linux Viva Question

Questions asked in viva (Linux and shell scripting)

1. What is the Linux Operating System? What is the difference between DOS and shell command?

Linux is a Unix-like operating system that is free and open-source. It is based on the Unix operating system, but it has been modified and extended over the years. Linux is a popular choice for servers, desktops, and embedded devices.
DOS (Disk Operating System) was a popular operating system for personal computers in the 1980s and 1990s. It was a single-tasking, single-user operating system that was based on a command-line interface. DOS was eventually replaced by Windows, but it is still used in some embedded devices.
The main difference between DOS and Linux is that Linux is a multitasking, multi-user operating system, while DOS is a single-tasking, single-user operating system. This means that Linux can run multiple programs at the same time, while DOS can only run one program at a time. Linux also has a graphical user interface (GUI), while DOS is a command-line operating system.

Feature	DOS	Linux
Operating system type	Single-tasking, single-user	Multitasking, multi-user
User interface	Command-line	Graphical user interface (GUI)
Popularity	Less popular	More popular
Use cases	Servers, embedded devices	Servers, desktops, embedded devices

DOS and shell command are two different things, although they are often confused with each other.

DOS is an operating system, while a shell command is a command that is used to interact with the operating system. The shell command is a program that is used to interpret and execute commands that are entered by the user.

In DOS, the shell command is called COMMAND.COM. It is the program that is responsible for displaying the command prompt and interpreting the commands that are entered by the user.

Other shells can also be used with DOS, such as the Norton Commander shell and the DOSKEY shell. These shells provide additional features and functionality over the standard COMMAND.COM shell.

Here is a table of the key differences between DOS and shell command:

Feature	DOS	Shell command
Type	Operating system	Program
Purpose	Provides a way to interact with the operating system	Executes commands that are entered by the user
Example	COMMAND.COM	DIR, CD, COPY, DEL

2. How to Check Memory Usage in Linux using command?

To check memory usage in Linux, you can use the `free` and `top` commands. Here's how you can use them:

> `free` command: The `free` command provides information about the system's memory usage, including total memory, used memory, free memory, and memory used by buffers and cache. Open a terminal and run the following command:

c

Copy code

```
free -h
```

The `-h` option is used to display memory sizes in a human-readable format.

> `top` command: The `top` command provides real-time information about system resources, including memory usage. Open a terminal and run the following command:

css

Copy code

```
top
```

By default, `top` displays the overall system status, including CPU and memory usage. Look for the "MEM" section to find memory-related information. You can also sort processes based on memory usage by pressing Shift + M.

Additionally, you can use the `htop` command, which is an interactive version of `top` and provides a more user-friendly interface with color-coded display and additional features. If `htop` is not installed on your system, you can install it using your package manager.

These commands provide you with an overview of memory usage on your Linux system. They are useful for monitoring system performance, identifying memory-intensive processes, and troubleshooting memory-related issues.

There are several commands that you can use to check memory usage in Linux. Here are some of the most common commands:

- free - This command displays the total amount of free and used physical and swap memory in the system, as well as the buffers used by the kernel.

- vmstat - This command reports information about processes, memory, paging, block IO, traps, and cpu activity.
- top - This command provides a dynamic real-time view of a running system. You can use top to see a list of processes currently being managed by the Linux kernel, as well as their memory usage.
- htop - This command is similar to top, but it provides a more graphical interface.

To use the free command, simply type `free` at the command prompt. You will see output similar to the following:

```
              total        used        free      shared    buffers
cached
Mem:          4194304    2237040    1957264       22080    248560
1601344
Swap:         4194304      0        4194304
```

The `total` line shows the total amount of memory available. The `used` line shows the amount of memory that is currently being used. The `free` line shows the amount of memory that is currently available. The `shared` line shows the amount of memory that is shared between processes. The `buffers` line shows the amount of memory that is used for buffers. The `cached` line shows the amount of memory that is used for caching.

To use the vmstat command, type `vmstat` at the command prompt. You will see output similar to the following:

```
procs -----------memory---------- ---swap-- -----io----
--------system--
 r  b   swpd   free   buff  cache   si   so    bi    bo   in
cs us sy id wa st
 1  0    768  39112  16296  132568    0    0     0     0    0
0  5  1 94  0
```

The `procs` line shows the number of processes currently running. The `memory` line shows the amount of memory that is currently being used. The `swap` line shows the amount of swap space that is currently being used. The `io` line shows the number of I/O

operations that have been performed. The `system` line shows the amount of time that the kernel has spent in various system states.

To use the top command, type `top` at the command prompt. You will see a list of processes currently being managed by the Linux kernel, as well as their memory usage. You can use the arrow keys to scroll through the list of processes. To exit top, press `q`.

