CSE 391, Spring 2019 Assignment 1: Basic Unix Shell Commands Due Tuesday, April 9, 11:59 PM

This assignment focuses on using the bash shell to execute common Unix commands. Some of the questions are Unix commands you must figure out, and others are general questions about the particular Linux system you are using. Note: Unless otherwise specified, the answers to each question in this assignment can be found entirely using commands shown in the lecture slides from the first week. You may use other commands if you like, but you should constrain yourself to those from lecture or from the *Linux Pocket Guide* textbook. Ask the instructor if you are unsure whether a particular command is allowed.

For Task 1 and Task 2 there is nothing to submit. For Task 3 and Task 4, you will submit your responses to the Google Form linked on the course website and calendar.

Task 1: Log in to a Linux environment

First, **log in to a Linux environment.** You have several options to choose from here. Everyone should be able to install the CSE Virtual Machine (VM) Image. Other options include installing Linux on your machine, using one of the Linux computers in the CSE basement labs or using the CSE department's attu Linux server using the ssh program. We refer you to the <u>Working At Home</u> link on our course home page for information on options available. If you aren't able to successfully log in to a Linux environment, please contact us for help or ask a classmate.

If logged on to a Linux desktop environment (the CSE VM, a Linux machine in the basement), launch a **Terminal** window and a **text editor** from the Linux user interface, generally from the top left drop-down applications menu. You can usually find the terminal program under System Tools or Accessories. If logged on to attu, you can access editors like emacs or vim by typing them at the command line. You can also edit your text file on your own machine.

Task 2: Prepare a directory

We have set up a ZIP archive full of support files that you must download to your Linux environment. Do the following:

- Create a directory inside your home directory named 391 .
- Create a sub-directory inside 391 named hw1.
- Download our file hw1.zip and save it into your new hw1 directory. You can do this in one of two ways:
 - 1. By opening a web browser on your Linux environment, browsing to our course web site, clicking the Homework link, finding the link to hwl.zip, right-clicking it, choosing Save Link Target As..., and browsing to the hwl folder;
 - 2. Or, by typing the following command into your terminal window, when the current directory is hw1:
 - wget http://courses.cs.washington.edu/courses/cse391/19wi/homework/1/hw1.zip
- Unzip the hw1.zip file's contents into your hw1 folder. You can do this in one of two ways:
 - 1. By running a file browser/manager (in the CSE virtual machine you should have an icon that looks like a file cabinet that will open up a file manager) and browsing to the hw1 folder, then right or double-clicking on the hw1.zip file, and using the graphical unzipping program to extract the files;
 - 2. Or, by typing the following command into your terminal window, when the current directory is hw1:
 - unzip hw1.zip
 - (We are trying to persuade you that doing things in a terminal can sometimes be the easier way!)

If you did everything correctly, you should now have several files and directories within your hwl directory, such as java/, website/, animals.txt, Burrot.java, numbers.txt, and songl.txt.

(Continued on next page)

Task 3: Learn more about the Linux environment

The following are questions about your Linux system for you to investigate and discover the answers. Most of the answers to the questions below can be found entirely using commands that you were shown in the week 1 slides. Any question that requires additional commands or features will have a label of "Self-Discovery". Remember that you can learn more about any command by typing:

man *command*

Each Linux system is different, so there is not one "correct" answer to every question. You will receive credit if your answer is plausible for a Linux system in general.

- 1. Try running each of the following commands on your Linux system, one at a time. (You don't need to turn in the output of the commands.) What is the each command doing? Describe what it does in one sentence. Remember that you can read more about a command and its parameters by looking at its man page.
 - ls
 ls -1 (in this case, the -1 is the number 1, not a lowercase L)
 ls -m
- 2. What is the full path of your home directory on this Linux environment? Run an appropriate command to find out.

(Note: You can select text on the Terminal window and copy/paste it into your text editor. The terminal program usually has a top menu bar where you can choose Edit \rightarrow Copy and Edit \rightarrow Paste. Right clicking may work if you are NOT on attu.)

- 3. What is the exact current date and time, as reported by this Linux environment? Run a command to find out.
- 4. What other users have home directories on this Linux environment? (Hint: Find out what other user directories exist within the same parent folder as your home directory.) Provide a list of all such users in the format produced by the ls command. *If there are more than 5 users, listing the last 5 is sufficient.*
- 5. (Self-Discovery) What release of Linux does this computer use?

To find out, run the Linux **uname** command, which outputs various information about the current system. It has various parameters that you can learn in its man page.

You are looking for the release of the <u>kernel</u> (the core) of the Linux OS, not the name of the "distribution" such as Fedora or Ubuntu. (*Hint: The answer begins with something like 4.14.90*)

(Continued on next page)

Task 4: Linux Bash shell commands

For each of the numbered items below, determine a single bash shell statement that will perform the operation(s) requested. Each of your solutions must be a single one-line shell statement and should not use Linux's multi-statement joining operators such as |, &&, ||, ad ;. (We will learn about these next week.)

Note that you can return to the Google form and modify and re-submit your answers multiple times before the deadline. In response to each question, write the command that will perform the task described, not the output that the command produces.

To test your commands, you should have unzipped hw1.zip into the current directory. You can assume you are in the "hw1" directory when doing these problems. Most of the questions below entirely use commands shown in the week 1 lecture and/or slides. Several questions require you to learn new parameters to those commands; find these out by looking at man pages or the *Linux Pocket Guide*.

- 1. Copy the file MyProgram.java from the current directory to the java subdirectory.
- 2. List all files in the /var directory, in reverse alphabetical order. (Notice the slash!)
- 3. List all files in the current directory, in "long listing format".
- 4. Print a text calendar for the month of July 2019.
- 5. Rename the file Burrot. java to Borat. java. (Hint: Renaming is done using the same command as moving.)
- 6. Delete the files diff.html and diff.css . Note that your answer must be a single command and not multiple commands. (*Hint: Many commands can accept more than one parameter.*)
- 7. (Self-Discovery) Set the file MyProgram.java to have a last-modified date of January 1, 2019, 4:15am. (Hint: the man page for the proper command describes the timestamp 'STAMP' format to use. Look for this!) (Also note that Linux is case-sensitive when you are specifying file or directory names.)
- 8. (Self-Discovery) You can use a * (asterisk) as a "wild-card" character to specify a group of files. For example, *foo means all files whose names end with foo, and foo* means all files whose names begin with foo. You can use a wildcard in the middle of a file name, such as foo*bar for all files that start with foo and end with bar.

List all web page files (files whose names end with the extension .html or .css) in the current directory. Note that the ls command can accept more than one parameter for what files you want it to list (e.g. ls website/ java/).

- 9. Copy all the text files (files whose names end with .txt) from the current folder to the java subdirectory.
- 10. (*Self-Discovery*) The cat command outputs the contents of a file to the terminal. The less command outputs the contents of a file to the terminal, page by page, pausing for you to press a key.

Use whichever command is best suited to display the contents of the file lyrics.txt.

- 11. Display the **contents** of all files whose names begin with song and end with the extension .txt, such as song1.txt and song2.txt. (*Write a single command that displays all their contents concatenated.*)
- 12. (*Self-Discovery*) The head and tail commands output only the first or last few lines (respectively) of a file to the terminal.

Display only the first 7 lines of the file animals.txt from the current directory on the terminal.

 13. (Self-Discovery) The wc command outputs how many bytes, words, lines, etc. a file occupies. Display <u>only</u> the number of <u>lines</u> in the file song3.txt (It is fine to display the filename as well as the number of lines, just don't display the number of bytes and words.)