

1: What is a shell?

Shell is an interface between the user and the kernel. Even though there can be only one kernel; a system can have many shell running simultaneously. So, whenever a user enters a command through the keyboard, the shell communicates with the kernel to execute it and then display the output to the user.

2: What are the different types of commonly used shells on a typical Linux system?

csk,ksh,bash,Bourne . The most commonly used and advanced shell used today is "Bash" .

3: What is the equivalent of a file shortcut that we have a window on a Linux system?

Shortcuts are created using "links" on Linux. There are two types of links that can be used namely "soft link" and "hard link".

4: What is the difference between soft and hard links?

Soft links are link to the file name and can reside on different filesystem as well; however hard links are link to the inode of the file and have to be on the same filesystem as that of the file. Deleting the original file makes the soft link inactive (broken link) but does not affect the hard link (Hard link will still access a copy of the file)

5: How will you pass and access arguments to a script in Linux?

Arguments can be passed as:

scriptName "Arg1" "Arg2""Argn" and can be accessed inside the script as \$1 , \$2 .. \$n

6: What is the significance of \$#?

\$# shows the count of the arguments passed to the script.

7: What is the difference between \$* and \$@?

\$@ treats each quoted arguments as separate arguments but \$* will consider the entire set of positional parameters as a single string.

8: Use sed command to replace the content of the file (emulate tac command)

Eg:

if cat fille

ABCD

EFGH

Then O/p should be

EFGH

ABCD

```
sed '1! G; h;$!d' file1
```

Here G command appends to the pattern space,

h command copies pattern buffer to hold buffer

and d command deletes the current pattern space.

9: Given a file, replace all occurrence of word “ABC” with “DEF” from 5th line till end in only those lines that contains word “MNO”

```
sed -n '5,$p' file1|sed '/MNO/s/ABC/DEF/'
```

10: Given a file, write a command sequence to find the count of each word.

```
tr -s “(backslash)040” <file1|tr -s “(backslash)011”|tr “(backslash)040 (backslash)011”  
“(backslash)012” |uniq -c
```

where “(backslash)040” is octal equivalent of “space”

“(backslash)011” is an octal equivalent of “tab character” and

“(backslash)012” is an octal equivalent of the newline character.

11: How will you find the 99th line of a file using only tail and head command?

```
tail +99 file1|head -1
```

12: Print the 10th line without using tail and head command.

```
sed -n '10p' file1
```

13: In my bash shell I want my prompt to be of format '\$'Present working directory':hostname'> and load a file containing a list of user-defined functions as soon as I log in, how will you automate this?

In bash shell, we can create ".profile" file which automatically gets invoked as soon as I log in and write the following syntax into it.

```
export PS1='$ `pwd`:hostname`>' .File1
```

Here File1 is the file containing the user-defined functions and "." invokes this file in current shell.

14: Explain about "s" permission bit in a file?

"s" bit is called "set user id" (SUID) bit.

"s" bit on a file causes the process to have the privileges of the owner of the file during the instance of the program.

For example, executing "passwd" command to change current password causes the user to writes its new password to shadow file even though it has "root" as its owner.

15: I want to create a directory such that anyone in the group can create a file and access any person's file in it but none should be able to delete a file other than the one created by himself.

We can create the directory giving read and execute access to everyone in the group and setting its sticky bit "t" on as follows:

```
mkdir direc1
```

```
chmod g+wx direc1
```

```
chmod +t direc1
```

16: How can you find out how long the system has been running?

We can find this by using the command "uptime".

17: How can any user find out all information about a specific user like his default shell, real-life name, default directory, when and how long he has been using the system?

finger "loginName" ...where loginName is the login name of the

user whose information is expected.

18: What is the difference between \$\$ and \$!?

\$\$ gives the process id of the currently executing process whereas \$! Shows the process id of the process that recently went into the background.

19: What are zombie processes?

These are the processes which have died but whose exit status is still not picked by the parent process. These processes even if not functional still have its process id entry in the process table.

20: How will you copy a file from one machine to other?

We can use utilities like "ftp," "scp" or "rsync" to copy a file from one machine to other.

E.g., Using ftp:

FTP hostname

>put file1

>bye

Above copies, file file1 from the local system to destination system whose hostname is specified.

21: I want to monitor a continuously updating log file, what command can be used to most efficiently achieve this?

We can use tail -f filename. This will cause only the default last 10 lines to be displayed on std o/p which continuously shows the updating part of the file.

22: I want to connect to a remote server and execute some commands, how can I achieve this?

We can use ssh to do this:

ssh username@serverIP -p sshport

Example

ssh root@122.52.251.171 -p 22

Once above command is executed, you will be asked to enter the password

23: I have 2 files and I want to print the records which are common to both.

