Optional

Interview Questions

1) According to you, why is there no antivirus software for Linux?

Ans: The main reason why we do not need an antivirus in Linux is that very little malware exists in Linux as compared with other operating systems such as Windows. This is because by default, for any malicious program or file to run on the system, admin privileges are required, which are not available for all the users or to a malicious software on its own, for that matter.

In Linux systems, there is a clear distinction between user accounts and administrator accounts. Users cannot install system-wide programs, and they do not have access to the important system folders. Suppose you downloaded and ran a virus or any malware on Linux. In that case, it would mess up your user account but could not spread and infect the entire system unless you gave it administrative privileges. Viruses exploit bugs in the system, and since Linux is open sourced, it has very few bugs. If any bug is detected, it gets fixed fast as compared with other operating systems.

Live Help

Ans: Yes, we can find out how much memory Linux is using. It can be done using the ”cat” command by applying cat/proc/meminfo in the command shell. This will display the memory usage of Linux. The outcome is in the form of Mem:2204537 etc. This outcome is the memory that Linux has to offer you as available for usage.

3) How do you change permissions in Linux, assuming that you are the system administrator or the owner of a file/directory?

Ans: We can grant permission using the chmod command. We use the ‘+’ symbol to add permission and the ‘–’ symbol to deny permission, along with any of the following letters:

u (user),

g (group),

o (others),

a (all),

r (read),

w (write) and

x (execute).

For example, the command chmod go+rw FILE1.TXT grants read and write access to the file FILE1.TXT, which is assigned to groups and others.

4) What could be the problem when a command that was issued gave a different result from the last time it was used?

For example, to list all the files in a directory, you should type the command ls, and not LS. Typing LS will either result in an error message if no program by that exact name exists or may produce a different output if there is a program named LS that performs another function.

5) What is the use of '#!/bin/bash' while writing a script?

Ans: #!/bin/bash is the first line of a shell script and is known as shebang, where the symbol '#' is called a hash and '!' is called a bang. As it begins with the '#' symbol, it is read as a comment. #!/bin/bash specifies that the current file is a bash script, which will be interpreted with the bash or an interpreter /bin/bash.

In simple words, the shebang at the head of the script tells the system that this file is a set of commands to be fed to the command interpreter indicated, which is /bin/bash in this case.

6) What is the use of the '$?' sign in shell script?

Ans: While writing a script, if you want to check whether the previous command was executed successfully or not, you can use the '$?' sign with an if statement or an echo command; this will give you the exit status of the previous command.

```
root@localhost:~# ls /lsr/bin/share
/lsr/bin/share
root@localhost:~# echo $?
```

7) Is BASH a weakly typed language? Why?

Ans: Yes, bash is considered a weakly or loosely typed language because it does not require the type of data to be declared at the time of variable declaration. All bash variables are treated as a string by default and the type of the variable will be set based on the current value. Bash variables with data types can be defined using the declare command with a particular option; for example, we can declare an integer variable using the following:-

```
declare -i number=67
```

But the options to define data types are limited, and the declare command does not support all types of data. For example, float data type cannot be declared using the declare command.

8) How can you run multiple bash scripts in parallel?

Ans: Multiple bash scripts can be executed in parallel using the nohup command. We can execute multiple bash files from a folder in parallel using the script given below.-

done

9) How do you get a part of a string variable using the echo command only ?

Ans: We can use parameter expansion along with the echo command to get a desired part of the string variable. This can be done in the following way:-

```
echo ${variable:x:y}
```

x - start position

y - length

Three parts can be defined in parameter expansion, by separating by a colon to cut any part of the string data. Here, the first two parts are mandatory, and the last part is optional. The first part contains the name of the string variable that is to be cut, the second part is the starting position from where the string has to be cut, and the third part is the length of the cutting string.

Note that the starting position is counted from 0, and the length is counted from 1 to retrieve the cutting value.

Example:-

```
variable=”My name is XYZ, I am from ABC”
echo ${variable:3:7}
```

This will display ‘name is’ as the output.

configuration files that hold important data or set-up information. Setting these files as hidden makes them less likely to be accidentally deleted.

Report an error

FINISH SESSION
Interview Questions