CS 250 – Linux Lab

Introduction to the Linux Command Line and Emacs

We are going to be using the Computer Science department’s Linux server “ice” to develop our programs this semester. The purpose of this lab is to let you get familiar with using the **command line interface** (also known as the “terminal” or the “shell”) to write, compile, and execute programs. The command line is the traditional way for experts to interact with Linux systems, so it is important for you to develop skills in using the shell.

The easiest way to connect to ice (which we will refer to as the “remote” machine) from your desktop or laptop computer (which we’ll call the “local” machine) is by using a piece of software called X2Go. X2Go allows you to interact with a remote computer system by displaying the remote computer system’s desktop in a window on your local computer. X2Go is available for Windows and Mac computers. If you have a Linux computer, you don’t need X2Go to connect to another Linux computer. Ask me during office hours if you wish to connect to ice from a Linux system.

1. Open a web browser, and go to the website wiki.x2go.org. Find the section labeled “Get X2Go”. You want the X2Go **Client**, not the client/server or server software.Download the installer that is appropriate for your computer: either the Windows Installer (7 and Later) or macOS 10.13 and higher DMG. Install the software as you normally would for your system.
2. Once X2Go is installed on your machine, start it. The first thing we will do is create an X2Go session. A **session** is a collection of settings that allow you to connect from one the local machine to the remote machine in a specific way.
3. Once the X2Go client starts, perform the following steps to create a new session that will allow you to quickly connect to ice. You only have to create a session once for each computer that you want to connect to. After that, you can use the previously created session anytime you wish to connect.
	1. From the X2Go menu bar, select Session > New Session
	2. Under the Session tab, set the following text fields. Case is important.
		1. Session name: Ice
		2. Host: ice.truman.edu
		3. Login: <your TSU login name>
		4. Session type: XFCE
	3. Click the OK button.

The new session you just created appears on the left of the X2Go window.

1. Click on the session in the X2Go window to start the Linux login process.
2. Your username should already be filled. Type your password in the textbox and click OK. If you get a dialog box telling you that the server is unknown, and asking if you want to trust it, click "Yes". In a few moments, the Linux desktop should appear in a new window. Be patient, it can take a few seconds.
3. Notice that you now have two X2Go windows open – one is the X2Go session manager that shows all of the sessions that you have created, and the other is the Linux desktop. Minimize, but don’t close, the X2Go session manager window – we won't need it for a while. The other window contains the remote computer’s Linux desktop. Make the window that contains the Linux desktop larger by dragging the lower right corner of the window to enlarge it.
4. The Linux system may ask you some initial configuration questions. It will only ask these questions once. If you are asked to choose a password for a "new keyring", click "cancel". If you are asked to “configure panels”, choose the default.
5. You can interact with the Linux GUI from the Applications menu at the top left of the screen. You should feel free to look around and explore the Linux GUI, but do this later. In this class, we will not be using the GUI to interact with Linux, except in a few specific circumstances.

Instead, we will be learning to interact with Linux using the shell. The shell is an interactive program that reads commands from the user and then executes them on behalf of the user. There are many shells available in Linux. The one we will be using is called “bash”. Bash is the latest version of the original Unix shell, originally written by Stephen Bourne[[1]](#footnote-1). Bash is an acronym for “Bourne Again SHell”.

1. Access the command line by finding the icon representing the terminal at the bottom of the screen. Click on the icon to start a terminal session. When the terminal starts, it should look something like this:



1. When you are finished with your Linux session, **do not click the close box on the X2Go window. It is very important that you properly end your session either by logging off of the Linux system when you are done**.

End your session by logging out of the Linux system. To log off of Linux, go to the Applications menu in Linux, and choose Log Out. When the "Log Out" menu appears, click the appropriate button. After a few seconds, the session manager should appear again. Quit X2Go from the File menu. You have now properly closed your X2Go session.

If you do not log off of the Linux system when you are done, you will lose any unsaved work, and you may corrupt your Linux settings, which will make it impossible for you to login again without having the systems administrator reset your account. Simply closing the X2Go window is equivalent to turning off the power off on your desktop system instead of shutting it down; ***Do not do this.***

***If you access Linux using X2Go from a laptop, it is very important that you logout of Linux or suspend X2Go before closing your laptop.*** Closing the lid on most laptops puts the laptop to sleep. Before it sleeps, the laptop closes all of its network connections. This will also destroy your remote Linux session.

