**Linux File Management (75 minutes)**

1. Use ComptTIA Network+”Configuring Routing” Lab and LAMP virtual machine

* Username: lamp
* Password: Pa$$w0rd

1. Only hand in screen captures, questions with a blank are not required

**Exercise 1: Accessing the Command Line**

1. Start the LAMP virtual machine.
2. Logon with as **lamp**, password is **Pa$$w0rd**
3. You should see a command line prompt ending in a dollar sign ($).
4. At the prompt, type **whoami** and press Enter. You should see the regular user account name you used to log in to the system (though the name may be slightly different than what you saw when you logged in via the GUI), and then another command line prompt.
5. At the prompt, type **date** and press Enter. You should see the date and time (or at least what the computer thinks the date and time is currently), and then another command line prompt.
6. Type **Date** and press Enter. You should see a message that says “command not found” or something similar. This is because all command line commands are case sensitive. In other words, **date** does not equal **Date**.
7. Type **hostnamectl**. This should display the computer hostname and operating system.

**Exercise 2: Navigating through the Linux File Hierarchal System (FHS)**

1. Type **pwd** and press Enter. The pwd command shows your present working directory (in this case, it is your home directory). Any commands you enter in your current or present working directory affect the commands outcome. Record your home directory:
2. To see the files in your present working directory, type **ls** and press the Enter key. The ls command lists files in your "present working directory" only if you do not specify a directory.
3. To see how your present working directory affects the outcome of commands, change your present working directory to a new location by typing **cd /etc** and press Enter. The cd command allows you to change your present working directory. In this case, your present working directory is now the /etc directory (the location of several configuration files).
4. Type **pwd** and press Enter. Do you see /etc displayed? You should.
5. Type **ls** and press Enter. You should see different files than you did in step #2. You see different files because your present working directory is different than it was in step #2.
6. Linux color codes files and directories, the standard color coding is:

* White - “regular” files
* Blue - directory
* Green - executable file
* Sky Blue - linked file
* Pink - graphic image
* Red - archive/compressed file

1. Users familiar to Linux sometime turn off the color coding as they find it easier to see and work with files in the tty2 terminal.
2. To temporarily remove the color coding, type **ls –color=none**

**NOTE:** there are 2 hyphens before color**:** ls<space><hypen><hypen>color=none

1. To permanently turn off the coloring you could type *unalias ls*. This will “turn off” any color coding of directories by turning off the alias. (To redefine the alias and “turn on” color coding you would type *alias ls=’ls --color=auto’* (spaces between “alias ls” and “ls -“ , the two hyphens have no space between them))
2. Type **cd** and press Enter to go back to your home directory. The cd command with nothing entered after it is a shortcut to change your present working directory to your home directory.
3. Type **pwd** and make sure you are in your home directory. Your home directory should be the same directory you were in during step #1.
4. You can see the type of files in your home directory by using the –F option to the ls command. Type **ls -F** and press Enter. Any name followed by a forward slash (/). indicates the listing is a directory or, in this case, a subdirectory.
5. You can see a different file type by typing **ls -F /usr/bin/** and pressing Enter. An asterisk after a file means it is executable (can run as a program).
6. You can see another file type by typing **ls -F /etc/rc5.d/** and pressing Enter (several files will show). A @ after a file means it is a symbolic or “soft” link.
7. There are dot files (aka hidden files) in your home directory. Look at them by typing in **ls –a** and pressing Enter. A dot file, also called a "hidden file," starts with a dot (period) and is not displayed unless the -a option is used with the ls command.
8. Now just type in **ls** and press Enter. You should not see the dot (hidden) files in this listing. This happens because you need to use the –a option on the ls command in order to see them.
9. Now try a long listing of hidden files, by typing in **ls -a –l** and pressing Enter. The –l option is a lowercase L and not a number one. Don't worry about all the information shown; you will be learning about this information later.
10. Take a look at the items at the root (/) filesystem by typing **ls /** and pressing Enter. Can you tell whether these are directories or not? (It’s okay if you can’t tell. You’ll be able to tell in the next step.)
11. Add an indicator to the end listed item by typing **ls -F /** and pressing Enter. Remember that the forward slash (/) indicator tells you that the item is a directory, whereas a blank indicates the item is a file.
12. Now you will try something different and attempt to change your present working directory to various places across the filesystem using relative and absolute directory references. First navigate to the /etc directory by typing **cd /etc** and press Enter. Entering a directory or file reference starting with a forward slash (/) is called using an "Absolute Directory Reference."
13. Now see where your present working directory is currently located by typing in **pwd** and pressing Enter. This indicates your present working directory is now at the /etc directory.
14. Use another absolute directory reference to take you back to the user account’s home directory by typing cd directory and pressing Enter, where directory is your account’s home directory (for example for a user called rick this would be cd /home/lamp), which you recorded in step #1. Remember that an absolute directory reference ABSOLUTELY starts with a forward slash (/).
15. Check that you are in your account’s home directory by typing in **pwd** and pressing Enter.
16. Try using a relative directory reference to enter a subdirectory under your account’s home directory. Type **cd mail** and press Enter.
17. Check that you are in the Downloads directory (or the other directory you chose) by typing in **pwd** and pressing Enter.
18. Use another method to take you back to your home directory by typing: cd ~ and pressing Enter. The symbol after the cd command is a tilde (~) symbol. The tilde (~) symbol is typically located above the Tab key on your keyboard.
19. Check that you are in your home directory by typing in **pwd** and pressing Enter.
20. Now you will navigate up one directory level using a relative directory reference. Type **cd ..** and press Enter. The two dots or periods (..) are a relative directory reference meaning “one directory level up.” This is sometimes called the “parent directory.”
21. See where you are by typing **pwd** and pressing Enter.
22. Now try something a little different: use just one dot (or period) with the cd command. Type **cd .** and press Enter.
23. See where you are by typing **pwd** and pressing Enter. You stayed in the exact same place! This is because the single dot or period (.) is a relative directory reference meaning “current directory.”
24. Type **cd /etc**
25. Now navigate to your home directory using another relative directory reference:

