



UNIX Shell Scripting Interview Questions and Answers

Introduction

Shell scripting is essential for maintaining system configurations, automating processes, and increasing productivity. Professionals with experience writing Unix shell scripts are in high demand in the workforce. The top 20 Unix Shell scripting interview questions and answers are here to help you succeed. Develop your scripting skills with our [UNIX Shell Scripting course in Chennai](#).

List of UNIX Shell Interview Questions for Freshers

1. What is a shell script?
2. How can one write a shell script?
3. What does a shell scripting superblock mean?
4. List the frequently used shells on an average Linux system.
5. What does a shell scripting case statement serve to accomplish?

6. Explain the Shell variable.
7. Why are \$\$ and \$! different from one another?
8. What is a kernel?
9. In a shell script, how may a loop be created?
10. Mention a few of shell scripting's drawbacks.
11. Define the shebang line.
12. How many different kernels and shells are available in a typical UNIX environment?
13. With the UNIX operating system, how many shell scripts are included?
14. When is it inappropriate to use shell programming or scripting?
15. How can variables be set to be unchangeable?

Explore our [UNIX Shell Scripting course syllabus](#) and get started.

UNIX Shell Interview Questions and Answers for Freshers

1. What is a shell script?

A shell script is a file containing instructions that a shell interpreter translates and executes. It makes command execution and task automation possible on an operating system built on the Unix base.

2. How can one write a shell script?

A plain text file with the extension must be created. To create a shell script, type sh. You can write the script using any text editor, like Vi, Nano, or Cat.

3. What does a shell scripting superblock mean?

A superblock is software that holds the records of particular file systems. Size, counts of filled and empty blocks, use data, block group sizes, and the location and size of inode tables are among the attributes that are available in the block.

4. List the frequently used shells on an average Linux system.

The C-shell and the Bourne shell are the two primary types of Linux shells. These are their derivatives:

- C-shell: TENEX Z-Shell and C-Shell

- Bourne shells include Korn, POSIX, and Bourne-Again.

5. What does a shell scripting case statement serve to accomplish?

In shell scripting, the case statement is used to conduct conditional branching based on pattern matching in an organized and effective manner. Enabling the script to run distinct code blocks for particular pattern matches improves code readability and streamlines decision-making.

6. Explain the Shell variable.

A shell script's or program's central component is a shell variable. The shell software can handle and modify stored data thanks to the variable. In most cases, a string variable is used to hold it.

7. Why are \$\$ and \$! different from one another?

Special variables in Unix-like operating systems are \$\$ and \$!

- **\$\$:** It displays the shell or script's process ID (PID) that is present as '*script.d*'. When you wish to prevent naming conflicts by creating temporary files with distinct names.
- **\$!:** It displays the PID of the most recent background process that used the & operator to run in the background. It monitors background operations.

**Take your Knowledge
Test Report**

Check your Score



8. What is a kernel?

An operating system's kernel, a basic computer program, is in charge of managing and directing how the computer and its hardware operate.

9. In a shell script, how may a loop be created?

Loops like the “for” and “while” loops can be used in shell scripts. While the “while” loop is used to repeat commands as long as a specific condition is met, the “for” loop is used to iterate over a sequence of values.

10. Mention a few of shell scripting’s drawbacks.

The following are the drawbacks of shell scripting:

- Errors are expensive and common; one mistake can change the command.
- There is a slow pace of execution.
- The syntax or implementation of the language contains errors or shortcomings.
- It is not well-suited for large, complicated projects.
- Unlike other programming languages, it offers a simple data structure.

11. Define the shebang line.

In shell scripting, the first line of a script that starts with #! and ends with the path to the interpreter, such as /bin/bash, is referred to as the “shebang line.” It designates the interpreter that will be applied while running the script. This line of code is essential because it tells the system how to read and execute the script, ensuring compatibility and correct operation.

12. How many different kernels and shells are available in a typical UNIX environment?

There is just one kernel and numerous shells available in a typical UNIX setup.

13. With the UNIX operating system, how many shell scripts are included?

About 280 shell scripts are included with the UNIX operating system.

14. When is it inappropriate to use shell programming or scripting?

In general, the following situations shouldn’t involve the use of shell programming or scripting.

- When the work is complicated, such as when the complete payroll processing system is written.
- Where a high level of productivity is necessary.
- When several software tools are required or involved.

15. How can variables be set to be unchangeable?

Read-only can be used to make variables unchangeable. For example, read-only allows us to set the value of the variable “a” to 10 and prevent it from changing.

```
$ a=10
```

```
$ readonly a
```

Read our [Unix Shell Scripting Tutorial for Beginners](#) and learn from scratch.