To use the htop command, type `htop` at the command prompt. You will see a graphical interface that shows a list of processes currently being managed by the Linux kernel, as well as their memory usage. You can use the mouse to interact with htop. To exit htop, press `q`.

3. Explain Samba server?

Samba is a free and open-source software suite that allows you to share files, printers, and other resources between computers running Microsoft Windows and Linux. It is a popular choice for home users and businesses alike, as it is easy to set up and use.

Samba works by translating Windows network protocols into Linux ones. This allows Windows computers to see and access Linux files and printers as if they were on a Windows network. Samba also allows Linux computers to see and access Windows files and printers.

To set up a Samba server, you will need to install the Samba software on your Linux computer. Once the software is installed, you will need to configure it to share the files and printers that you want to share. The configuration process is relatively straightforward, and there are many online resources that can help you get started.

Once you have configured your Samba server, you can connect to it from a Windows computer using the \computername\sharename syntax. For example, if your Samba server is named "server" and you want to share the "myfiles" directory, you would connect to it using the \server\myfiles syntax.

Samba is a powerful tool that can be used to share files and printers between Windows and Linux computers. It is easy to set up and use, and it is a popular choice for home users and businesses alike.

Here are some of the benefits of using Samba:

- It is free and open-source software.

- It is easy to set up and use.
- It is compatible with Windows and Linux computers.
- It is a reliable and secure solution for sharing files and printers.

Here are some of the limitations of using Samba:

- It can be difficult to troubleshoot if there are problems.
- It is not as secure as some other solutions.
- It can be slow if the network is congested.

Overall, Samba is a powerful and versatile tool that can be used to share files and printers between Windows and Linux computers. It is easy to set up and use, and it is a popular choice for home users and businesses alike.

4. Explain Tc-shell

5. Access Control Lists(ACL) in Linux?

Access Control Lists (ACLs) in Linux are a way to control who has access to files and directories. They are more flexible than the traditional Unix permissions system, which only allows you to control access for the owner, group, and others.

ACLs allow you to specify permissions for individual users or groups, even if they are not the owner or group of the file or directory. This can be useful for granting access to files to users who are not members of the same group, or for granting more granular permissions than the traditional permissions system allows.

To use ACLs in Linux, you need to use the `setfacl` and `getfacl` commands. The `setfacl` command is used to set ACLs, and the `getfacl` command is used to view ACLs.

Here is an example of how to use the `setfacl` command to grant read and write permissions to the user `alice` on the file `/etc/passwd`:

```
setfacl -m u:alice:rw /etc/passwd
```

This command will add an ACL entry to the `/etc/passwd` file that grants read and write permissions to the user `alice`.

To view the ACL for a file or directory, you can use the `getfacl` command. For example, to view the ACL for the file `/etc/passwd`, you would use the following command:

```
getfacl /etc/passwd
```

This command will print out the ACL for the `/etc/passwd` file.

ACLs are a powerful tool that can be used to control access to files and directories in Linux. They are more flexible than the traditional Unix permissions system, and they can be used to grant more granular permissions.

Here are some of the benefits of using ACLs in Linux:

- They allow you to grant more granular permissions than the traditional permissions system.
- They allow you to grant permissions to users who are not members of the same group.
- They are more secure than the traditional permissions system.

Here are some of the drawbacks of using ACLs in Linux:

- They can be more complex to manage than the traditional permissions system.
- They are not supported by all Linux distributions.

Overall, ACLs are a powerful tool that can be used to control access to files and directories in Linux. They are more flexible than the traditional permissions system, and they can be used to grant more granular permissions. However, they can be more complex to manage than the traditional permissions system, and they are not supported by all Linux distributions.

6. What is C shell and Bourne shell?
7. What are shells in Linux? its types?
8. What do you mean by shell variable?and its types?

n the context of shell scripting, a shell variable is a symbol or name that represents a value stored in memory. Shell variables are used to store and manipulate data within the shell environment. They provide a way to assign values, retrieve values, and perform operations on those values within a shell script.

There are different types of shell variables based on their scope and usage:

> Environment Variables: These are variables that are available to all processes running in the shell environment. They are typically used to define system-wide settings and configurations. Environment variables are set using the `export` command, and their names are conventionally written in uppercase. Examples of environment variables include `PATH` (specifying directories to search for executable programs), `HOME` (the user's home directory), and `LANG` (the default language setting).
> Local Variables: Local variables are limited in scope to a specific shell script or function. They are created and used within the script or function and are not accessible outside of it. Local variables are useful for storing temporary data and function parameters. They are assigned values using the assignment operator (`=`) without the `export` keyword. Local variable names are conventionally written in lowercase.
> Positional Parameters: Positional parameters are special variables that hold arguments passed to a shell script or function. They are represented by numbers: `$0` represents the name of the script or function itself, `$1` represents the first argument, `$2` represents the second argument, and so on. The total number of arguments passed can be obtained using the `$#` variable.
> Special Variables: Shell environments provide a set of predefined variables that have special meanings. These variables are set and maintained by the shell itself. Examples of special variables include `$?` (the exit status of the last command), `$$` (the process ID of the current shell), and `$!` (the process ID of the last background command).

