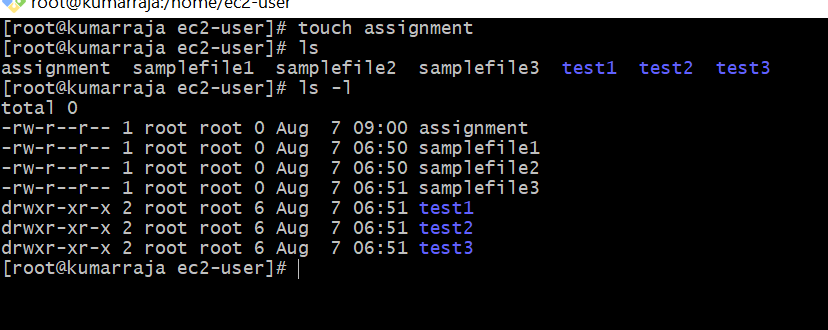
**Assignment 1**

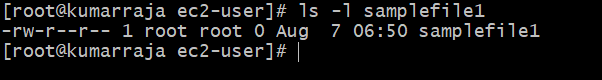
1)List the contents of a directory and their attributes

A: ls



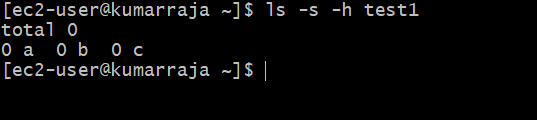
2)Long list the content with file, directory ownership, permissions,sizes, etc…

A: ls -l



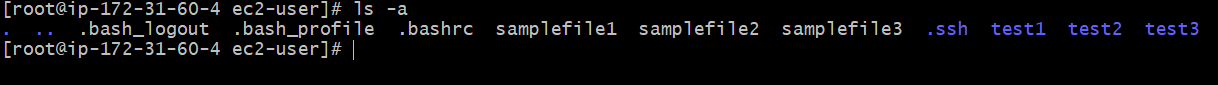
3)Display the size of the file in human readable format

A: Ls –s –h



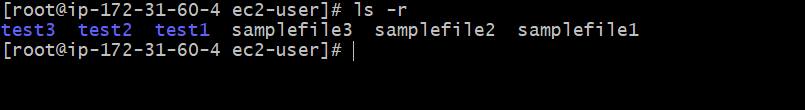
4)Show all files and folders including hidden one

A: ls –a



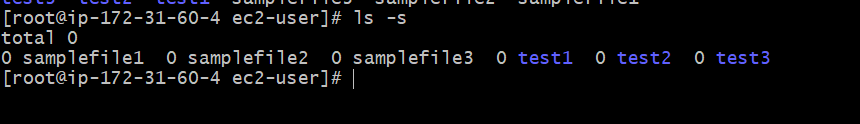
5)list directories recursively

A: ls -r



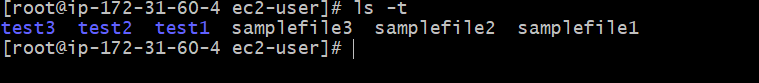
6)Sort the files by size with largest at the top

A: ls –s



7)Sort the files by last time modified displaying the newest first.

A: ls –t



8)Display the location of a program/command, where it is installed.

A: If it is in your path, then you can run either type git or "which git". The which command has had problems getting the proper path (confusion between environment and dot files). For type, you can get just the path with the -p argument.

If it is not in your path, then it's best to look for it with "locate -b" git It will find anything named 'git'. It'll be a long list, so might be good to qualify it with "locate -b git | fgrep -w bin".

9)Which command is used to switch directory from one to another

A: cd

10)List all the environment variables set for the current shell environment

A: env

11)Did you notice something in the output of "env" command?

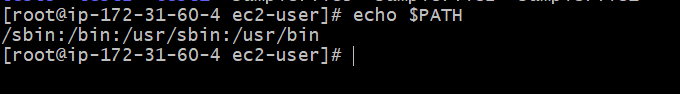
A: We can get all environmental variables (shell type, user name ,etc)

12)Which command is used to print the text or any variables value in the Console/Terminal?