List of UNIX Shell Interview Questions for Experienced

1. Positional parameters: what are they? Give an example.
2. Which three distinct security features does UNIX offer for a file or set of data?
3. Which three ways can you use the vi editor? Give a brief explanation.
4. How many different kinds of control instructions are there in a shell, and what are control instructions? Give a brief explanation.
5. Explain Metacharacters with an example.
6. What is the purpose of the continue statement?
7. What is the purpose of a break statement?
8. Which two files does the crontab command reside in?
9. Which command is required to take a backup?
10. What commands are available to verify the amount of disk space used?

UNIX Shell Technical Interview Questions and Answers for Experienced

1. Positional parameters: what are they? Give an example.

In response, a shell defines the variables known as positional parameters. Additionally, we use them anytime we need to provide the program with information. Moreover, parameters can be specified at the command line to accomplish this. There are 9 positional parameters in all, ranging from \$1 to \$9.

Example:

\$ Test The Indian IT sector has expanded far more quickly.

Positional parameters are assigned in the manner described above.

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

\$1 -> Indian

\$2 -> IT and so on.

2. Which three distinct security features does UNIX offer for a file or set of data?

UNIX has three distinct security features for a file or set of data, which are:

- It gives the user a special user ID and password, preventing unknown or unauthorized individuals from accessing it.
- By granting read, write, and execute rights for file access, it offers security at the file level.
- Finally, it uses file encryption to offer security. Using this method, a file can be encoded in an unreadable format. Even if someone manages to access a file, they won't be able to view its contents until it has been decrypted.

3. Which three ways can you use the vi editor? Give a brief explanation.

The three ways that vi editors can function are:

- **Command Mode:** In this mode, all keys pressed by the user are converted into editor commands.
- **Insert Mode:** With this mode, you can add new text, alter current text, and more.

- **Ex-command mode:** The command line can be used by a user to enter commands in ex-command mode.

4. How many different kinds of control instructions are there in a shell, and what are control instructions? Give a brief explanation.

To put it another way, control instructions allow us to tell the computer in what order to execute the different instructions in a script or program. In essence, they establish a program's control flow. In a Shell script, four different kinds of control instructions are accessible.

- **Sequence Control Instruction:** An instruction known as a sequence control instruction makes sure that the commands are carried out in the same order as they appear in the program.
- **Selection or Decision Control Instruction:** The computer can decide which instruction to run next with the help of a selection or decision control command.
- **Loop control instructions,** often known as repetition, assist a computer in repeatedly executing a set of statements.
- **Case-Control Instruction:** This is applied when there are multiple options to choose from.

5. Explain Metacharacters with an example.

In a program or data field, metacharacters are unique characters that offer details about other characters. Another name for them is shell regular expressions.

Example:

ls s – All of the files that start with the letter “s” are listed.*

Using Shell Interpreter/Editor for execution

```
/u/user1/Shell_Script2024> ls s*
```

Script for Meta Characters

Output:

```
script1 script2
```

\$ cat script1 > script2 – The output of script1 or the cat command will now move to script2.

Practice with our [Unix Shell Scripting project ideas](#).

6. What is the purpose of the continue statement?

The keyword “continue” is used to move the cursor to the beginning of the loop and pass the statements inside that haven’t been performed yet.

Control immediately shifts to the start of any loop in the program when the keyword Continue is encountered. Continue is typically connected to an if statement.

**Take your Knowledge
Test Report**

Check your Score



7. What is the purpose of a break statement?

To exit a loop immediately and without having to wait to return to the control command, we utilize the break keyword.

Any time the program’s loops contain the keyword break, control is automatically transferred to the first statement that follows the loop. An if is typically connected to a break.

8. Which two files does the crontab command reside in?

There are two crontab command files:

The ***cron.allow*** function determines which users must be able to use the crontab command.

The ***cron.deny*** function determines which users should not be allowed to use the crontab command.

9. Which command is required to take a backup?

The command that must be used to take a backup is tar. Tape Archive is what it stands for. The primary purpose of the tar command is to transfer files to and from an archive medium, such as tape.

10. What commands are available to verify the amount of disk space used?

To check the disk usage, use one of the three available commands. They are as follows:

- **df:** To check the amount of free disk space, use the *df command*.
- **du:** This command is used to view the disk utilization by directory.
- **dfspace:** Use the command “dfspace” to see how much free disk space there is in megabytes.

Conclusion

Whether you are a fresher or a working professional, we hope these Unix Shell scripting interview questions and answers will be helpful to ace your interviews. Thrive in the industry through our best [Software Training Institute in Chennai](#).