Type **cd**  **$HOME** and press Enter. Be sure to use the correct case ($HOME is all upper-case letters) or this command will not work correctly) $HOME is an environment variable that points to your current home directory.

1. Check that you are in your home directory by typing in pwd and pressing Enter.
2. Now try to navigate up two directory levels using a relative directory reference. Type **cd ../..** and press Enter.
3. Type **pwd** and press Enter to see where your present working directory is currently located.
4. Use the cd command to navigate to your account’s home directory using one of the four methods explored in this exercise.

**Exercise 3: Looking at Command History**

1. While in the tty2 virtual terminal, type **clear** and press Enter. This will clean up the displayed text in terminal.
2. To see all your command history (commands you have previously entered in the bash shell), type **history** and press Enter. You should see several commands listed along with a number by each one.
3. Normally, the history command displays information kept in a hidden file, which is located in your home directory. To see this file:
   1. Type **cd ~** to go to your home directory in case you are currently not there
   2. Type **cat .bash\_history** and press Enter.
4. Type the following command: **history > my\_history** .This saves the output of the history command to a file called “my\_history”. Next, type the following command: **cat my\_history**

*\*\*\** ***SS1:*** *Take a snip of the terminal window showing the above command and what the resulting display is. Create a file coos181Linux1CST1##.docx (where 1## is your CST number), this will be your lab worksheet. Type the text SS1, then paste the clip in the file. \*\*\**

1. Type **history** and press Enter again to view your recent commands.
2. Pick one of the commands from the history list and note its number (for example find the number of the ls –l command you used to show a long directory listing). Type **!#** (where # is the number of the command in your history) and press Enter. The command should display and execute! This is another method, besides the up arrow key, you can use to recall previous commands.

