

1. Enter the following commands at the Linux prompt and interpret the output.

a. **whoami**

Show who the current logged on user is.

More precisely, print the user name associated with the current effective user ID.

b. **who**

With no options, display information about the names of users currently logged in, their (pseudo)terminal, the time they have been logged in, and the name of the host from which they have logged in.

c. **w**

Show who is logged on and what they are doing (i.e. what processes they have running). Specifically, the following information is displayed:

i. **USER**: login name

ii. **TTY**:

PTS is a Pseudo Terminal Slave. The is the client component of a pseudo terminal device, which itself is a virtual console/terminal connected to a TTY (a Teletypewriter which means any terminal on a Unix/Linux system). The main difference between TTY and PTS is the type of connection to the computer. TTY ports are direct connections to the computer such as a keyboard/mouse or a serial connection to the device. PTS connections are SSH connections or telnet connections. All these connections can connect to a shell which will allow you to issue commands to the computer.

iii. **FROM**: The remote name. In the case of the Google Cloud Shell, it is localhost.

iv. **LOGIN@**: The login time.

v. **IDLE**: The idle time.

vi. **JCPU**: The JCPU time is the time used by all processes attached to the tty. It does not include past background jobs, but does include currently running background jobs.

vii. **PCPU**: The PCPU time is the time used by the current process, named in the "what" field

viii. **WHAT**: The absolute fullpath name of the command line of their current process.

2. State what these commands do.

a. **pwd** Displays the full path name (absolute path) of your current directory/folder.

b. **cd path**: Change to the directory specified by path.

c. **cd ~** : Change to your home directory – what happens?

d. **cd -** : Change to your previous directory (not parent directory) if you were previously in a different directory.

e. **cd ..** : Change to the current directory's parent – what directory are you in now?

f. **cd ../..** Change to the current directory's grandparent folder.

g. **cd** Change to the HOME folder of the current logged on user.

3. Try to interpret the output

a. **echo hello world** Send (echo) the input *string* to standard output. This is the /bin/echo command

b. **echo -e "Hello \bOSF1 \nClass \v I love this class"** The -e command line switch to echo enables interpretation of backslash escapes and other escape sequences. Thus \b performs a backspace, \n inserts a new line, \v inserts a vertical tab.

- c. **date** Print the current date and time
- d. **date "+%Y-%m-%d %T"** Display the date in the format YYYY-MM-DD HH:MM:SS
- e. **hostname** display or set the name of current host system.
- f. **uname** Print Kernel name.
- g. **uname -a**  
Print information about the machine and operating system. With **-a** prints all the system information.  
Specifically:
  - i. kernel-name
  - ii. nodename
  - iii. kernel-version
  - iv. machine-processor
  - v. Kernel-release
  - vi. hardware-platform
  - vii. operating-system
- h. **uptime** Print the current time, how long the system has been running, the number of users currently logged in (which may include the same user multiple times), and the system load averages for the last 1, 5 and 15 minutes.
- i. **man ls** (*Press q to quit*) Display information from the online reference manuals on the ls command
- j. **man who** (*Press q to quit*) Display information from the online reference manuals on the who command
- k. **clear** Clear the terminal display. Equivalent to pressing Ctrl-L
- l. **du -hs ~** By default, DU estimates the disk space usage for a specified set of files or path (recursively). The switch **-h** means print in human readable format. The switch **-s** means display only a summary total for each argument (do not recurse).
- m. **du -h ~** Same as above except recursively for all subfolders.
- n. **df -h** Report file system disk space usage for entire file system.

#### ***4. Use the linux command line to perform the following steps. Show the command you use for each step***

- a. Create a file called test.txt in your HOME directory. To create a file you can use the touch command. The format of this command is : touch file1

**touch test.txt**

- b. Create a folder called Lab04 in your home directory and change into that folder

**mkdir Lab04 && cd Lab04**

- c. Copy the file test.txt from the home directory into your Lab04 directory

**cp ~/text.txt Lab04**

- d. If the destination file exists before you give a copy command, the copy command will overwrite it. For example in c above. if a file called test.txt already exists in the Lab04 directory, then it will be overwritten if you use the basic form of the copy command. How can you modify the copy command to ensure that the system will prompt before carrying out such an operation.

**cp -i ~/text.txt Lab04**

This is the "interactive" form of the cp command – uses the -i option.

- e. Make 2 directories in the Lab04 directory – myDir1 and myDir2.

**mkdir myDir1**

**mkdir myDir2**

- f. Change into myDir2. **cd myDir2**

- g. Move test.txt to myDir2 **mv ../test.txt .**

- h. What command do you use to find out information such as the owner and size of the file that you just moved? **ls -l test.txt**

- i. Rename test.txt to happy.txt? **mv test.txt happy.txt**

- j. Create another file in your home directory: call it Listing.txt **touch Listing.txt**

- k. Redirect the output of the `who` command to `Listing.txt`

(The redirection operators are the same as in windows: `>` and `>>`)

**`who > Listing.txt`**

- l. Use the `cat` command to print the contents of the file `Listing.txt` on the screen.

(Read/concatenate one or more *files* and print them on standard output).

**`cat Listing.txt`**

- m. Append the output of `ls -l` to the `Listing.txt` file

**`ls -l >> Listing.txt`**

- n. Print the contents of `Listing.txt` on the screen.

**`cat Listing.txt`**

## Question 5

- a. Use the `cat` command and the redirection operator to create an empty file called `q5.txt`. You should use `<ctrl-D>` after entering the command – this will create an empty file

**`cat > q5.txt`**

- b. Use the interactive form of `rm` to delete this file – using the interactive form the system should prompt you with the name of the file before it is deleted.

**`rm -i q5.txt`**

- c. Create a file called `courses.txt` using `cat` and redirection. Enter a few lines of text before using `<ctrl-D>` to return to the prompt.

**`cat > courses.txt`**

**`os`**

**`maths`**

**`sdev`**

**`comm.`**

**`<ctrl-D>`**

**d.** Print the contents of this file (`courses.txt`) to the screen.

```
cat courses.txt
```

**e.** Search for a string in this file using the **grep** command

- For example **grep 'de' courses.txt** will search for all strings in the file `courses.txt` that contain the pattern "de"
- **grep os courses.txt** will search for the string "os" in `courses.txt`