root directory
    - ~ ⇒ Go directly to the user directory

Part 1) Directory Navigation Illustration

- Let's assume that we start at the following directory:
  - **/home/shogom**
- Running ‘ls’ will output the following:
  - **Desktop Documents Downloads eclipse eclipse-workspace Music Pictures [...]**
- How do we move into the eclipse-workspace directory?
  - **cd eclipse-workspace**
- What is the resulting new directory (via pwd)?
  - **/home/shogom/eclipse-workspace**
- List everything with their respective author:
  - **ls -author**
- Go (atomically) back to /home/shogom:
  - **cd ..**

Part 2) Files

- cp
  - **“Copy files and directories”**
    - cp filename1 filename2 ⇒ Copies the the file and places it in your **current directory**
    - cp filename /directory/ ⇒ Copies the file to the supplied directory.
- rm
  - **“Remove files”**
    - rm filename1
    - rm directory/file.*
- touch
  - Used to:
    - Update timestamps (“access / modify”) of a file
    - **Create Files**
    - **Exercise**: Use the manual to find out how to invoke the touch instruction properly!
- less
  - Returns the contents of a text file to the console / terminal.
- cat
  - **“Concatenate files and print on the standard output.”**
    - **Exercise:** Use the manual to find out how to invoke the concatenate instruction properly!

Part 2) Files Illustration

- Lets assume that we start at the following directory:
  - **/home/shogom**
- Assume that we have two files in the directory with the following directories:
  - **Directory 1** ⇒ BagA ⇒ Shogo.txt
  - **Directory 2** ⇒ BagB ⇒ You.txt
  - **Directory 3** ⇒ CoolPeople ⇒
- **How do we Copy Shogo.txt and You.txt to the CoolPeople Directory using the commands that we've covered so far?**
  - **cp BagA/Shogo.txt CoolPeople**
  - **cp BagB/You.txt CoolPeople**
- **Exercise: Look into the move (mv) instruction to do this in a more optimal way!**

