Unix Tutorial

> Those who do not understand Unix are condemned to reinvent it, poorly.
>
> Henry Spencer, University of Toronto

> Unix: Some say the learning curve is steep, but you only have to climb it once.
>
> Karl Lehenbauer

1 Introduction

Modern versions of Linux allow an experienced computer user to do normal computing tasks immediately without any additional training, but to begin to harness some of the power of Linux a little work is required. In particular, much of the power of Linux and other Unix variants can be found in their command line interfaces (CLIs). Many current users of Microsoft and Apple operating systems have little experience with CLIs and are only used to graphical user interfaces (GUIs). The purpose of this exercise is to you expose to the Linux CLIs in order to give you the experience necessary to be more productive using Linux and other Unix operating systems in the future. Note that if you find the Unix tools that you will learn about here useful, there are options for available for installing them on your Microsoft Windows computers. MacIntosh OS X users will find that most of the tools mentioned here are already installed on their computers (though they may not be well advertised), and the once that are not can be downloaded fairly easily.

2 VMWare

Currently, there are few student accessible Linux terminals on campus. You can access campus using VMWare Horizon View. VMWare is software that allows you to remotely log into both Linux and Windows software. You can also install VMWare on your own computer.

Instructions for using VMWare are available at `https://sharepoint.csbsju.edu/itservices/kb/Pages/vmware_vmware.aspx`.

3 Account Setup

You will probably find many occasions during this lab when you will want to transfer files back and forth between your Windows and Unix accounts. While there are many ways to do this (including email them to yourself, put them a USB drive, or on a CD), the easiest ways involve transferring the files directly between the accounts involved. To get this to work takes some setup. In this section you will see how you can get direct access to your Windows files from Linux, and vice-versa.

Some documentation about the CSB/SJU Linux system is at: `https://csbsju.teamdynamix.com/TDClient/2466/Portal/KB/ArticleDet?ID=124068` Also note that you can run a single Linux terminal window using ssh (`https://csbsju.teamdynamix.com/TDClient/2466/Portal/KB/ArticleDet.aspx?ID=123999`) or a remote Linux desktop using VMWare (`https://csbsju.teamdynamix.com/TDClient/2466/Portal/KB/ArticleDet?ID=124153`) on your own computer.

3.1 Windows File Access from Linux

Your CSB/SJU OneDrive should be available from any browser on Linux. Log in and access your files from: `https://www.csbsju.edu/o365`, or whatever way you like to get to OneDrive.

4 VMWare Horizon View

While logged into your own computer, or one of the campus Windows computers, run through the VMWare Horizon Client Setup instructions - `https://csbsju.teamdynamix.com/TDClient/2466/Portal/KB/ArticleDet?ID=124153&SIDs=9090` While setting it up, instead of picking "Windows Academic", choose "Linux Student Pool."

4.1 Linux File Access from Windows

1. On Windows open the "Computer" of File Explorer, and click "Map Network Drive".
2. On the Map Network Drive Window, pick any unused letter for the Drive: choice (X:, Y:, Z:, etc.) . For the folder, put:

   \\\\`elm.ad.csbsju.edu`\\`YOUR_UNIX_USERNAME`

   where you should replace YOUR_UNIX_USERNAME with your Unix username (e.g. astuden001).

   should also be accessible from other Windows programs.

5 Playing with the GUI

Before we get to learning about the Linux CLI, I would like you to play around a bit with the Linux GUI. Explore the menus. Try out a few programs. Change some of the settings (maybe the wallpaper?). Then pick a game (alas, there used to be more choices) that you have never played before and try it out for a few minutes. Record down the name of the game, how it works, and what score you got (if the game has a score) in your lab notebook. Or since their are so few games, try another program that you have never used before, play with it, and write notes on what you have tried. Some good ones to try include gimp (Gnu Image Manipulation Program - under "Graphics" on the menu), pencil (also under "Graphics"), or Aladin (under the "CSBSJU—Physics" menu.

6 Tutorial

The Unix tutorial at `http://linuxclass.heinz.cmu.edu/doc/Unix-Tutorial-surrey/`, which will be the focus of this exercise. Start at the first section and work your way through the entire tutorial.Make sure that you do all of the exercises listed.

