Lab 3. LINUX commands.

Purpose and rationale

The purpose of this lab is to quickly get students up to speed with basic usage of the Linux development environment, as a preparation for all future lab activities.

Part 1. Prepare a script to show your work:

Log on to a Linux computer.

- At the end of the practice session, please be sure to exit script session with **exit** command.
- If you need to leave the script before you are finished,
  re-open the script and append to it by typing: **script –a StudentName_lab3.txt**
- You might run into errors while executing these examples. Look at the errors and see if they make sense. **Correct the issues if possible and rerun the commands, and then just keep going.**
- Follow the commands as listed below. **The occasional extra command (like ls , pwd, or cd, for example) is just fine.**
- If you incorrectly type a command, then backspace to fix it, it might look something like this:
  [bielr@sp1 lab3]> wc [K[K[Kgrep lab file1
  Do not try to edit your script file. Just leave it as is. I know what it is.
- You may find a few extra "q" for quit commands. That is OK. Just keep going.
- Practice the Shell commands using the **below list**.

Start execution of Part 1.

Type:	**script StudentName_lab3.txt**	To make a script of your terminal session.
	history -c	**C**lear the previous history so your script is not a mile long.
	cd csc60	Move to your directory for this class.
	mkdir lab3	Make a directory named lab3.
	cd lab3	Move to directory lab3.
	pwd	Print current working directory (*lab3*).
		We will be moving back and forth between *csc60*, *lab3*, and *aaa*.
	mkdir aaa	Make a new directory *aaa*
	cd aaa	Change current directory to *aaa*
	pwd	To check that you moved to directory *aaa*.
	cd ..	Change to upper directory which would be *lab3*
	pwd	Print current working directory. You should be back in lab3
	ls > file1	List directory content and redirect output to a file called "file1"
	cat file1	Display text content in file1

→ more commands on next page → → →

less file1	Like *cat* but will show a page at a time for a large file.
q	To quit the **less** command
file *	Get info on file types of all files

wc file1	Print newline, word, and byte counts for file1
wc *	Show Word Count in all files in directory
grep file file1	Find word *lab* in file1. Empty result

grep file file1	Non-empty result
cp file file2	Copy file1 to a new file
ls	**Check** that you have both files

cd aaa	Move one directory below Lab3.
pwd	To check that you moved to directory *aaa*.

cp ../file1 .	Copy file1 from directory above to *aaa* directory. *Note the* **space-dot** *at the end of the command.*
ls	Check that you got file1 copied here.
mv file1 file2	Rename file1 to file2

ls	Check to see that file 1 changed to file 2
mv ../file1 .	Move file1 from directory above (lab3) to here (aaa). *Note the* **space-dot** *at the end.*
cd ..	Move up to Lab3

ls	Check that you now have file2 here.
cd aaa	Move back down to directory aaa
ls	Check that aaa still contains both file1 and file2.

cmp file1 file2	Compare file1 with file2, show differences. Same file so no differences.
ls > aaalist	Create a different file
cmp file1 aaalist	Now compare two files known to be different

diff file1 aaalist	Like *cmp* except shows more info
rm -i file1	Remove file1. Answer the delete prompt with: **y** **Note:** if we typed **rm file1**, the file is deleted no extra notification.
ls	Verify its removal.

ps u	Show all user's running Process ID's
ps -l	Show processes (lower case L) (including Process ID Parent Process ID)
!!	Repeat previous command

→ more commands on next page → → →

Two choices here to get from *aaa* to *csc60*. Pick one choice:	
(1) cd ..	Move up a directory to *lab3*.
cd ..	Move up a directory to *csc60*.
or do one command instead of the two "cd .." commands.	
(2) cd ../..	Move up past *lab3* to *csc60*.
pwd	Print current working directory. You should be back in csc60

cd lab2	Move down to the lab2 directory *[**NOTE 2**: The commands **below** will require that you be on in the directory where **lab2.c** resides. If you path differs, you still need to move to the directory where your **lab2.c** file resides, and then try these commands.]*
head lab2.c	List first 10 lines of code
head -20 lab2.c	List first 20 lines of code
q	←(Might not need this, depends on shell) (Not needed in bash)
tail lab2.c *or*	List last 10 lines of code
tail -20 lab2.c	List last 20 lines of code
q	←(Type q to quite the application if needed) (Not needed in bash)
ls -al | less	Directory listing (too long) 'piped' to 'less' for viewing
q	Type q to quite the application if needed (Needed in bash)
history	History of commands entered
exit	Leave and save the script file.
exit	Exit your login if you are ready.

Deliverable: 45 points

Please upload to Canvas:

1. Lab 3 script file (**StudentName_lab3.txt**)

Deliverable:

Due for full points by March 7.
Due before it locks date is March 21.

Note to folks with their *own* Linux machines:

I expect you to do the above assignment. I expect to see the "history" command. If you feel it invades your privacy, then you have three choices:

(1) Log off and back in to start a fresh new session
(2) At the prompt,
type: **history -c** as directed which will clear the command history of your computer
(3) Do your work on the *Coding computers* like everyone else.

****
****

Part 2. Introductory Linux lab3 (also known as *Give-Linux-some-time)* It is strongly recommended that the following list of commands should be looked at over the course of time. It is not bound by a due date. You owe it to yourself and your future to have a concept of Linux.

Read/browse the **man** pages for the Shell commands listed below in Step #6

1. The Part 2 instruction to read/scan various commands is **not** to be included in your script file.
2. To view the manual for the command "script", type **man script**.
3. Use the space bar to scroll through the display from **man**.
4. Type **q** to quit each session.
5. **SEARCH**. Some of the commands below will show up as ~~e~~ or BASH_BUILTINS
    a. In this mode, the needed information is somewhere in a big display.
    b. Example: When doing a **man history**, you get more information that you expect.
        i. Type **/history** to **search** for the word "history" and see occurrences of that word.
        ii. Typing an **n** will take you to the next occurrence.
        iii. When you are finished searching, turn off the highlights by typing: **Esc-u**
6. Check out the "man" pages for the following two columns of commands.

script	cp
man (note the standard sections of the manual i.e 1, 2, 3 ...)	diff
who (also try w)	rm
gcc	history
touch	jobs
top	make
mkdir	ssh
ls	head
ps	tail
cd	logout
file	vi
cat	view (*This one is buried in the **vi** page…search for it.)*
wc	exit
grep	