COMP 2710 Introduction to Operating Systems
Project 1 – Using the Linux Terminal

Points Possible: 100 due: 4:59pm Jan 25th, 2019

There should be no collaboration among students. A student shouldn't share any project code with any other student. Collaborations among students in any form will be treated as a serious violation of the University's academic integrity code.

Objectives:

- Get to know the Linux operating system
- Learn basic Linux commands
- Prepare a Linux programming environment for the future projects
- Learn how to remotely access Linux servers (see Option 1 below).
- Learn how to install VirtualBox and unix-core operating system (see Option 2 below).
- Compile and debug your first C++ program in Linux

Requirements:

- Each student should **independently** accomplish this project assignment. You may discuss with other student to solve the coding problems.
- To embark on this project, you may choose one of the following four options.
  - Important! Option 1: For Mac and Linux users, use SSH to connect to a remote Linux server. Please read files in "tutorial" on Canvas for details.
  - Important! Option 2: For Window10 users, Please read "*enViro_tutorial*", "Putty Tutorial" and "Win subsys Linux" for details.
- You are highly recommended to use a Linux operating system.

1. Setup Linux Programming Environment (10 Points)

Important! When you access a remote Linux machine in the Linux lab, please *don't use the "sudo" command*, because you have no administrative rights. You may use the "sudo" command on your virtual machine or local machine where you are a system admin.

You should choose one of the following four options to setup your Linux programming environment.

1.1 Option 1: Remotely Connect to Linux Machines

Please follow the instructions specified in a "Tutorial" folder on Canvas to learn how to access a remote Linux server. You must keep in mind that when you access a remote Linux machine in the Linux lab, please *don't use the "sudo" command*.

1.2 Option 2: Install VirtualBox/Windows subsystem for Linux/Putty
Please follow the instructions in a "Tutorial" folder on Canvas to learn a way of installing VirtualBox and CentOS in your own laptop or desktop computers.

2. The `script` Command

The script command line tool allows you to save a session of your terminal. In addition to saving each command per line in a text file, the script command makes a typescript of everything that happens on your Linux terminal. Screen casting tools to a desktop session(GUI) is what script is to a terminal. Let us demonstrate the usage of `script` through the following example:

```
$ script
$Script started, file is typescript
$ cd
$ ls
file1 file2 file3
$ exit
exit
Script done, file is typescript
```

Then, you may use the `mv` command to change the file name from `typescript` to any name you like. Alternatively, you may specify the name of your log file upfront as below:

```
$ script sample.script
$Script started, file is typescript
$ cd
$ ls
file1 file2 file3
$ exit
exit
Script done, file is sample.script
```

3. Tasks (90 Points)

Script the following session using the `script` command. You may save each session (i.e., each task below) in one script file. Using the *`tar`* command to submit a tarred and compressed file named `project1.tgz` (see Section 4.2 for details).

3.1 (Task 1: 33 points.) Please use the `script` command to create a file named "`commands.script`" demonstrating that you understand how to use the following basic Linux commands.

- `pwd` – displays the pathname for the current directory.
- `ls` – lists the files in the current directory.
- `mkdir` – makes a new directory.

- `cp` – copies a file from one location to another.
- `mv` – moves a file from one location to another.
- `rm` – removes a file.
- `chmod [options] mode filename` – changes a file's permissions.
- `clear` – clears a command line screen/window for a fresh start.
- `who [options]` – displays who is logged on.
- `nproc` – displays the number of cores.
- `g++ --h` – displays the help information for the g++ compiler.

Reference: How to Start Using the Linux Terminal
https://www.howtogeek.com/140679/beginner-geek-how-to-start-using-the-linux-terminal/

3.2 (Task 2: 12 points.) When you log in a Linux system, you should get to know the system in more detail. You are asked to find out your computer system's attributes, including **CPU frequency, cache size, memory size, the list of PCI devices, hard drive, network MAC address and link speed,** and **the devices generating interrupts**. The following system commands can help you. Many of these commands are for system administration, so you should run them as root if needed.

```
$more /proc/cpuinfo
$more /proc/meminfo
$more /sbin/lspci
$more /proc/interrupts
```

Tip: You may pipe the output of any Linux command to a file on Linux using

```
Your command > test.txt
```

The above command writes all outputs from each commande to a file called "test.txt" instead of displaying the files on a monitor.

Please store the output from the above four commands into the following four files using the pipe (see the above tip).

- `cpuinfo.txt`
- `meminfo.txt`
- `lspci.txt`
- `interrupts.txt`

3.3 (Task 3: 35 points.)With the computer up and running, you should give it a try to see if you can use the utilities on the system. For a system programmer, these include at least the editor, the compiler, the libraries, and the debugger. You are asked to do the following

3.3.1. Using your favorite editor, code a program (simple.cpp) that processes an array of 10 numbers, calculates: (1) a factorial value based on how many positive numbers users want to input, (2) the value of standard deviation, and prints it out. I recommend either **vi/vim** as an editor for our course. Both "factorial.cpp" and "standard deviation.cpp" are uploaded on Canvas.

3.3.2. The GNU compile is the default open source compiler on Linux. You should check a little on what gcc you have, and then compile your program as follows.

```
$g++ -v
$g++ simple.c [-o your_favorite_obj_file_name]
```

4. Deliverables

4.1 Multiple Script Files. (10 points. A tarred file 5 points, filename 5 poitns)

You need to submit one tarred file with a format: Firstname_Lastname.tar.gz including 6 files: (1) `commands.script` (2)`cpuinfo.txt` (3)`meminfo.txt` (4)`lspci.txt` (5)`interrupts.txt` (6)`simple.cpp`

Since you have generated multiple script files, please save all the script files in one directory (i.e. project1). Then, you should achieve all the script files into a single tarred and compressed file with a `tar` command.

Syntax: `tar -zcvf tar-archive-name.tar.gz source-folder-name`

Assume that the script files and your report are located in */home/cse_h1/xzl0031/comp2710/project1*, then you can follow the instructions below to prepare a single compressed file.

```
tar -zcvf Xuechao_li.tar.gz /home/cse_h1/xzl0031/comp271/project1
```

5. Grading Criteria

1) Setting up your Linux programming environment: 10% (see Section 1)
2) Using Linux commands: 33% (see Section 3.1)
3) Getting to know your system: 12% (see Section 3.2)
4) Using `g++`: 35% (see Section 3.3)
    - Compiling: 10%
    - Execution: 10%
    - Output: 15%
5) Using `tar`: 10%

6. Rebuttal period

- You will be given a period of **three business days** to read and respond to the comments and grades of your homework or project assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.