We can use "comm" command as follows:

comm -12 file1 file2 ... 12 will suppress the content which are

unique to 1st and 2nd file respectively.

24: Write a script to print the first 10 elements of Fibonacci series.

```
#!/bin/sh

a=1

b=1

echo $a

echo $b

for I in 1 2 3 4 5 6 7 8

do

c=a

b=$a

b=$((a+c))

echo $b

done
```

25: How will you connect to a database server from Linux?

We can use isql utility that comes with open client driver as follows:

```
isql -S serverName -U username -P password
```

26: What are the 3 standard streams in Linux?

- 0 - Standard Input
- 1 - Standard Output
- 2 - Standard Error

27: I want to read all input to the command from file1 direct all output to file2 and error to file 3, how can I achieve this?

```
command <file1 1>file2 2>file3
```

28: What will happen to my current process when I execute a command using exec?

"exec" overlays the newly forked process on the current process; so when I execute the command using exec, the command gets executed on the current shell without creating any new processes.

E.g., Executing **"exec ls"** on command prompt will execute ls and once ls exits, the process will shut down

29: How will you emulate wc -l using awk?

```
awk 'END {print NR}' fileName'
```

30: Given a file find the count of lines containing the word "ABC".

```
grep -c "ABC" file1
```

31: What is the difference between grep and egrep?

egrep is Extended grep that supports added grep features like "+" (1 or more occurrence of a previous character), "?" (0 or 1 occurrence of a previous character) and "|" (alternate matching)

32: How will you print the login names of all users on a system?

/etc/shadow file has all the users listed.

```
awk -F ':' '{print $1}' /etc/shadow | uniq -u
```

33: How to set an array in Linux?

Syntax in ksh:

```
Set -A arrayname= (element1 element2 ..... element)
```

In bash

```
A=(element1 element2 element3 .... elementn)
```

34: Write down the syntax of "for" loop

Syntax:

```
for iterator in (elements)
```

```
do
```

```
execute commands
```

```
done
```

35: How will you find the total disk space used by a specific user?

`du -s /home/user1`where user1 is the user for whom the total disk space needs to be found.

36: Write the syntax for “if” conditionals in Linux?

Syntax

If condition is successful

then

execute commands

else

execute commands

fi

37: What is the significance of \$?

The command `$?` gives the exit status of the last command that was executed.

38: How do we delete all blank lines in a file?

`sed '^ [(backslash)011(backslash)040]*$/d' file1`

where `(backslash)011` is an octal equivalent of space and

`(backslash)040` is an octal equivalent of the tab

39: How will I insert a line “ABCDEF” at every 100th line of a file?

`sed '100i\ABCDEF' file1`

40: Write a command sequence to find all the files modified in less than 2 days and print the record count of each.

`find . -mtime -2 -exec wc -l {} \;`

41: How can I set the default rwx permission to all users on every file which is created in the current shell?

We can use:

umask 777

This will set default rwx permission for every file which is created for every user.

42: How can we find the process name from its process id?

We can use “ps -p ProcessId”

43: What are the four fundamental components of every file system on Linux?

Bootblock, super block, inode block and Datablock are found fundamental components of every file system on Linux.

44: What is a boot block?

This block contains a small program called “Master Boot record”(MBR) which loads the kernel during system boot up.

45: What is a super block?

Super block contains all the information about the file system like the size of file system, block size used by its number of free data blocks and list of free inodes and data blocks.

46: What is an inode block?

This block contains the inode for every file of the file system along with all the file attributes except its name.

47: How can I send a mail with a compressed file as an attachment?

```
zip file1.zip file1 | mailx -s "subject" Recipients email id
```

Email content

EOF

48: How do we create command aliases in a shell?

alias Aliasname="Command whose alias is to be created".

49: What are “c” and “b” permission fields of a file?

“c” and “b” permission fields are generally associated with a device file. It specifies whether a file is a special character file or a block special file.

50: What is the use of a shebang line?

Shebang line at the top of each script determines the location of the engine which is to be used to execute the script.

[Guru99](#) Provides [FREE ONLINE TUTORIAL](#) on Various courses like

Java	MIS	MongoDB	BigData	Cassandra
Web Services	SQLite	JSP	Informatica	Accounting
SAP Training	Python	Excel	ASP Net	HBase
Project Management	Test Management	Business Analyst	Ethical Hacking	PMP
Live Project	SoapUI	Photoshop	Manual Testing	Mobile Testing
Data Warehouse	R Tutorial	Tableau	DevOps	AWS
Jenkins	Agile Testing	RPA	JUnit	Software Engineering
Selenium	CCNA	AngularJS	NodeJS	PLSQL

Stay updated with new
courses at Guru99
Join our Newsletter