**Exercise 4: Creating and Using New Subdirectories**

1. In the virtual terminal, type **cd ~** and press Enter, then type **pwd** and press Enter. The pwd command shows your present working directory (in this case, it is your home directory).
2. Type **ls -F** and press Enter. You should see a file name with a / at the end of its name, indicating that it is a subdirectory.
3. Type **mkdir /home/lamp/NewDir** press Enter. This will create a new subdirectory in /home/lamp/ called NewDir. You are using an absolute directory reference with the mkdir command to create the new subdirectory.
4. Type **ls -F** and press Enter. You should see your newly created subdirectory.
5. Type **cd /home/lamp/NewDir** and press Enter. This will change your present working directory to the new subdirectory. Notice that you used an absolute directory reference with the cd command.
6. Type **ls -F** and press Enter. You should not see any files in the new subdirectory you just created. Why? You haven’t put any files in there yet!
7. Type **touch MyFile** and press Enter. The touch command either creates an empty file or updates the access and modification time stamps of a file. In this case, you just created a blank empty file called MyFile.
8. Type **ls -F** and press Enter. You should now see the file you just created, called MyFile in the new subdirectory.
9. Type in **ls -F /home/lamp/NewDir** and press Enter. You should see the same file listed as in the preceding step. In the preceding step, you looked at the files in your present working directory. In this step, you used an absolute directory reference to see files listed there.
10. Type **cd ..** and press Enter. (Don’t miss those two periods after the cd command, or this step won’t work.) This will move your present working directory from the new subdirectory up one level to your HomeDirectory.
11. Check that you are back at your HomeDirectory by typing **pwd** and pressing Enter.
12. **Warning:** This next command will generate an error. At the command prompt, type **mkdir Stuff/MoreStuff** and press Enter. You should get an error message similar to: “mkdir: cannot create directory ‘Stuff/MoreStuff’: no such file or directory.” This mkdir command attempts to create a subdirectory called Stuff and another subdirectory within Stuff called MoreStuff. However, a subdirectory cannot be created if a parent directory does not exist. The mkdir command knows the Stuff subdirectory does not exist yet, so it refuses to create anything.
13. Type **mkdir -p Stuff/MoreStuff** and press Enter. The –p option forces the mkdir command to create both the Stuff subdirectory and the MoreStuff subdirectory. Notice that you used a relative directory reference to create the /home/student/Stuff/MoreStuff subdirectories.
14. Type **ls -RF Stuff** and press Enter. The MoreStuff subdirectory should display.

*\*\*\** ***SS2:*** *Take a snip of the terminal window showing the above command and what the resulting display is. In the lab worksheet type the text SS2, then paste the clip in the file.\*\*\**

1. **Warning:** The next command will generate an error. Type **rmdir Stuff** and press Enter. You should get an error message similar to “rmdir: failed to remove ‘Stuff’: Directory not empty.” This rmdir command attempts to delete a subdirectory. However, if the directory contains subdirectories and/or files, it will not work, as demonstrated here.
2. Type **rmdir Stuff/MoreStuff** and press Enter. You got no error messages! Why? Because the subdirectory MoreStuff is empty. Notice you used a relative directory reference to remove the subdirectory MoreStuff.
3. Type **rmdir Stuff** and press Enter. You got no error messages, because the subdirectory Stuff is now empty (you deleted the subdirectory contents in the preceding step.)
4. Next you will attempt to use the rmdir command to delete the NewDir subdirectory you created earlier. Do you think this will work? Using the home directory, type **rmdir /home/lamp/NewDir** and press Enter. It did not work. Do you know why? (Look back at step #10 if you need some help answering this question).
5. Remove both the MyFile file from NewDir and the directory by typing the command **rm -iR /home/lamp/NewDir** and pressing Enter. When the command asks something similar to “descend into directory?” type y and press Enter. Type **y** and press Enter for each item it asks to remove. This will remove the NewDir subdirectory and all its contents.
6. Type **ls -F** and press Enter. Is the NewDir subdirectory gone?

**Exercise 5: Copying Files**

1. To move to you home directory, Type **cd ~** and press Enter, then type **pwd** and press Enter. The pwd command shows your present working directory (it should be your home directory).
2. Type **touch chores** and press Enter. This will create an empty file in your present working directory (pwd) called chores.
3. Type **ls chores** and press Enter. You should see the chores file listed.
4. Type **cp chores chores.bck** and press Enter. This will create a copy of the file chores and give it a new name, chores.bck.
5. Type **ls chores\*** and press Enter. You should see the files chores and chores.bck.
6. Type **mkdir Todo** and press Enter. This will create a subdirectory in your home directory called Todo.
7. Type **ls -F** and press Enter. You should see the subdirectory Todo.
8. Type **cp chores /home/lamp/Todo/** and press Enter. This will copy the file chores into the subdirectory you just created, Todo.
9. Type **ls Todo** and press Enter. You should now see a copy of the file chores in the subdirectory Todo.
10. Type **cp chores Todo/done** and press Enter. This will copy the file chores from your home directory to the subdirectory, Todo, and rename the file to done.
11. Type **ls -F Todo** and press Enter. The file done should be there.
12. Type **cp -R Todo NewTodo** and press Enter. This cp command with the –R option will copy an entire subdirectory and all of its contents to a new subdirectory. In this case, you are copying the Todo subdirectory and all of its files to a new subdirectory called NewTodo.
13. Type **ls -F Todo** and press Enter. Note the files located here.
14. Type **ls -F NewTodo** and press Enter. Note the files located here. Do they match the files in the preceding step? They should, because you copied them over into this new subdirectory in step #12.

