


[Home](#) > [Blog](#) > [Unix Shell Scripting](#) >

# Unix Shell Scripting Interview Questions

Rating: 4.7 ★★★★★

 92221[GET TRAINED AND CERTIFIED](#)

If you're looking for Unix Shell Scripting Interview Questions for Experienced or Freshers, you are in the right place. There are a lot of opportunities from many reputed companies in the world. **According to research Unix Shell Scripting has a market share of about 17%.**

So, You still have the opportunity to move ahead in your career in Unix Shell Scripting. Mindmajix offers Advanced **Unix Shell Scripting Interview Questions** 2022 that helps you in cracking your interview & acquire a dream career as Unix Shell Scripting Developer.

*If you want to enrich your career and become a **professional in Unix Shell Scripting**, then enroll in "[Unix Shell Scripting Training](#)" - This course will help you to achieve excellence in this domain.*

## Top 10 Frequently Asked Unix Shell Scripting Interview Questions

1. [Compare UNIX Shell Scripting with others](#)
2. [Can you tell me something about the Zombie Processes in shell scripting?](#)
3. [In Shell scripting, how will put separate you the grep and egrep?](#)
4. [How you will create a shortcut in Linux?](#)
5. [Tell something about the Super Block in Shell scripting?](#)
6. [Is it possible to pass arguments to the scripts when the same is running in the UNIX?](#)
7. [Can you name the important standard streams in the UNIX shell scripting?](#)



9. What is shell scripting.

## 10. What is Unix shell scripting?

**ps -f command in UNIX**

Column Names of a result obtained by ps -f command in Unix	
UID	User ID
PID	The ID of the process
PPID	The process ID of Parent
C	Process's CPU Utilization
STIME	Start Time of the Process
TTY	Process Terminal Type
TIME	Process CPU Time lapsed
CMD	To start the process

# Unix Shell Scripting Interview Questions

## 1) Compare UNIX Shell Scripting with others

UNIX	Other approaches
The overall shell scripts are vast (around 280)	Other approaches cannot handle such a vast term
There is no need to write a complete payroll system	There is often a need for the same
Dependency on other software tools is less	They need additional help in most the cases

## 2) Can you tell me something about the Zombie Processes in shell scripting?

These are generally defined as the scripts that have completed their life span but are yet to be picked by parent processes associated with them. In the process table, the users can locate the process id of the same despite it remains non-functional.



### 3) In Shell scripting, how will put separate you the grep and egrep?

Grep can easily be extended and the same can then be called egrep. In other words, the egrep is an advanced version of grip. There are some added features in it and i.e. it can easily be considered for the additional occurrence of a previous character. This can also be considered when it comes to alternate matching.

### 4) How you will create a shortcut in Linux?

This can be done with the help of links present in the UNIX. For this, basically, there are two links that are considered often. They are generally categorized as

- Soft Link
- Hard Link

### 5) Tell something about the Super Block in Shell scripting?

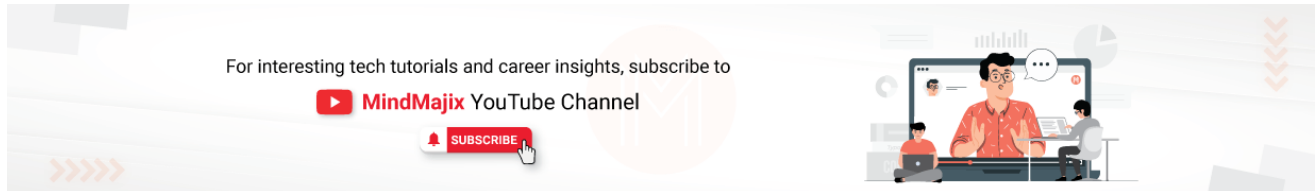
It is basically a program that contains all the information regarding a specific file system. It reflects the block size that is used by its associated number, the size of the system in terms of data handling and programming.

It also provides information regarding the free inodes and the data blocks which are currently associated with the system.

**Majix**

the UNIX.

Yes, it is possible and this practice is followed often. It is done to make sure that the system doesn't have any errors while executing the commands and the scripts are running smoothly. There are other reasons as well for which it can be executed.



## 7) Can you name the important standard streams in the UNIX shell scripting?

These are Standard Input, Standard Output as well as Standard Error.

## 8) In shell scripting, what is the significance of the Shebang line?

It simply provides information regarding the location where the engine is placed. The engine is the one that executes the script. The Shebang line is present at the top of the script and it can be neglected by the users if they want the same.