A: echo

13)Print the value of the env variable "PATH" on the console

A: echo $PATH



14)Is linux a case-sensitive operating system?

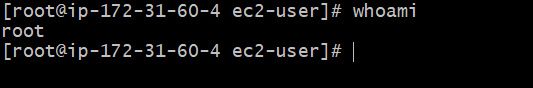
A: Yes

15.Is, "Ls" same as "ls"?

A: Yes

Display your currently logged in user

A: whoami



16)how do you change the currrently logged in user to another user?

A: su - username

17)Which command is used to leave a shell environment that you are currently logged in to?

A: chsh

18)How do you reboot the system?

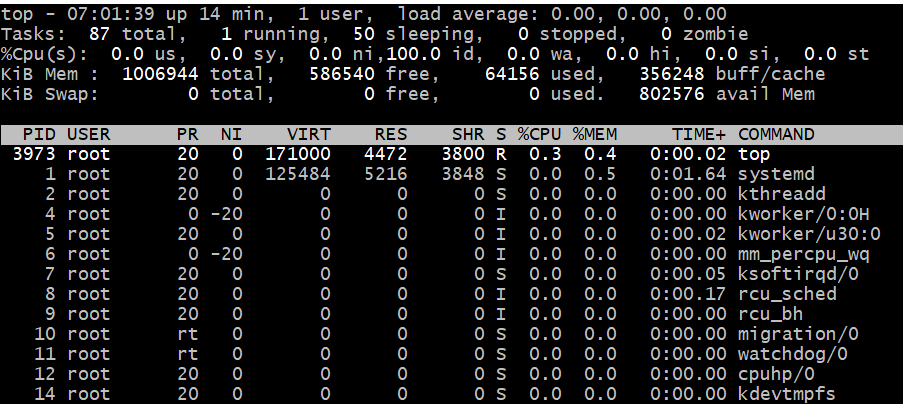
A: init 6

19)How do you shutdown the system?

A: init 0

20)Display all the major running processes in the system

A: top



21)Understand the output of command used in above question of displaying processes, explain the meaning of each column and what data it displays?

A: top displays the current time.followed by the system uptime, which tells us the time for which the system has been running.the number of active user sessions

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

PID: Process ID

USER: username

PR: shows the scheduling priority of the process from the perspective of the kernel

NI: nice” value of a process,The nice value affects the priority of a process

VIRT: total amount of memory consumed by a process

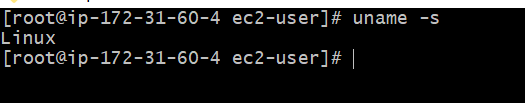
RES: the memory consumed by the process in RAM

MEM: expresses this value as a percentage of the total RAM available

SHR: the amount of memory shared with other processes

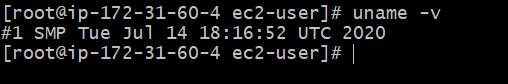
22)Display the name of the system kernel

A: uname –s



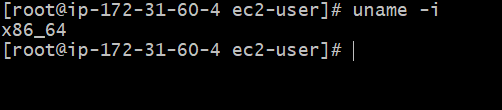
23)display the kernel release number

A: uanme –v



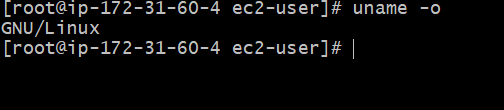
24)Display teh machine type of the current kernel

A: uname –i



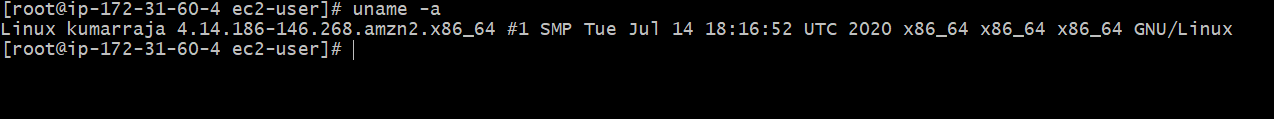
25)Display the name of the operating system that the kernel is running on

A: uname -o



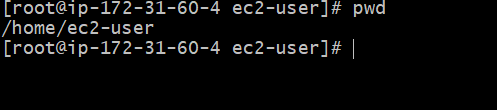
26)Display all info that uname command can show.

