CIT117 Name: \_\_\_\_Alex DeGhetto\_\_\_\_\_\_\_\_\_\_

Unix/Linux Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lab # 5 – Working with the BASH shell

**Objectives**:

• Bash shell commands

• Piping and redirection

• Grep, sed, and awk

• Basic shell scripting

**Procedures**:

1. Use the man utility to determine the usage of the ‘last’ command. What does the command do?

Shows a listing of the last logged in users.

2. Run the ‘last’ command and list the first few lines of output.

Occc :0 :0 Wed Apr 19 18:39 still logged in

Reboot system boot 5.15.0-58-genri Wed Apr 19 18:39 still running

Occc :0 :0 Wed Feb 15 20:12 - down

3. Run the following command

last | awk '{print $1}'

List and explain the output of the command:

The output only has the first column which is the user

4. Use a shell utility to sort the output of the ‘last’ command from question 3 above. What commands did you use?

Last | awk ‘{print$1}’ | sort

5. Use the man utility to research the ‘uniq’ command. What does the command do?

To report or delete repeated lines.

6. Use the commands from the previous few questions to list the unique users that have recently logged in to your machine. List the users along with the number of times they logged in. What commands did you use to accomplish this?

Last | awk ‘{print$1}’ | sort | uniq -c

How might we alter the above command so that we could list all of the unique IP addresses/hosts that users logged in from? What commands would you use?

Last -i | awk ‘{print$3}’ | sort | uniq -c

How might we alter the above command so that we could list all the unique dates that user logged in. What commands would you use?

Last -i | awk ‘{print$3}’ | sort | uniq -c

7. Use the grep and wc commands along with the information in the /etc/passwd file to discover how many users on your system use the bash shell. What commands did you use to accomplish this?

Last -F | awk ‘{print$3,$4,$5,$6,$7,$8,$9,$10}’ | sort | uniq -c

8. Alter the bash prompt to read: “Awaiting command: “

What command did you issue to achieve this?

Edited bash w

Sudo nano ~/.bashrc

Added the line PS1=“Awaiting command: >”

Source ~/.bashrc

9. Create a directory called logs in your home directory. Create an empty file in the logs directory named user.log. What commands did you use to accomplish this?

I did mkdir logs

Then cd logs

Then ls

Then touch user.log

What is the absolute path to the logs directory?

/home/occc/logs

Create a user-defined shell variable that holds the absolute path to your new log directory. Make sure to export it for all subshells. What command did you use to accomplish this?

Export woah=/home/occc/logs

10. Explain the operation and output of the command:

sed ‘s/:.\*//’ /etc/passwd | tr [:lower:] [:upper:]

It edited the /etc/passwd file

11. Use the man pages to determine how to take a long directory listing with the file sizes in “human readable format”. Create an alias to this command named lh. Test the alias to make sure it works. What commands did you use to accomplish this?

I did ls -l -h then I did alias lh=‘ls -l -h’

12. Cat the contents of the /etc/nsswitch.conf file. Redirect the output of the command to a file named newns in the /tmp directory. What commands did you use to accomplish this?

Cat /etc/nsswitch.conf | /tmp/newns

Use the vi editor to change the last line of the newns file to read:

“Hello from the newns file”

How did you accomplish this?

Vim /tmp/newns

Use the man utility to research the diff command. Use diff to compare the /etc/nsswitch.conf file to the newns file you created in the above steps. What commands did you use to accomplish this?

Diff /etc/nsswitch.conf /tmp/newns

What is the output of the diff command?

“Hello from the newns file”

13. Run the command:

cat /etc/noFile

What does the command output to the screen?

No such file or Directory

Run the command again, redirecting the standard output of the previous command to a file named /tmp/noFileOut?

What command did you use to accomplish this?

Cat /etc/noFile | /etc/noFileOut

What is printed to the screen when you run the command?

No such file or directory

Explain why a message is printed to the screen, even though you redirected the output.

WE still cat it so we see the contents.

How would you redirect the message to the file /tmp/noFileOut?

The same as before?

How would you redirect both the standard out and the standard error to the file /tmp/noFileOut?

Cat /etc/noFile -e | /tmp/noFIleOut

14. Write a shell script that prints the following menu:

SUNY Orange File Manager

C)hange into a directory

L)ist the files in the current directory

M)ove a file

K)opy a file

P)rint the contents of a file

Command:

Have the script read in the user’s choice and use appropriate shell commands to execute the stated function, prompting the user for any necessary arguments. For example, if the user choosed ‘p’, prompt the user for a file name, and print out the contents of the file. Print out your shell script and attach it to this lab.

It won’t let me upload the menu.sh so you can run it but here is the code:

#! /bin/bash

# Present the menu

echo "

SUNY Orange File Manager

C)hange into a directory

L)ist the files in the current directory

M)ove a file

K)opy a file

P)rint the contents of a file

"

# Read input from user

read userCommand

# Ask for path based on choice and validate data

alias c='cd' C='cd' l='ls' L='ls' k='cp' K='cp' m='mv' M='mv' P='cat' p='cat'

if [[ $userCommand = [CclL] ]]

then

echo "What is the directory path"

elif [[ $userCommand = [pP] ]]

then

echo "What is the file path"

elif [[ $userCommand = [mMkK] ]]

then

echo "What is the source file path"

else

echo "$userCommand is not a valid choice"

exit

fi

# Ask for path

read path

# Run command or ask for target path

if [[ $userCommand != [mMkK] ]]

then

echo "Running command: $userCommand $path"

eval $userCommand $path

if [[ $userCommand = [Cc] ]]

then

echo "Changed location to:"

pwd

fi

exit

else

echo "What is the target folder path"

fi

# Get target path folder

read targetPath

# Run either move or copy file command

echo "Running command: $userCommand $path $targetPath"

eval $userCommand $path $targetPath