**Srikar D. and Mohammed Ali**

**CSS343**

**January 19, 2020**

**1.** Get the manual for ls

**‘**$ man ls**’**

**2.** Search the manual page names and descriptions for anything containing git

man git

**3.** What happens when you type l followed by TAB TAB

This command returns all the results and bin

**4.** Print the name of the current working directory

pwd

**5.** After you execute ls -al .bash\_profile, what is the shortcut to repeat the same command

Up arrow and enter

**6.** After you execute ls -al .bash\_profile, how do you execute a modified version editing the line

Up arrow and then edit the line

**7.** Machine name (something like csslab1.uwb.edu)

‘hostnamectl’

Uw1-320-15

**8.** Command to get the IP address

Ifconfig -a

**9.** Find the IP address (or addresses) for google.com

‘dig google.com’

216.58.193.78

**10.** The Operating System

uname

**11.** Number of processes that are currently running on the machine

‘ps aux | wc -1’

181

**12.** The process that is using the largest percentage of memory

top and then Shift+M to sort by memory

**13.** The process that is using the least amount of CPU

‘top’

cpuhp/2

**14.** The list of processes without the header line (start from 2nd line)

ps --no-headers

**15.** Keyboard input to stop running the command python -c 'while True: 1\*1'

Depends if on Mac or Windows

Control/ctrl + c or z

**16.** Run python -c 'while True: 1\*1' in the background. Find all running processes that have the work python in them

python -c 'while True: 1\*1' &

pgrep python

**17.** Find all running processes that have the word python AND while in them

‘ps -f’

**18.** Find and terminate the process that has the word python AND while in it

ps -f | grep python | grep while| awk '{print $2}' | xargs kill

**19.** Your username

‘echo “$USER”’

**20.** What groups do you belong to

id -Gn

**21.** What shell are you using

‘echo $SHELL’

**22.** Create an empty file named funny

Touch funny

**23.** Print a long listing of the file funny, so you can see its permissions

‘ls -la funny’

**24.** Make funny to be read-write-executable by you and no-one else

chmod 700 funny

**25.** Add the command ls -al / as the first line of funny (without using an editor)

‘vim funny’

Key command: “I”

Insert ‘ls -al /’

Press: ESC and type “:wq”

**26.** Execute funny

./funny

**27.** Add the command sleep 10 as the second line of funny (without using an editor)

‘vim funny’

Key command: “I”

Insert ‘sleep 10’

Press: ESC and type “:wq”

**28.** Copy funny to not-so-funny AND preserve its permissions

cp -p funny not-so-funny

**29.** Make not-so-funny to be readable by everybody, but not writable or executable by anybody

‘chmod 004 not-so-funny’ make it readable to everyone

‘chmod 200 not-so-funny’ make it editable only by owner

‘chmod 100 not-so funny’ make it executable only by owner

**30.** Delete not-so-funny without getting a confirmation prompt

rm -f not-so-funny

**31.** Use scp to copy funny to /tmp in csslab9.uwb.edu machine with the new name funny-XXX where XXX is your username

‘mv funny funny-myousefa’

‘scp funny-myousefa [myousefa@csslab9.uwb.edu:/tmp/](mailto:myousefa@csslab9.uwb.edu:/tmp/)’

**32.** List all files in /tmp in csslab9.uwb.edu, use ssh but do not login to csslab9

ssh [srikardu@csslab9.uwb.edu](mailto:srikardu@csslab9.uwb.edu) ls /tmp

**33.** Execute /tmp/funny-XXX on csslab9.uwb.edu, use ssh but do not login to csslab9, XXX is your username

‘ssh [myousefa@csslab9.uwb.edu](mailto:myousefa@csslab9.uwb.edu)’

‘cd’

‘ls /tmp’

‘./funny-myousefa’

**34.** Count how many files there are in /usr/bin that has the word zip in them

ls /usr/bin | grep zip | wc -l

**35.** List all files in /usr/bin that has the word yum in it using find

ls /usr/bin/ | grep yum | wc -1

**36.** Make a directory called yummies

mkdir yummies

**37.** Copy all files in /usr/bin that has the word yum in it into the directory yummies

‘cp /usr/bin/yum\* /home/NETID/myousefa/yummies/ | grep yum’

**‘**cd yummies/’

‘ls -la’

**38.** Create a zip file, my-yummies.zip, that includes the directory yummies

zip -r my-yummies.zip yummies

**39.** Create a compressed tar file, my-yummies.tar.gz, that includes the directory yummies

**‘**tar -zcvf my-yummies.tar.gz /home/NETID/myousefa/yummies/’

**40.** Append my-yummies.zip to big-file 3 times

seq 1 3 | xargs -Inone cat my-yummies.zip >> big-file

**41.** Check the file size of big-file in a human readable format

**42.** You want to download the .nanorc file at <https://gist.github.com/jclosure/63820f48cb6551a43a0b272c3542389f> The raw file URL is <https://gist.githubusercontent.com/jclosure/63820f48cb6551a43a0b272c3542389f/raw/093eff95fcfc6c3c547d9854ebb9a027bb285558/.nanorc> Use wget do download it

wget <https://gist.githubusercontent.com/jclosure/63820f48cb6551a43a0b272c3542389f/raw/093eff95fcfc6c3c547d9854ebb9a027bb285558/.nanorc>

**43.** Use git to clone https://github.com/pisan343/hello

‘git clone https://github.com/pisan343/hello’

**44.** Change into the directory hello and execute simplecompile.sh in that directory

cd hello/

./simplecompile.sh

**45.** Fix main.cpp and if necessary simplecompile.sh so you do not get any memory leak warnings or style warnings. Use nano as your text editor

**no bites lost when running ‘**valgrind ./simplecompile.sh --leak-check=full’

**46.** Print main.cpp on the standard output

cat main.cpp

**47.** Display main.cpp in the standard output waiting for a key at the end of each page and with the ability to go back pages

‘less main.cpp’

**48.** Display the time it takes to execute sleep 1 in terms of total time elapsed and the time consumed by system overhead

time sleep 1

**49.** The full path for the executable for ls located

‘which ls’

**50.** (Not a unix command) Why is <https://xkcd.com/149/> funny?

Sudo puts you in the seat of the superuser. So, since you’re the boss, you can do it yourself.