

Shell scripting or programming mostly consists of the features which today's modern programming languages offer.

Right from a simple to the complex script can be developed using Shell Scripting. Shell scripting is nothing but series of UNIX commands written in a plain text file to accomplish a specific task. And with the help of shell scripting, some tasks of the day to day life can be automated.

There are hardly few documents available over the internet on shell scripting interview questions and answers. Hence, I have chosen Shell Scripting as my topic to help those who are in need of it.

1. What is Shell?

Ans: Shell is a command interpreter, which interprets the command which the user gives to the kernel. It can also be defined as an interface between a user and operating system.

2. What is Shell Scripting?

Ans: Shell scripting is nothing but series or sequence of UNIX commands written in a plain text file. Instead of specifying one job/command at a time, in shell scripting we give a list of UNIX commands like a to-do list in a file to execute it.

3. What is the Importance of writing Shell Scripts?

Ans: The points given below explain the importance of writing shell scripts.

- Shell script takes input from the user, file and displays it on the screen.
- Shell scripting is very useful in creating your own commands.
- It is helpful in automating some tasks of the day to day life.
- It is useful for automating system administration tasks.
- Mainly it saves time.

4. What are the different Types of Shells available?

Ans: There are mainly 4 important types of shells that are widely used.

And they include:

- Bourne Shell (sh)
- C Shell (csh)
- Korn Shell (ksh)
- Bourne Again Shell (bash)

5. What are the Advantages of C Shell over Bourne Shell?

Ans: The advantages of C Shell over Bourne Shell are:

- C shell allows aliasing of commands i.e. a user can give any name of his choice to the command. This feature is mainly useful when a user has to type the lengthy command again and again. At that point of time, instead of typing a lengthy command a user can type the name that he has given.
- C shell provides command history feature. C shell remembers the previously typed command. Thus, it avoids typing the command again and again.

6. When should shell programming/scripting not be used?

Ans: Generally, shell programming/scripting should not be used in the below instances.

- When the task is very much complex like writing the entire payroll processing system.
- Where there is a high degree of productivity required.
- When it needs or involves different software tools.

7. What are the default permissions of a file when it is created?

Ans: 666 i.e. rw-rw-rw- is the default permission of a file when it is created.

8. What are the two types of Shell Variables? Explain in brief.

Ans: The two types of shell variables are:

#1) Unix Defined Variables or System Variables – These are standard or shell defined variables. Generally, they are defined in CAPITAL letters.

Example: SHELL – This is a Unix Defined or System Variable, which defines the name of the default working shell.

#2) User Defined Variables – These are defined by users. Generally, they are defined in lower letters

Example: \$ a=10 –Here the user has defined a variable called 'a' and assigned value to it as 10.

9. What are positional parameters? Explain with an example.

Ans: Positional parameters are the variables defined by a shell. And they are used whenever we need to convey information to the program. And this can be done by specifying arguments at the command line.

There are totally 9 positional parameters present i.e. from \$1 to \$9.

Example: \$ Test Indian IT Industry has grown very much faster

In the above statement, positional parameters are assigned like this.

\$0 -> Test (Name of a shell program/script)

\$1 -> Indian

\$2 -> IT and so on.

10. What are the different blocks of a file system? Explain in brief.

Ans: Given below are the main 4 different blocks available on a file system.

Super Block: This block mainly tells about a state of the file system like how big it is, maximum how many files can be accommodated etc.

Boot Block: This represents the beginning of a file system. It contains bootstrap loader program, which gets executed when we boot the host machine.

Inode Table: As we know all the entities in a UNIX are treated as files. So, the information related to these files are stored in an Inode table.

Data Block: This block contains the actual file contents.

11. **What are the three modes of operation of vi editor? Explain in brief.**

Ans: The three modes of operation of **vi editors** are,

(i) Command Mode: In this mode, all the keys pressed by a user are interpreted as editor commands.

(ii) Insert Mode: This mode allows for insertion of a new text and editing of an existing text etc.

(iii) The ex-command Mode: This mode allows a user to enter the commands at a command line.

12. **What are control instructions and how many types of control instructions are available in a shell? Explain in brief.**

Ans: Control Instructions are the ones, which enable us to specify the order in which the various instructions in a program/script are to be executed by the computer. Basically, they determine a flow of control in a program.

There are 4 types of control instructions that are available in a shell.