The tutorial is setup for users at another college, but all of the commands in it should work here as well, though the file paths are different. Another thing to note about the tutorial is that it uses the command shell called **csh**. Here at CSB/SJU we use another shell, **tcsh** which is based on **csh**, but has some more advanced features. This should not be a problem because **tcsh** is more or less a superset of **csh**, so you should be able to do everything in the tutorial. Note, though that there are other shells that are not as compatible with **tcsh**. In particular, the most commonly used shell on Linux is **bash** and **bash** has many syntax differences from **tcsh**. If you would like to try a different shell, you can type its name at the command prompt. You can also change your default shell, but I wouldn't recommend that at this point.

To run through this tutorial you should log into one of the department's Linux computers. Start up a web browser — Firefox, Google Chrome, Konqueror, or any other browser should work fine — and go to the page mentioned above. To run the examples described in the tutorial you will also need to have a terminal window (also known as a command shell) open. There are several types that will work: **xterm**, **gnome-terminal**, and **rxvt** to name a few. You should be able to find them under one of the menus in the GUI.

For this exercise, you need to keep a record of the commands that you typed. I recommend that you use the **script** command to do this. After you open a terminal type: `script tutorial.txt` — where "tutorial.txt" is the file name that your commands will be saved in. When you are done with you session, type "exit" and it will be saved. If you have to stop this exercise and start it up again you will want to use a different file name the second time (e.g. `script tutorial2.txt`, otherwise you will lose your work from the first file.

Note that for this exercise each of you should work on your own. You are free to ask for help from other students, but do the entire tutorial yourself.

Before you get started let me add a couple more time saving hints. In **tcsh** (and **bash**), you can scroll through previous commands that you have typed with the up and down arrow keys. This is a big time saver if you make a typo. "Tab completion" is a related feature. When you are typing the name of a command (or file), if you hit the "Tab" key, bash will attempt to complete the command (or file) name for you.

Also, Unix guis typically have several methods of copy and pasting. The Unix style way of copy and pasting is to select text using the left mouse button, then move the pointer to the place you want to paste to, and then click the middle mouse button. Try it one or twice. Copying and pasting this way would make it quite easy to do this entire assignment without doing any typing, but I suggest you type most of the commands given, since that will make it more likely that you will remember them. Many programs also support the Microsoft Windows style copy and pasting from their "Edit" menus (also available using Control-C, -V, and -X). Another trick with the middle mouse button involves web browsers. If you middle click on a link in most Unix browsers, the linked page will open in a new tab or window. Also, if you select the text of an URL with the left mouse button, and you paste to a blank spot on web page, the URL will be loaded on the browser. (You can also configure Firefox to act this way on Windows and Mac computers.)

Also, if you get stuck while running a program from the terminal try hitting Control-C, which should exit out of the program. If that does not work, you can also try Control-Z, which suspend the program.

Another useful command is **finger**. Try: "`finger YOUR_USERNAME`" and "`finger jcrumley`".

7 Post-Test

After completing the tutorial, do the following tasks:

- Make a directory called "370" in your home directory, and put the text file(s) that you created recording your work on the tutorial in that directory. As part of your grade for this tutorial I will look at those text files, so in order to enable me to do that, you will have to change the permissions so that I can look at them. Note that you will need to add "read" and "execute" permissions for "group" for your home directory and the "370" directory. You will also have to add "read" permissions for the files that hold the recording of your session. You may want to use the "X" option to chmod (as opposed to "x"). I suggest that you keep all of your lab files in this directory, or even better in subdirectories for each lab.

- List *full* details about *all* of the files (including "hidden" files) in a directory using **ls**, listing the files from the newest file to the oldest. You will need to look at the documentation (**man** page) to figure this out. (Note: you should only have to use **ls**, **sort** will not be needed). Use this command to list the files in the directory "`/usr/people/plasma/group_docs`". Save the command that you used and your results to a file, print the file using "**enscript** `-r FILENAME`", and tape the results in your lab notebook.

- Type "**more** `~jcrumley/public_html/370/students/$USER`" to receive a special message. Write this message in your lab notebook. How does this command work? Listing the contents of that directory with "`ls -l`" may help you figure it out. '

8 Turning in your work

Hand in your lab notebook when you are done with this exercise.