## 9) In UNIX, what is the primary fundamental component of a file system?

These are

- Boot block
- Inode Block
- Data Block
- Super Block

They remain present with all the file systems in UNIX and can be clubbed together in some special cases when the need for the same is felt to perform a specific task.

## 10) How can you put separate soft and hard links in shell scripting?

Well, [shell scripting](#) is a powerful approach. The links are used when it comes to creating shortcuts just like Windows. Soft links are those which generally related to the file name and don't have any specific location. They can be anywhere on the file system.

On the other side, the hard links are related to the node and have a particular location which is fixed in most cases. They remain present on the same file system.



scripting. Name a few shells that are commonly used.

Basically, it acts as a direct link between the kernel and the user. It is possible for the system to run multiple shells at the same time despite the fact that there is always only one [kernel](#) that remains present.

Thus, when the command is given by the user, the shell begins exchanging information with the kernel. The operations are then executed and the information is conveyed to the user.

The shells that are commonly used in the scripting are ksh, bourne, bash, csh

## 12) How can you calculate the overall number of arguments that have been passed to a script? Why it is important?

This can easily be done by a command \$#. The same is required to know the overall load on a script. Every limit has a limit on handling the same and users have to check it often to make sure the script remains in the running mode and doesn't get fail.

## 13) In UNIX shell scripting, what there is a need to connect the system to the remote server according to you?

This is done generally when it comes to executing some special commands and the users are free to get this done through various commands. This simply lets users get an additional control or get some additional support or help.

This is actually considered as one of the effective ways for the same. However, this approach is to be secured with an authentication process that can be a simple username and a password.

Explore - [Linux Networking Commands](#)

## 14) What is the function of utilities that come with the open client driver in the shell scripting?

They are useful for connecting the system with a database server simply. The users have no performed this task to accomplish so many tasks that are relevant and important in scripting.

Actually, scripting needs a lot of data and information and it is not always possible that the same is kept at a particular location only. The users have to make sure of an error-free outcome in this approach.



for the users.

It stands for Master Boot Record and is basically a small program whose prime function is to make sure that the kernel is loaded during the system boot set up. It is present in the boot block.

It is nothing but the MBR that simply makes sure that the users can perform the boot looping tasks accurately and reliably.

### **16) Suppose you execute a command using exec, what will be the status of your current process in the shell?**

All the forked processes which are new get overlays when the exec is executed. The command simply gets executed without making any impact on the current process. Also, no new process will be created in this scenario.

### **17) In shell scripting, where you will find the login names of all the users on a system and how they can be printed at the same time?**

There is a file "etc/shadow" file which contains all the information related to the users and the same can be printed or displayed anytime when the need of same is felt. You have no reasons to worry about this and the command `awk-F` can let you print them all in one go.

### **18) How you will treat the quoted arguments as separate ones from the general? Is it possible to treat them in a single string?**

This is done through the command `$@`. Yes, the same is possible and for this, the command `$*` can be applied directly.

### **19) In Shell scripting, how you will display the process id of the process executing currently and the id of processes that take place recently?**

For this, there are actually two dedicated commands which can be executed directly and they are `$$` and `$`.

### **20) What according to you is the simplest method for copying a file from one system to another?**

There are certain utilities with the help of which this can be done easily and without worrying about anything. The users can simply keep up the pace with them when it comes to the



These utilities are FTP, SCP, and rsync. It is necessary to specify the hostname before these utilities are considered.

### **21) In Shell scripting, how the user can frequently monitor a log file that is updating?**

There is a concept of tailing in the shell scripting that can be applied for this. By using the tail-f filename this can be done. Actually, it will enable users to display the previous 10 lines on the output. The same reflects the part of the file which is updating continuously.

### **22) Is it possible to use a shell script to determine if the directory actually exists or not?**

Yes, this is possible and for this, the users are free to simply consider the UNIX test command. The option that is useful is the -d option. It is not always necessary that the information regarding the existence of a directory is displayed only when the directory is recognized by the system.

Information regarding all the directories whether known to the system or not can be displayed with this command. Generally, directories are present in the variable \$mydir.

### **23) Tell the procedure of accessing the command line arguments from a script?**

It can be done simply by executing the \$ in the system. It will show all the arguments that have already passed from the command line to a script.

Sometimes you need to specify the argument's numeric position and the users have to make sure that things are done in a rightful manner.

### **24) Are you familiar with the methods that are useful in debugging a problem related to the shell script?**

The answer to this question depends on the skills of a professional. There are various methods and it is not always necessary that all users consider a similar method for problem debugging. Also, it actually depends on the problem itself.

