



InterviewBit

# Unix Interview Questions



To view the live version of the page, [click here.](#)

© Copyright by Interviewbit

# Contents

---

## Basic Unix Interview Questions

1. Explain Unix Architecture
2. Define a single-user system.
3. Name a few significant features of UNIX?
4. Can you write a command to erase all files in the current directory including all its sub-directories?
5. Describe a link in UNIX.
6. Describe pipes in Unix
7. In Shell scripting, how do you separate the grep and egrep?
8. What is the fork() system call?
9. What is meant by the term Super User?
10. What do chmod, chown, chgrp commands do?
11. Can you name the important standard streams in the UNIX shell scripting?
12. What is the 'nohup' in UNIX?

## Intermediate Unix Interview Questions

13. Differentiate between swapping and paging?
14. What is a daemon?
15. Can you explain the method of changing file access permission?
16. Explain the process model of Unix?
17. Explain the term filter.
18. What can you tell about shell variables?

## Intermediate Unix Interview Questions

(.....Continued)

19. What do you know about the MBR?
20. Explain the file system in UNIX
21. In shell scripting, what is the significance of the Shebang line?
22. Can you enlist some commonly used network commands?
23. Explain a path in UNIX and different types of pathnames.
24. Explain Superblock in UNIX.
25. Enlist some file manipulation commands in UNIX.
26. Explain networking stack and protocol.
27. Explain the alias mechanism.
28. What is a wildcard and how is it used?
29. What are system calls and library functions in terms of Unix commands?

## Advanced Unix Interview Questions

30. Explain Virtual memory
31. Explain the kill() system call and its return values?
32. Name the various commands that are used for the user information in UNIX.
33. Explain mount and unmount commands.
34. Can you explain a little bit about command substitution?
35. What is the difference between ps -ef and ps -auxwww?
36. What is the Zombie process in UNIX? How do you find the Zombie process?

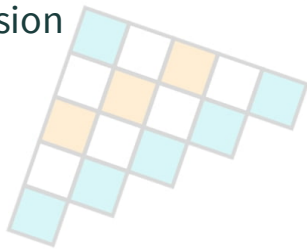
## Advanced Unix Interview Questions

(.....Continued)

- 37. You have an IP address in your network, how will you find the hostname and vice versa?
- 38. Explain system bootup in UNIX.
- 39. What are different classes of jobs?
- 40. What are the various IDs in UNIX processes?

## Conclusion

- 41. Conclusion



# Let's get Started

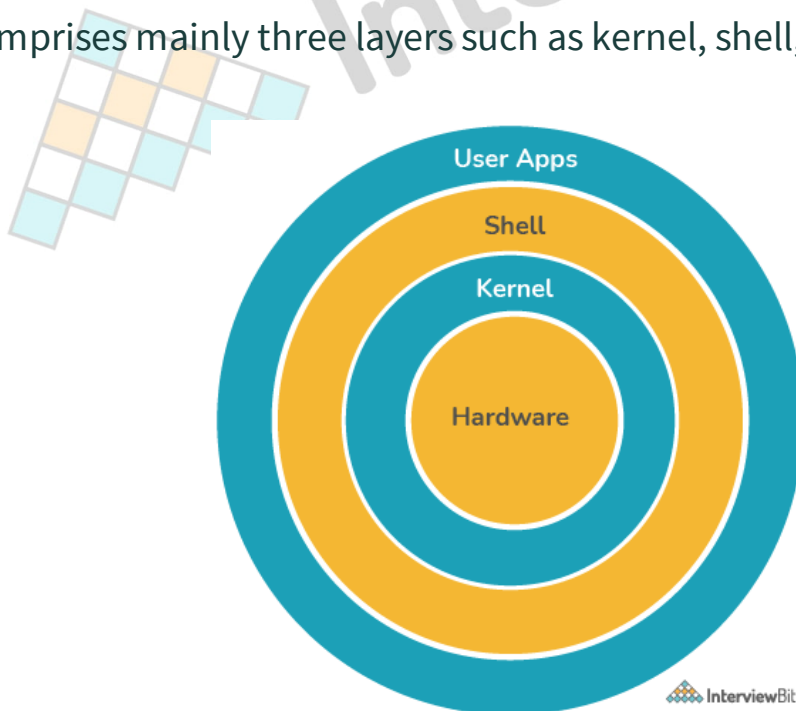
## **Premise**

Unix was started as an experiment at AT&T Bell Labs in the mid-1960s. Later on, it became a fully-fledged [operating system](#) that is designed for both efficient multi-tasking and multi-user functions. It is hardware-independent. It is written in a way that lets users do processing and gives control under a shell. The UNIX operating system is highly configurable and has evolved into a very complex, versatile, and scalable operating system handling almost every modern-day user task.

