Linux assignment

Name: Pranav H.Raikar sec: Cy(3b)

USN:ENG24CY0041 roll no: 39

1) **What is a shell in Linux OS? How many categories of shell is currently exists in Linux? Why bash shell is very popular in Linux distribution?**

Ans:

In the linux shell it act like a bridge between the user and the operating system. It takes the command you type and tells the system yo what is to do, then shows the result to you .

There are two category of shells in linux:

Command-line shells – this is where you type the commands (bash, sh, zsh).

Graphical shells – here where you interact with icons, menus, and windows and other things (GNOME Shell, KDE Plasma).

In all these the bash is the most popular thing . The reason is simple:

It will come pre-installed in almost every Linux distribution.

It is powerful easy to use and also supports scripting

also support thebackward compatible with older shells.

Therefor the bash is popular because it is user-friendly , powerful ,easy to use and widly supported.

2) **What does the ls -Z command display?**

Ans: The “ls” command just show the file names. But when we add –Z to the command it adds an extra column, which shows the **User,Role Type and Level.**

3) **Write a command to list all hidden files in the current directory.**

Ans: “ls -a” this command will list all file in the diertry if we want only hidden file we have to use the command (ls -d .*)

4) **Explains the differences between hard links and soft links (symbolic links) in Linux**

Ans: In the linux both **hard links** and **soft links is** to create shortcuts or references to the files, but they will work differently.

Hard Link

The hard link will give a file another name.

In that both the original file and the hard link pointed to the **same data file on the disk**.

If we deleted the original file from the system the hard link will still works.

Limitation is hard links cannot be link to the directories or files which is on different partitions.

Soft Link

soft link is a **shortcut**

It points to the **path** of the original files and not to the actual data.

If in the the original file is deleted then the soft link will be breaked .

5) **A file has permissions -rwxr-x--x. Explain who can read, write, and execute it.**

Ans: always the permition is in the form of [type] [owner] [group] [others]

So in this (“-”) It gives a **regular file.**

(“rwx”) it means the **owner** can **read, write, and execute**.
(r-x “) it means the **group** can **read and execute**, but **cannot write**.
(“--x “) it means **others** can also **execute**, but **cannot read or write**.

6) **Write the command to change the group ownership of a file data.txt to group staff.**

Ans: The command to change ownership i**s (“ chgrp staff data.txt”).** here the

Staff represents the new group name.

data.txt to the file whose group ownership you're changing.

7) **Why is it dangerous to give 777 permissions to a file? Explain with an example.**

Ans: in linux system file permissions are written as three numbers which is owner, group, others.

If we give the 777 give know it means:

Owner read, write, execute

Group read, write, execute

Others read, write, execute

Why this is dangerous

Because **anyone can do anything with the file** – read it, modify it, or delete it.

Example: command (" **-rwxrwxrwx 1 user staff myscript.sh.** ") in this

With the insertion of 777 any user can open myscript.sh and insert malicious commands

So it is dangerous to give that command.

8) **What is the differences between apropos (i.e., man -k) and whatis (i.e., man -f)?**

Ans: whatis (man -f):

This command gives a **shortcut and one-line description** of a command. It's useful when you already about the command name but just want to cheeck what it does and what are the flag can be performed .

apropos (man -k):

This command searches manual pages for all the commands related to a keyword or other things. It is only helpful when you donnot remember the exact command name but only know the function what we need.

9) **Write a command to redirect the error output of a command to a file named error.log.**

Ans: the command will be (" ls /nonexistent_directory 2> error.log ")

10) **How can you use the tee command toappend output to a file instead of overwriting it?**

Ans: command ("command_name | tee -a filename.txt ")

(command_name) which rerepresents the command whose output which want to save.

(| tee) it takes the output and writes it to a file **and also shows the it on the screen display.**

(-a) it is **append mode** so it that adds the new output at the end of the file

(filename.txt) it is command that the file where the output will be stored inmemory .