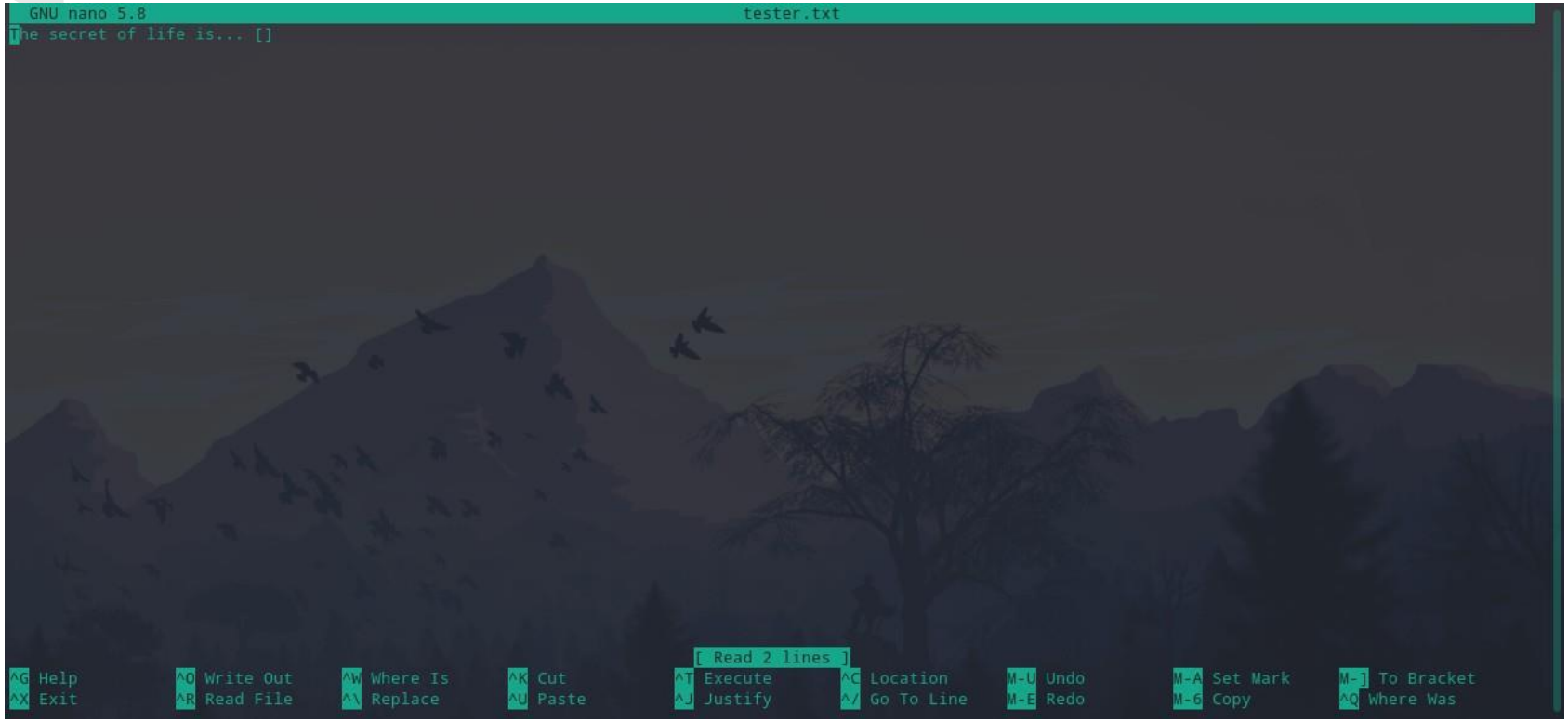
Part 2) Files Illustration

- Let's assume that we start at the following directory:
  - **/home/shogom**
- Assume that we have a file in the directory, “The_Meaning_of_Life.txt”
- How can we return the contents of this file to the terminal?
  - **less The_Meaning_of_Life.txt**

Part 3) Editing

1. Nano
    a. **Allows for <u>quick editing</u> in the terminal...**
        i. Does not automatically hard-wrap lines that become overlong.
        ii. Includes the line below the title bar in the editing area
        iii. Does linewise (smooth) scrolling
    b. Conclusion: Nano is useful for very simple editing. One can think of nano **as a good introduction** to the far more powerful editor, Vim!
2. Vim
    a. **Allows for <u>sophisticated editing</u> in the terminal**
    b. A little bit unintuitive...
        i. **Easy to use, requires practice to master...**
        ii. **Syntax Highlighting!!!!!**

Nano


GNU nano 5.8
tester.txt
The secret of life is... []
[Read 2 lines]
^G Help
^X Exit
^O Write Out
^R Read File
^W Where Is
^\ Replace
^K Cut
^U Paste
^T Execute
^J Justify
^C Location
^/ Go To Line
M-U Undo
M-E Redo
M-A Set Mark
M-6 Copy
M-] To Bracket
^Q Where Was

Vim

Part 4) File Conversions

- convert
  - **“Convert between image formats as well as resize an image, blur, crop, despeckle, dither, draw on, flip, join, re-sample, and much more.”**
  - **Exercise:** What are the parameters? Try this for yourself!
- ffmpeg
  - **“A very fast video and audio converter [...] It can also convert between arbitrary sample rates and resize video on the fly with a high quality polyphase filter.”**
  - **Exercise:** What are the parameters? Try this for yourself!

Part 5) Github

- Git
  - **“The stupid content tracker.”**
  - Provides a very **fast and optimal terminal-interface** for interacting with Github.
    - Create Repositories
    - Push Changes
    - Create Pull Requests
    - Download & Build GitHub Hosted Projects
  - **Learning a version control will be integral in your future career! (Start now!)**

Part 6) Compiling Code

- java
  - **“Launches a Java Application”**
    - javac file.java
    - java file
  - Classes ⇒ OOP, JDBC
- gcc
  - **“GNU project C and C++ Compiler”**
    - “gcc file.c -o output_contents”
    - ./output_contents
  - Classes ⇒ Programming (C), Robotics, Computer Organization, Intro to Operating Systems

Part 6) Compiling on the EECS Server

Motivation

- Grading is based on **whether your code compiles** and runs **on the EECS server**.
  - Your local machine may be running a different version of C (ANSI C 98')
  - Your local machine may be running a different version of JRE (JRE 8)
- You must always make sure that your code runs as expected on the EECS server to ensure that no missmarking occurs as a result of deprecated / unexpected programming behaviours on your end.
- **Knowing how to compile code with the terminal gives you more power. Using flags to enforce certain behaviours teaches you how to troubleshoot effectively!**