*\*\*\** ***SS3:*** *Take a snip of the terminal window showing the above command and what the resulting display is. In the lab worksheet type the text SS3, then paste the clip in the file.\*\*\**

1. Type **rm -Ri NewTodo** and press Enter. Type **y** and press Enter to all the questions asking if it is okay to descend into the directory, delete the files, etc. Remember, the rm command with the –R option descends down into the subdirectory to delete all the files, and the –i option asks you if it is okay to delete the files/directories.
2. Type **rm -Ri Todo** and press Enter. Type **y** and press Enter to all the questions asking if it is okay to descend into the directory, delete the files, etc.
3. Type **rm -i chores\*** and press Enter. Type **y** and press Enter to the question asking if it is okay to delete the files.

**Exercise 6: Moving Files**

1. Type **cd ~** and press Enter, and then type **pwd** and press Enter. The pwd command shows your present working directory (it will be your home directory).
2. Type **touch math** and press Enter. This will create an empty file called math.
3. Type **ls math** and press Enter. You should see the file math located in your home directory.
4. Type **mv math algebra** and press Enter. This will rename the file math to algebra.
5. Type **ls -F** and press Enter. The file named math should not be shown, because it has been renamed (using the mv command) to algebra. Of the two files, only the file named algebra should be shown.
6. Type **mv algebra gym** and press Enter. This will rename the file algebra to gym.
7. Type **ls -F** and press Enter. The file named algebra should not be shown, because it has been renamed (using the mv command) to gym. Of the two files, only the file named gym should be shown.
8. Type **mkdir School** and press Enter. This will create the subdirectory School in your home directory.
9. Using the home directory you recorded in step #1 for HomeDirectory in this command, type **mv gym /home/lamp/School/** and press Enter. This will move the file gym to the subdirectory School.
10. Type **ls -F** and press Enter. This will list all the files and subdirectories in your home directory. The file gym should be gone, because you moved it into the subdirectory School.
11. Type **ls -F School** and press Enter. The file gym should now be showing, because you are listing out the contents of the subdirectory School where you moved the file gym to.
12. Type **touch lunch** and press Enter. This creates a blank empty file called lunch in your home directory.
13. Type **mv lunch School/** and press Enter. This does NOT rename the file lunch to School, because School is a subdirectory in your pwd. Instead, this moves the file lunch to the subdirectory School.
14. Type **ls –F School/** to confirm this.
15. Type **touch reading** and press Enter. This creates a blank empty file called reading in your home directory.
16. Type **mv reading School/history** and press Enter. This step moves the file reading to the sudirectory School and then renames it to history.
17. Using the home directory you recorded in step #4 for HomeDirectory in this command, type **ls /home/lamp/School** and press Enter. You should see three files located in the subdirectory School and they should be: gym, history, and lunch.

*\*\*\** ***SS4:*** *Take a snip of the terminal window showing the above command and what the resulting display is. In the lab worksheet type the text SS4, then paste the clip in the file.\*\*\**

1. Type **mv School College** and press Enter. This renames an entire subdirectory named School to College.
2. Type **ls -F College** and press Enter. You should see that the three files gym, history, and lunch are now located in the subdirectory College.
3. Type **ls -F School** and press Enter. You should get an error message on this step. That is because this subdirectory no longer exists. You renamed School to College in a preceding step—therefore the subdirectory School no longer exists.
4. Type **rm -R College** and press Enter. This should remove all files and directories within College with interactively asking you. Remember, use the rm command without the –R option to descend down into the subdirectory College to delete all the files, and the –i option to ask you if it is okay to delete the files/directories. –R alone deletes without asking.
5. Type **ls** to confirm College has been removed.

**\*\*\* Save the file coos181Linux1CST1##. Submit it to Brightspace dropbox, “Linux File Management Lab” \*\*\***