These are some of the commonly used types of shell variables. Each type has its own purpose and usage, allowing shell scripts to store and manipulate data efficiently.

9. What are the two files of the crontab command?

The crontab command is used to manage cron jobs, which are tasks that are executed on a regular schedule. There are two files associated with the crontab command:

- The crontab file: This file contains a list of cron jobs. Each cron job is a line in the file that specifies the command to be executed, the schedule for executing the command, and any environment variables that should be set.
- The crontab directory: This directory contains the crontab files for all users on the system. The crontab file for a particular user is located in a subdirectory of the crontab directory that is named after the user.

The crontab file is a plain text file that can be edited using any text editor. The syntax for the crontab file is described in the crontab man page: https://linux.die.net/man/5/crontab.

The crontab directory is typically located in the `/etc` directory. The permissions on the crontab directory are set so that only root can access it.

To view the crontab file for a particular user, you can use the `crontab -l` command. To edit the crontab file for a particular user, you can use the `crontab -e` command.

The crontab command is a powerful tool that can be used to automate tasks on a regular schedule. It is a valuable tool for system administrators and users who want to automate their tasks.

10. how to create a directory in linux.? What is the use of dir?
11. How do you compare 2 strings in a shell script?
12. How to Create Shell Script? Why do we need to use shell scripting?
13. What is an archive file in Linux?
14. What is the Korn shell?
15. How to use pipe in Linux command?
16. What are grep and find commands
17. What is NFS and how does it work?
18. Parameter and the variable?
19. What is a task in Linux?
20. Control Statements in Shell Programming?
21. How do you get the sequence of numbers in Linux?

There are a few ways to get the sequence of numbers in Linux.

One way is to use the `seq` command. The `seq` command generates a sequence of numbers, starting from a specified number and ending at another specified number. The syntax for the `seq` command is:

```
seq [options] start end
```

For example, to generate a sequence of numbers from 1 to 10, you would type the following command:

```
seq 1 10
```

This would output the following sequence of numbers:

```
1
2
3
4
5
6
7
8
9
10
```

Another way to get the sequence of numbers in Linux is to use the `while` loop. The `while` loop is a control flow statement that repeats a block of code as long as a specified condition is true. The syntax for the `while` loop is:

```
while [ condition ]; do
    commands
done
```

For example, to generate a sequence of numbers from 1 to 10, you would use the following code:

```
i=1
while [ $i -le 10 ]; do
    echo $i
    i=$((i+1))
done
```

This code would first initialize the variable `i` to 1. Then, it would use the `while` loop to repeat the block of code as long as the value of `i` is less than or equal to 10. The block of code would simply echo the value of `i`. Finally, the code would increment the value of `i` by 1 and repeat the loop.

The `seq` command and the `while` loop are two ways to get the sequence of numbers in Linux. The `seq` command is a more concise way to generate a sequence of numbers, but the `while` loop gives you more flexibility.

22. Which special characters are used in Shell Programming?
23. What is the builtin command in Linux? and its use?

-> What is the mount point?

A mount point is a directory on a file system that is logically linked to another file system. When a partition is mounted, it is made accessible to the operating system and can be used to store files.

The mount point is a critical part of the mounting process, as it allows the operating system to know where to find the mounted partition. The mount point is also used to determine the permissions that users have to the mounted partition.

The mount point is specified in the /etc/fstab file. This file contains a list of all the partitions that are mounted on the system, as well as the mount points for each partition. The mount point is listed in the second field of the /etc/fstab file.

For example, the following line in the /etc/fstab file mounts the /dev/sda1 partition on the /mnt/disk1 mount point:

```
/dev/sda1 /mnt/disk1 ext4 defaults 0 0
```

The mount point is a very important part of the mounting process, and it should be specified carefully. If the mount point is incorrect, the partition may not be mounted correctly, or it may not be accessible to users.

Here are some of the commands that you can use to manage mount points in Linux:

- `mount`: This command is used to mount a partition.
- `umount`: This command is used to unmount a partition.
- `df`: This command displays a list of all the mounted partitions.

I hope this helps! Let me know if you have any other questions.

30 viva questions most commonly asked in viva

https://www.tecmint.com/linux-interview-questions/

60 viva questions about commands most commonly asked in viva
https://www.youtube.com/watch?v=vo9F_I_wwZs&list=PL0tP8lerTbX2z5RS4RhOjknFjSRdcntjq

File system

https://www.youtube.com/watch?v=RCUeSps59rM

60 viva questions about commands most commonly asked in viva

https://www.guru99.com/linux-interview-questions-answers.html