The first method that can help users to formulate the problem is simply inserting some debug statements. This will actually help the pinpoint to know how and where the problem



## 25) How the break and continue in the system are different from one another in scripting?

When a "Break" is used within a shell scripting loop is considered, it simply terminates the entire loop. On the other side, the "Continue" will make only the current iteration terminate.

However, you can have the loop available with you in the next iteration. These are the two powerful and in fact very useful features when it comes to constructing the shell.

## 26) Why writing the Shell Scripts is important according to you?

Well, this is important due to so many reasons and a few important ones are spotlighted below.

- When it comes to creating customized commands or creating your own, shell scripting is very useful and powerful.
- There are a lot of tasks that can easily be automated in the process.
- The users are responsible for providing the input and this clearly means there will be no errors in the output.
- It is good enough to be considered for time-saving.
- A lot of system administration tasks can be accomplished and can be automated.

## 27) Where exactly you can store the Shell programs in the system?

They are stored in a file which is tagged as Sh (Bourne Shell)

## 28) In Shell scripting, how can you say that C Shell is better than Bourne Shell?

- All the commands can be aliased simply with the C shell whereas the same is not possible in the case of Bourne Shell.
- Lengthy commands can be used again and again in the C shell whereas Bourne doesn't allow the same in all the cases.
- The command history can be accessed through the C shell but it cannot be accessed through the Bourne.
- There is no need to type the command again and again in the case of C.





The soft links can simply be made inactive in your delete the file and the same doesn't have any impact on the hard link and i.e. the users can access the file copy.

### 30) Do you need a separate compiler for executing a shell program?

No, the same is not required

### 31) What is Unix Shell?

A Unix shell is a command-line interpreter that provides a Unix-like command-line user interface.

### 32) List some Unix Shells.

- The Bourne Shell (sh)
- The Bourne Again Shell (bash)
- The C shell (csh or tsch)
- The Korn Shell (ksh)

### 33) When should we use the "if" condition (before)?

When running several commands "if" condition meets.

### 34) How do I print all arguments submitted on a command line?

echo \$@ or echo \$\*

### 35) Which is the Best Way To Pretty Print JSON On The Command-Line?

It can handle very large JSON structures, including streams.

Ex:

```
$ jq . <<< '{ "foo": "lorem", "bar": "ipsum" }'  
{  
  "bar": "ipsum",  
  "foo": "lorem"  
}
```



The following command is used: [ \$a -gt 12 ]

### 37) What are the advantages and disadvantages of shell scripting?

Advantages of shell scripting:

1. Can design applications (software) according to their platform.
2. To run a sequence of commands as a single command.
3. Portable (It can be executed in any Unix-like operating system without any modifications)

Disadvantages of shell scripting:

1. Slow execution speed compared to any programming languages
2. When a typing error occurs during the creation then it will delete the entire data as well as partition data.

### 38) What is the difference between &/&&?

- & - "Bitwise AND", evaluates both sides of the operation.
- && - "Logical AND Operator", evaluates at the left side of the operation (If it is True) it continues at the right side.

### 39) How to get the end line from a file?

```
tail -1
```

### 40) In a program, how to get/set an environment variable?

Getenv() and putenv()

### 41) Which command is helpful to forward errors to a file?

The command that used for forwarding error to a file is 2> filename

### 42) Which command helps in getting all files in the 3rd place an r, g, or l?

Command with on the 3rd place an r, g or i is Ls ??[rgi]\*

### 43) What Is The Difference Between \$\* And \$@?]



#### 44) Name the Command used for Displaying the list of files in a directory.

\$ ls -lrt | grep ^-? the command used for displaying the list of files in a directory.

#### 45) What is Shell Scripting?

Shell Scripting is a collection of commands put together into a file. The script is a command or an instruction given to process and this set of instructions is put together in a file to perform some task.

**Note:** In order to save time from manually providing one instruction at a time to process it, shell scripting is made available that can accept all the commands put together into a file and process them one after the other.

#### 46) What is Unix shell scripting?

Unix Commands are put together into one text file to execute one after the other.

#### 47) What are Shell Variables? Mention types of Shell Variables?

- Shell variables provide information to execute scripts/commands. There are two types of shell variables: Unix defined and User Defined.
- Unix-defined variables are by default set by UNIX, to define properties. For example, SHELL is a Unix variable that stored information about the working shell set to default.
- User-defined variables are defined by a developer to store/manage information. For example, \$ mm = 2018 defined by user means, the value of 2018 is assigned to variable mm.