Adding Yet Another Layer of Complexity

- We now have a basic understanding of Linux commands...
- **We can actually execute certain commands on directories/files with particular naming conditions! (Powerful and Easy Filtering Techniques)**
    - Wild Card = *
    - Single Character = %
    - Set of Characters = []

Adding Yet Another Layer of Complexity Example

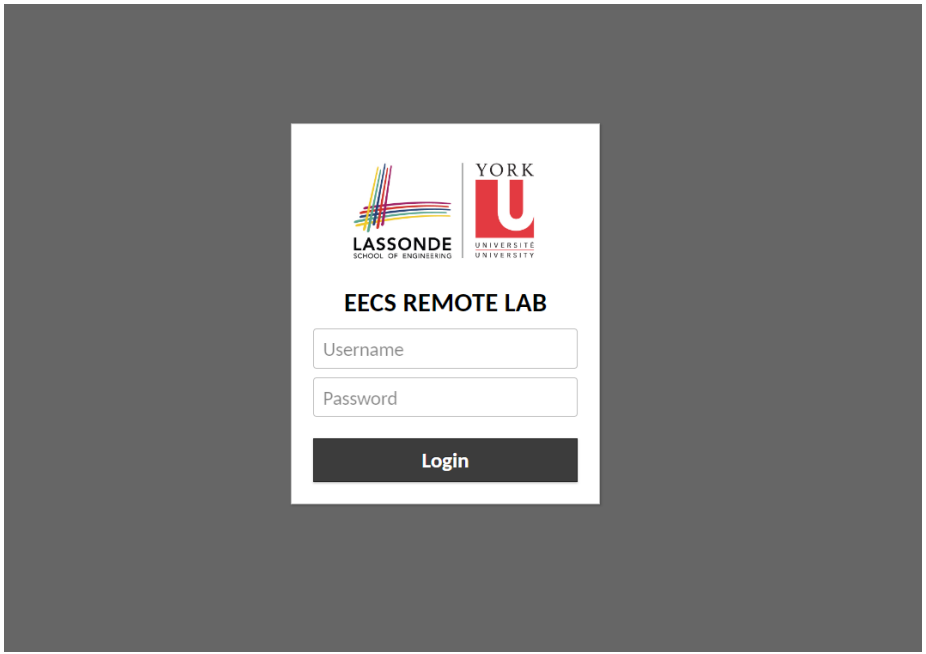
- Assume that we're in the following directory: **/home/shogom**
  - Assume that there are 10,000 files here, with many uniquely generated identifiers.
- We want to delete all text files that begin and end with, "B"

- Option A) Use the Instruction **rm *.txt**
- Option B) Use the Instruction **rm B*B.txt**
- Option C) Use the Instruction **rm [BC]*[BC].txt**
- Option D) **rm -getPattern "B(..)B" -fileType ".txt" -user_perm = "T"**

How Do We Connect to the EECS Server!?

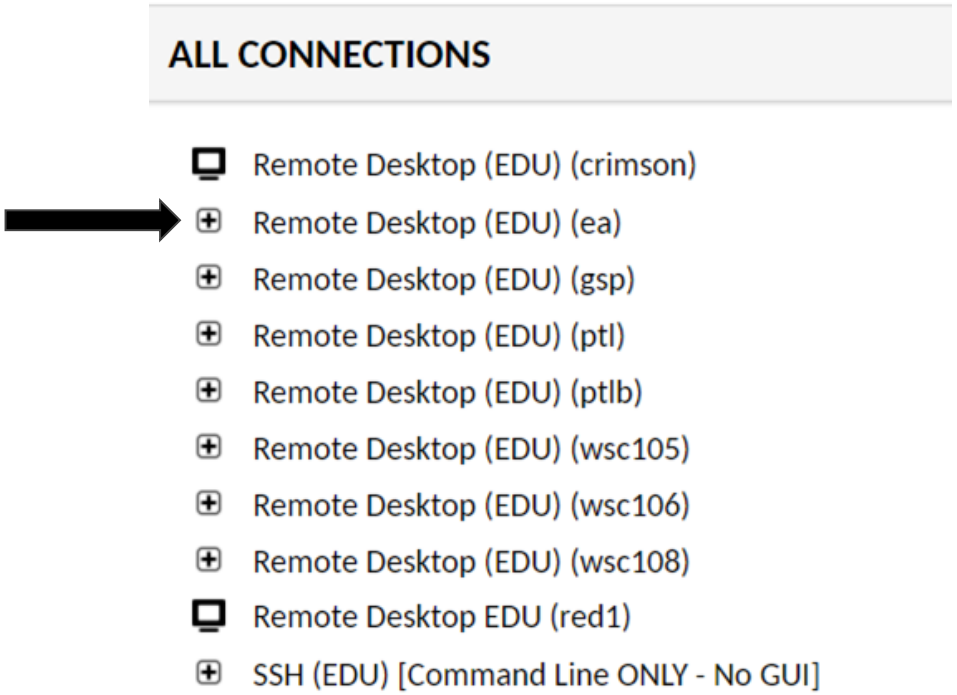
- Open a terminal of your choice, and use ssh:
  - **ssh red.eecs.yorku.ca -l EECS USERNAME**