1. We will now begin to learn the Linux command line. Open X2Go again, login to ice, and open a terminal if one does not appear automatically.
2. Today we will be learning some of the basic commands that are required to successfully work in the Linux command line environment. As we learn new commands, you will be asked to answer some questions about the commands. Write your answers on the last page.
3. As in most other operating systems, the Linux file system is organized into a tree structure. The very top of the tree is called the root. The root is denoted by the symbol '/'. Folders in Linux are called "directories" (dirs, for short).

We can name a specific directory or file anywhere in the computer by listing the set of folders encountered as we traverse the file system from the root to the directory or file. This listing is called the directory or file's "pathname". In a pathname, each folder level in the file system tree is separated from the one above it by the '/' symbol.

Each user on a Linux system has a directory that belongs just to them, called their **home directory**. When you first log in, your home directory is also your **working directory**. You can change your working directory at any time, but your home directory is always yours and it never changes. Whenever you give a command that refers to a file, Linux will always look first in your working directory for the file.

You can find your working directory using the "pwd" (print working directory) command. Type the command pwd in the terminal window, and then hit the <enter> key. What is the pathname of your working directory?

Write your answer here \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. A subdirectory is a directory contained inside of another directory. The "ls" command is used to display (LiSt) the files and subdirectories in the working directory. Ls will show regular files in white, and subdirectories in purple. Execute the ls command. What subdirectories exist in your home directory?

1. You can make your own subdirectories in the working directory using the "mkdir" command. Type "mkdir CS250". (Don't put a space between CS and 250. In general, you should avoid creating dirs and files with spaces or other non-alphanumeric characters in the name.) Use the ls command to verify that you created a subdirectory named "CS250".
2. The "cd" command is used to change your working directory. Type "cd CS250" to change your working directory to CS250. Use the pwd command to verify that CS250 is your current working directory.
3. When navigating through the file system, the shell provides several shortcuts to help you move around. The shortcut does not include the quotation marks.
* The shortcut ".." (two periods) always refers to the directory *above* the working directory.
* The shortcut "." (one period) always refers to the current working directory.
* The shortcut "~" (tilde) always refers to your home directory.

Type "cd ..", and then verify that you are back in your home directory.

1. Change your working directory back to CS250.
2. To start an application from the shell, you type the application's name and press the enter key. Start the web browser "firefox". Once Firefox opens, make the terminal window the foreground window by clicking on it.
3. When you run an application, the shell waits for the application to finish before it allows you to run another command. Quit firefox from its file menu. Notice that only after the application has ended does the shell print another prompt, indicating that it is ready for you to type the next command.

Often, we want an application to start and "run in the background", allowing us to continue to interact with the shell while the application runs. You can tell the shell to launch an application and then run the application in the background by following the application name with an ampersand when you start it.

Type "firefox &" in the shell. Firefox will start up, and the shell will then immediately print a prompt indicating that you can type the next command. Quit firefox.

1. The editor that we will be using this semester is called "emacs". We are using emacs because it seamlessly integrates many of the Linux developer tools like the compiler (clang), the build manager (make), and the debugger (gdb) that we will be using this semester.

There are other editors installed on most Linux systems, and you should feel free to experiment with these outside of class. However, you must develop some skill with emacs, because it is the de-facto programmer's editor used to develop software on Linux.

1. Start emacs in the background by typing "emacs &". You should see a window that looks like this:



1. A few emacs commands are available from its menu system, but most advanced commands in emacs are issued by typing special key combinations, using either a <esc> key, or a <ctrl> key sequence. Emacs refers to the escape key as the meta key.
2. Click on the blue text labeled "Emacs Tutorial" and work through the tutorial. If you aren't finished by the time that class ends, make a note of your stopping place, so you can continue on your own before the next class meeting. To exit emacs, choose quit from the File menu, or use the key sequence <ctrl>-x <ctrl>-c.
3. If you haven't finished the emacs tutorial, make sure that you do so by the beginning of the next class period.
1. Stephen Bourne is a British computer scientist who worked at Bell Labs at the same time as Dennis Ritchie. He is also the past president of the ACM. [↑](#footnote-ref-1)