LINUX ASSIGNMENT-2

1.What are inode and process id?

ANS: **INODE:** Linux® must allocate an index node (inode) for every file and directory in the filesystem. Inodes do not store actual data. Instead, they store the metadata where you can find the storage blocks of each file's data.

PROCESS ID: In Linux, when an executable stored on disk is called a program, and a program loaded into memory and running is called a process. A process is given a unique number called process ID (PID) that identifies that process to the system, when it is started.

2. Which are the Linux Directory Commands?

ANS: Linux Directory Commands:

- pwd Command. The pwd command is used to display the location of the current working directory.
- mkdir Command. The mkdir command is used to create a new directory under any directory.
- rmdir Command. The rmdir command is used to delete a directory.
- ls Command. ...
- cd Command.

3.What is Virtual Desktop?

ANS: A virtual desktop is a computer operating system that does not run directly on the endpoint hardware from which a user accesses it.

Virtual desktops are accessible through client software installed directly on an endpoint, which presents the desktop to the user and allows them to interact with it using a keyboard, mouse, touchscreen and peripherals.

Virtual desktop clients are available for a wide variety of devices, including PCs, tablets, smartphones and Raspberry Pi.

The three major virtual desktop providers are Citrix, Microsoft and VMware. Many other vendors offer products and services to help IT professionals deploy, manage, secure and optimize virtual desktops.

4. Which are the different modes of vi editor?

ANS: The vi editor has two modes: Command and Insert. When you first open a file with vi, you are in Command mode. Command mode means that you can use keyboard keys to navigate, delete, copy, paste, and do a number of other tasks—except entering text.

5.What are daemons?

ANS: A daemon is a long-running background process that answers requests for services. The term originated with Unix, but most operating systems use daemons in some form or another. In Unix, the names of daemons conventionally end in "d". Some examples include inetd , httpd , nfsd , sshd , named , and lpd .

6.What are the process states in Linux?

ANS: While these processes exist, they'll be in one of the five possible states:

- Running or Runnable (R)
- Uninterruptible Sleep (D)
- Interruptable Sleep (S)
- Stopped (T)
- Zombie (Z)

7.Explain grep command?

ANS: Grep is an acronym that stands for Global Regular Expression Print. Grep is a Linux / Unix command-line tool used to search for a string of characters in a specified file. The text search pattern is called a regular expression. When it finds a match, it prints the line with the result.

8. Explain Process Management System Calls in Linux?

ANS: System call provides an interface between user program and operating system. The structure of system call is as follows –

When the user wants to give an instruction to the OS then it will do it through system calls. Or a user program can access the kernel which is a part of the OS through system calls.

It is a programmatic way in which a computer program requests a service from the kernel of the operating system.

Types of system calls:

The different system calls are as follows –

- System calls for Process management
- System calls for File management
- System calls for Directory management

System calls for Process management:

A system is used to create a new process or a duplicate process called a fork.

The duplicate process consists of all data in the file description and registers common. The original process is also called the parent process and the duplicate is called the child process.

The fork call returns a value, which is zero in the child and equal to the child's PID (Process Identifier) in the parent. The system calls like exit would request the services for terminating a process.

Loading of programs or changing of the original image with duplicate needs execution of exec. Pid would help to distinguish between child and parent processes.

Example:

Process management system calls in Linux.

- **fork** − For creating a duplicate process from the parent process.
- **wait** − Processes are supposed to wait for other processes to complete their work.
- **exec** − Loads the selected program into the memory.
- **exit** − Terminates the process.

fork() − A parent process always uses a fork for creating a new child process. The child process is generally called a copy of the parent. After execution of fork, both parent and child execute the same program in separate processes.

exec() − This function is used to replace the program executed by a process. The child sometimes may use exec after a fork for replacing the process memory space with a new program executable making the child execute a different program than the parent.

exit() − This function is used to terminate the process.

wait() − The parent uses a wait function to suspend execution till a child terminates. Using wait the parent can obtain the exit status of a terminated child.

9.Explain the ‘ls’ command?

ANS: The ls command is used to list files or directories in Linux and other Unix-based operating systems. Just like you navigate in your File explorer or Finder with a GUI, the ls command allows you to list all files or directories in the current directory by default, and further interact with them via the command line.

10. Explain the redirection operator?

ANS: Redirection can be defined as changing the way from where commands read input to where commands sends output. You can redirect input and output of a command. For redirection, meta characters are used.