Using the EECS Remote Labs

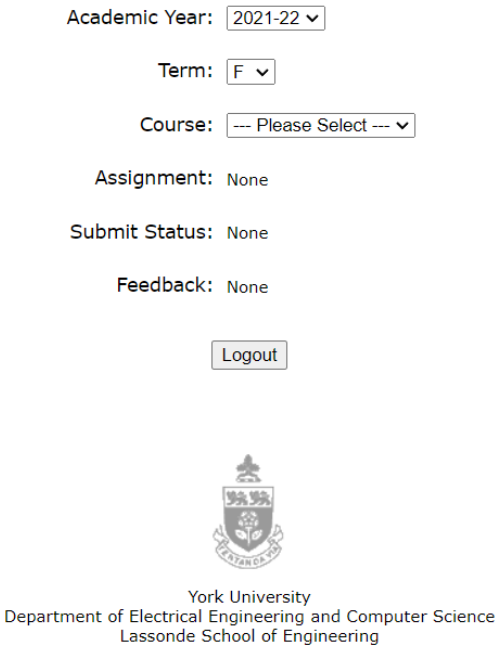


- https://remotelab.eecs.yorku.ca/

Using the EECS Remote Labs



Web Submit



- https://webapp.eecs.yorku.ca/submit/

But wait… What about using Linux Commands via the Terminal!?

```
submit(1)                          General Commands Manual                          submit(1)



NAME
       submit - submit files for a report for a course

SYNOPSIS
       submit [-l] course assignment file [file ...]
       submit -l course assignment

DESCRIPTION
       submit  allows  students  to  submit  files as part of a report for an assignment.  The name of the course and
       assignment is up to the professor and will be provided by them.

       The -l option will give a detailed list of all files that you (the student) have submitted for that assignment
       both through this call to the submit command and through previous calls to the submit command, after the files
       have been properly submitted.  With the "-l" option, you can also choose not to specify any files to  be  sub-
       mitted.  In this case, the program will simply list out all files that you have previously submitted.

       At  least one file must be submitted, but any number of files may be submitted.  If the same file is submitted
       twice, then the latest file will overwrite the old one.  Only the name of the file, not the directory  of  the
       file will be used when determining if you are overwriting an old file.  For example if you submit
              dir1/a1.c
       and then submit
              dir2/a1.c
       Then the second submission will overwrite the first.  In other words, there will only be one copy of a1.c sub-
       mitted.  However, if the names are different:
              dir1/a1.c
 Manual page submit(1) line 1 (press h for help or q to quit)
```

Example Submit Command for EECS Courses

- submit 3421 assignment1 *.sql
  - Typically the command will be given in your assignment instructions...
  - It's important that you still know the specifics about the command. Read the manual specifications!

Concluding Thoughts

GitHub Project for You

(Your First Side Project…!?)

1. Host a personal website on your EECS account:
    a. Accessed By: https://www.eecs.yorku.ca/~**EECS_USERNAME**

    (Hint: You must use the www directory, and learn about file permissions!)

1. Learn the required languages / frameworks
    a. **HTML5**
    b. **CSS**
    c. **Javascript (Node, Angular, Vue, etc)**
    d. **PHP**
    e. **Ajax**
2. Push the code onto a GitHub repository
3. Send me your website after it's ready to deploy worldwide! 😊

Thanks for Coming 😂