#### 48) Command to wipe out variables defined.

If we consider \$ mm = 2018 as a variable defined by the user, then in order to wipe out the variable, the unset command is used.

Syntax: \$ unset mm

#### 49) List out a few Unix commands used very often.

Following are the list of basic Unix commands one should be aware of:



- mkdir - is to create a new folder/directory.
- rmdir - is to remove directory/file.
- cp - is to copy files.
- mv - is to move files.
- touch - creates an empty file

## 50) Syntax for ls command.

Syntax: \$ ls

## 51) Syntax for the cd command.

Syntax:

1. \$ cd - current directory is changed to the HOME directory.
2. \$ cd mindmajix: the current directory is changed to mindmajix directories.
3. \$ cd ..: Moves to the parent directory of the current working directory.

## 52) Syntax for mkdir and rmdir.

Syntax:

1. \$ mkdir mindmajix - Creates new folder with name "mindmajix"
2. \$ rmdir mindmajix\_temp - Removes folder with the name "mindmajix\_temp"

## 53) Difference between process and thread?

To put it in simple words, a thread is a small piece of code/instruction that needs to be executed in a process, whereas a process is a collection of one or more thread executions to complete one complete task.

**Note:** A process consists of multiple threads that can share resources among them as they belong to the same process. And threads that belong to different processes cannot share their resources.

## 54) Without restarting the machine, how can we remove all the running processes?

All the running processes in the current shell can be removed using the [Linux command](#).



While working with Vi Editor, different operations we perform are done in different modes.

1. **Command Mode** - Launching Vi automatically starts command mode.
2. **Edit Mode** - This mode provides an environment to edit text.
3. **Ex Mode** - Vi editor interaction is made available with file processing instructions.

### 56) Why do we use "\$?" in shell scripting?

- This command returns the exit status of the previously executed command.
- 0 for successful execution, and non-zero for failure.

### 57) What is "\$\*" in shell scripting?

Arguments list passed in the current process.

### 58) What is the command for a number of arguments passed in command-line arguments?

"\$#" is the command.

### 59) How can you get the PID of the current process?

"\$\$" is used.

### 60) Command to work with file permissions?

umask is the command to modify file permissions.

### 61) Explain the structure of File Permissions.

Each permission is assigned with weight to represent in command lines.

In the command line,

- if we mention 7, it includes all the three permissions r,w, and x (i.e., 4+2+1)
- if we mention 6, it includes two permissions r and w (i.e., 4+2)
- if we mention 5, it includes two permissions r and x (i.e., 4+1), and so on.

Consider below example command for file permissions:

```
$ chmod 754 mindmajix.txt
```



- Owner - 7
- People in the same group - 5
- Others - 4

## 62) Types of File Permissions.

- Read as r with weight 4
- Write as w with weight 2
- Execute as x with weight 1

## 63) Command to delete any file?

If we suppose the file name as "mindmajix\_temp\_files.txt", then

"`$rm mindmajix_temp_files.txt`" is the command to remove the file.

## 64) Command to append output to a file.

">>" is used to append output to the file mentioned.

For example, if we want to append Date to today.txt, the following is the command

```
$ date >> today.txt
```

# Join Our Newsletter

Stay updated with our newsletter, packed with Tutorials, Interview Questions, How-to's, Tips & Tricks, Latest Trends & Updates, and more ► Straight to your inbox!

**Majix**

Name

Email

**Subscribe**

## Course Schedule

Name	Dates	
Unix Shell Scripting Training	Jun 18 to Jul 03	<a href="#">VIEW DETAILS</a>
Unix Shell Scripting Training	Jun 21 to Jul 06	<a href="#">VIEW DETAILS</a>
Unix Shell Scripting Training	Jun 25 to Jul 10	<a href="#">VIEW DETAILS</a>
Unix Shell Scripting Training	Jun 28 to Jul 13	<a href="#">VIEW DETAILS</a>

**Last updated: 15 June 2022**

## About Author

**Ravindra Savaram**

Ravindra Savaram is a Content Lead at Mindmajix.com. His passion lies in writing articles on the most popular IT platforms including Machine learning, DevOps, Data Science,...

[Read More](#)



Linux Networking Training

🕒 30 hours

👤 1226



4.6

< 4 / 6 >

» EXPLORE COURSES

No comments

Sign up

Sign in



Start a conversation ...



Terms of use - Privacy - Report a bug

powered by GraphComment