A: uname -a



27)Display the name of directory that you are currently pointing to

A: pwd



28)change the current directory to another directory that you have in your system.

A: cd absolute path

29)Go up one directory

A: cd ..

30)Return to last directory

A: cd

31)Change the current directory to home(logged in user's) directory

A: cd ~user name

32)How to check all the command used from the prompt (Command History)

A: history

33)In which file the history of commands are stored in?

A: .bash\_history

34)How many lines of history does the system keep and from where you can change it?

A: 500 you can delete some but can't change history

35)How can you modify bash's history behaviour

A: Edit the ~/.bashrc file and add the following line:

$ PROMPT\_COMMAND='history -a'

Now whenever you execute any command,it will be immediately added to the history file

NOTE:- TAB key is your friend when it comes to command completion and having long file and directory names autocompleted at the bash prompt for you. JUST BE LAZY AND USE TAB KEY FOR AUTO COMPLETION ;-)

36)What are the different types of shell and where are they used and how do we use them?

A:

Bourne Shell : It is faster and more preferred. It lacks features for interactive use like the ability to recall previous commands

C Shell :It includes helpful programming features like built-in arithmetic and C-like expression syntax

Korn Shell : It includes features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities.It is faster than C shell. itis compatible with script written for C shell.

Bourne-Again Shell: It is compatible to the Bourne shell. It includes features from Korn and Bourne shell

37)What is the difference between login shell and non-login shell?

A: login shell: A login shell is the first process that executes under your user ID when you log in for an interactive session

non-login shell: When you log in on a text console, or through SSH, or with su -, you get an interactive login shell

38)How do we start login shell and non-login shell?

loginshell=yes : Place in .profile file

Non loginshell : These shells don't read .login or .profile. In addition, bash allows a nonlogin shell to read ~/.bashrc or not

39)What happens when you start a login shell (which files are read and used and Why)?

it first reads and executes commands from the file /etc/profile, if that file exists. After reading that file, it looks for ~/.bash\_profile, ~/.bash\_login, and ~/.profile, in that order, and reads and executes commands from the first one that exists and is readable

40)What happens when you start a non-login shell (Which files are read and used and Why)?

no login means only prices can be run

While shell login ID disabled. Only files with full perm and own perm can be read

41)What are Shell Configuration Files, why do we need it?

A: The /etc/profile file – it stores system-wide environment configurations and startup programs for login setup. All configurations that you want to apply to all system users’ environments should be added in this file.

42)Explain the Order of file usage from the system/user's home directory when user logs in to the System.

A. /home/username

43)What are Shell Variables, list major shell variables and what do they represent?

A variable is a character string to which we assign a value. The value assigned could be a number, text, filename, device, or any other type of data.

A variable is nothing more than a pointer to the actual data. The shell enables you to create, assign, and delete variables

1.Defining Variables

2.Accessing Values

3.Read-only Variables

4.Unsetting Variables

44)How we see all our env variables?

A: env

45)How we see all env variables in alphabical order?

A: env | sort

46)What Format does the env var and its values are stored?

A: Variables have following format

KEY=value

KEY="Some other value"

KEY=value1:value2

47)How do you create your own varible?

A: To create a variable merely choose a lower-case name for the variable and give it a value using an equal (=) sign. Make certain that there are no spaces on either side of the equal sign.

48)How do you start a new bash shell?

A; bash -c 'gnome-terminal -x cd /absolute-path && program\_name'

49)Difference between Local/Shell variables to Global Variable

A: 1.Local variable is declared inside a function whereas Global variable is declared outside the function.