- **Sequence Control Instruction** – This ensures that the instructions are executed in the same order in which they appear in the program.
- **Selection or Decision Control Instruction** – It allows the computer to take a decision as to which instruction is to be executed next.
- **Repetition or Loop Control Instruction** – It helps a computer to execute a group of statements repeatedly.

- **Case-Control Instruction** – This is used when we need to select from several alternatives.

13. **What are Loops and explain three different methods of loops in brief?**

Ans: Loops are the ones, which involve repeating some portion of the program/script either a specified number of times or until a particular condition is being satisfied.

3 methods of loops are:

- **For** loop – This is the most commonly used loop. For loop allows specifying a list of values which the control variable in the loop can take. The loop is then executed for each value mentioned in the list.
- **While** loop – This is used in a program when we want to do something for a fixed number of times. While loop gets executed till it returns a zero value.
- **Until** loop – This is similar to while loop except that the loop executes until the condition is true. Until loop gets executed at least once till it returns a non-zero value.

14. **What is IFS?**

Ans: IFS stands for **Internal Field Separator**. And it is one of the system variables. By default, its value is space, tab, and a new line.

It signifies that in a line where one field or word ends and another begins.

15. **What is a Break statement and what is it used for?**

Ans: The break is a keyword and is used whenever we want to jump out of a loop instantly without waiting to get back to the control command.

When the keyword break is encountered inside any loop in the program, control will get passed automatically to the first statement after a loop. A break is generally associated with an if.

16. **What is Continue statement and what is it used for?**

Ans: Continue is a keyword and is used whenever we want to take the control to the beginning of the loop, by passing the statements inside the loop which have not yet been executed.

When the keyword continue is encountered inside any loop in the program, control automatically passes to the beginning of a loop. Continue is generally associated with an if.

17. What is Shebang in a shell script?

Ans: Shebang is a # sign followed by an exclamation i.e. !. Generally, this can be seen at the beginning or top of the script/program. Usually, a developer uses this to avoid repetitive work. Shebang mainly determines the location of the engine which is to be used in order to execute the script.

Here '#' symbol is called as hash and '!' is called a bang.

Example: #!/bin/bash

The above line also tells which shell to use.

18. How to debug the problems encountered in shell script/program?

Ans: Though generally it depends on the type of problem encountered. Given below are some common methods used to debug the problems in the script.

Debug statements can be inserted in the shell script to output/display the information which helps to identify the problem.

Using "set -x" we can enable debugging in the script.

19. What is the difference between = and ==?

Ans:

= using for assigning value to the variable.

== using for string comparison.

20. What is the difference between diff and cmp commands?

Ans: diff – Basically, it tells about the changes which need to be made to make files identical.

cmp – Basically it compares two files byte by byte and displays the very first mismatch.

21. Explain in brief about awk command with an example.

Ans: awk is a data manipulation utility or command. Hence, it is used for data manipulation.

Syntax: awk options File Name

Example:

Script/Code

cat > awkscript

```
Hello Kali
```

```
Hello Ubuntu
```

```
Hello Redhat
```

```
Hello Cent
```

```
Hello FoxuTech
```

awk utility/command assigns variables like this.

\$0 -> For whole line (e.g. Hello kali)

\$1 -> For the first field i.e. Hello

\$2 -> For the second field

Execution over Shell Interpreter/Editor

awk '{print \$0}' awkscript

The above script prints all the 5 lines completely.

Output:

```
Hello Kali
Hello Ubuntu
Hello Redhat
Hello Cent
Hello FoxuTech
```

Execution over Shell Interpreter/Editor

awk '{print \$1}' awkscript

The above script prints only first word i.e. Hello from each line.

Output:

```
Hello
Hello
Hello
Hello
Hello
```

22. **Explain in brief about sed command with an example.**

Ans: sed stands for **stream editor**. And it is used for editing a file without using an editor. It is used to edit a given stream i.e. a file or input from a pipeline.

Syntax: sed options file

Example: Execution over Shell Interpreter/Editor


```
# echo "Hello World" | sed 's/Hello/Hi/'
```

Here **'s'** command present in **sed** will replace string **Hello** with **Hi**.

Output:

```
Hi World
```

www.wezva.com

facebook

<https://www.facebook.com/wezva>

Linked in

<https://www.linkedin.com/in/wezva>



+91-9739110917

+91-9886328782