2.Local variables are created when the function has started execution and is lost when the function terminates, on the other hand, Global variable is created as execution starts and is lost when the program ends

Show the real life use case of exporting a variable

export JAVA\_HOME=jdk-install-dir

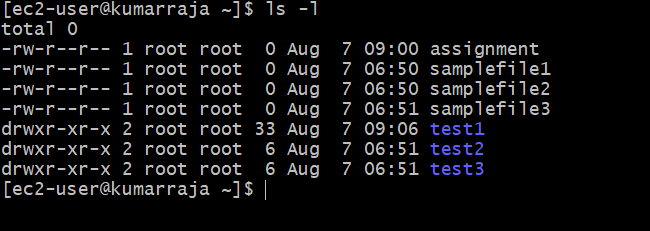
50)What is Globbing? Explain in depth with examples?

A: The Bash shell feature that is used for matching or expanding specific types of patterns is called globbing. Globbing is mainly used to match filenames or searching for content in a file

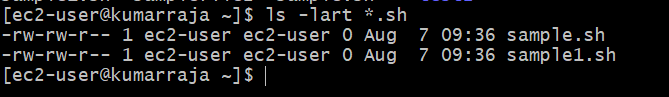
ls -l ????.txt

51)Display files end with extension "txt"

ls -l foot????.doc



52)List all entries with extension ".sh"

A: ls –lart \*.sh

List all entries with numbers in it.

A: ls -l [1-5] : you can give the set of range

List all entries that starts with a character and ends with a number

A: s -l {?????.sh,\*st.txt}, ls a\*+(5|7)

List all entries that name length more than 5 characters

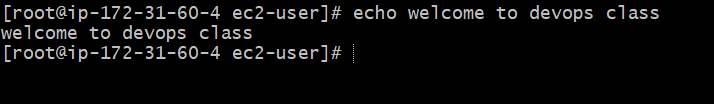
A : s -l {?????.sh}

What is Quoting? and Why do we need it?

A: Quoting is used to remove the special meaning of certain characters or words to the shell. Quoting can be used to disable special treatment for special characters, to prevent reserved words from being recognized as such, and to prevent parameter expansion.

59.Write few(minimum 3) unique examples that shows, how a particular problem is solved using Quoting.

A: #!/bin/sh



60)How do you find a particular files/directories based on a particular search criteria?

HINT:- look for commands -> locate, find and whereis

A: find . -name devops.txt: Search files on name

find / -type d -name devops: Search on directory

locate "\*.html" -n 20: show the first 20 files that end with 'html'

locate -i \*text.txt\*: Ignore case sensitive

whereis ls: search for ls command

61)Write major difference between locate, find and whereis?

A:

whereis: will search only particular paths to find binaries and or manpages. The manpages tells you where whereis looks.

locate: locate uses a database created by an updatedb to efficiently locate files. Works great, assuming your database is updated often enough to be reasonable upto date. Most boxes using locate have the updatedb occuring in cron.

find: find is perhaps one of the most powerful commands there is. For just locating a file/program of a particular name, it'll definitely be slower than locate or whereis becuase it will search each and every path recursively from it's start point

62)How Globbing is different from locate, find and whereis?

A: Globbing is find files based on charecters and numbers only.when it comes to locate, find and whereis find files by time, location, permissions, date, etc.

Explain the Linux File System.

A: A file system is a logical collection of files on a partition or disk. A partition is a container for information and can span an entire hard drive if desired.

Your hard drive can have various partitions which usually contain only one file system, such as one file system housing the /file system or another containing the /home file system.

Explain absolute and Relative Paths

A: An absolute or full path points to the same location in a file system, regardless of the current working directory. To do that, it must include the root directory. By contrast, a relative path starts from some given working directory, avoiding the need to provide the full absolute path

63)What are the different ways of creating a File in linux System? Write an example of each and the difference between them.

A:

Touch command

Cat command

Echo command

Printf command

Nano text editor

Vi text editor

Vim text editor

64)In how many ways we can delete the files from linux system? write an example of each and teh difference between them.

A:

rm apps.log

delete multiplr files at a time

rm test1.txt test2.txt test3.txt text4.txt

65)Archiving files using linux command, write a command to archive set of files from linux commands.

tar cf archive.tar test1 test2 test3

66)Extract the archived files from the above step.

tar xf archive.tar