## **Basic Unix Interview Questions**

### **1. Explain Unix Architecture**

The Unix comprises mainly three layers such as kernel, shell, and user application.



Kernel - This is the core layer of the operating system which interacts with the hardware of the system. The kernel provides API via system calls to process the user requests. The kernel also provides services such as signal handling, synchronization, interprocess communication, file system services, network services, and hardware monitoring. Each time a process is started, the kernel has to initialize the process, assign the process a priority, allocate memory and resources for the process, and schedule the process to run. When the process terminates, the kernel frees any memory and/or resources held by the process.

## 2. Define a single-user system.

Single-user systems are the ones with an operating system to cater to only one user at a time. These are commonly our personal computers and became popular due to low hardware cost and wide range of software.

## 3. Name a few significant features of UNIX?

The following are a few features of UNIX:

- Hardware independent
- Multi-user operations
- Unix Shells
- Hierarchical file system
- Pipes and filters
- Utilities
- Development tools

## 4. Can you write a command to erase all files in the current directory including all its sub-directories?

`rm -r*` is used to erase all files in the current directory including all its sub-directories. `rm` is used for deleting files, but with the addition of option `-r` it erases all files in directories and subdirectories, and finally, an asterisk represents all entries.

## 5. Describe a link in UNIX.

Link is used to assigning more than one name to a file. It is like a pointer to a file and one file can have multiple pointers. There are two types of link

- **Hard link**- These hard-linked files are assigned to the same inode value as the original and thus reference the same physical location of the file. Also if the file is moved to a different directory the link will still work

```
ln [original filename] [link name]
```

- **Symbolic link**- A soft link is similar to the file shortcut feature and it contains a separate inode than the original one. If the original file is moved then the link might not work, but it can reference across the different file systems.

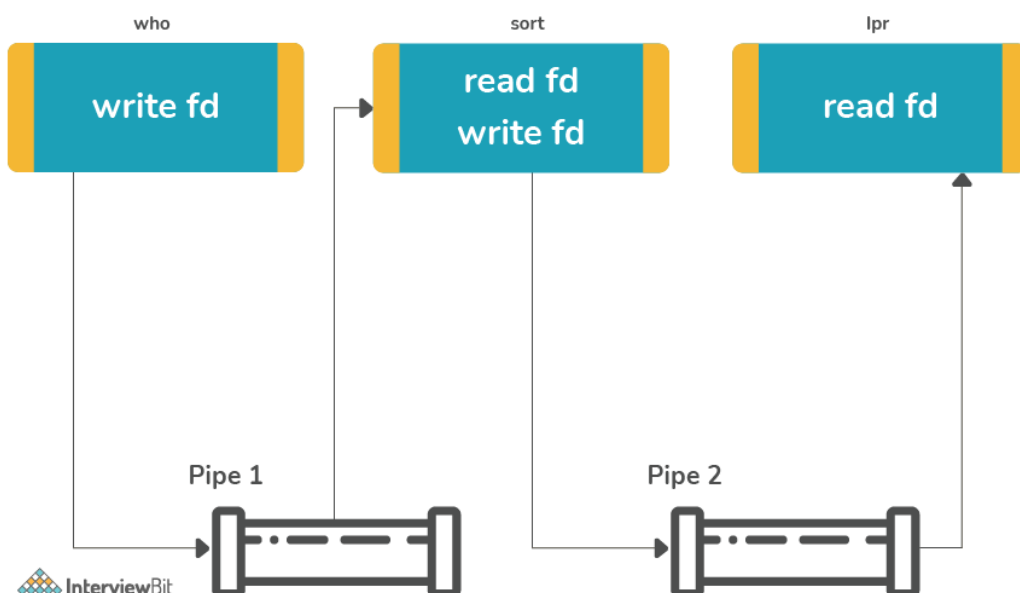
```
ln -s [original filename] [link name]
```

## 6. Describe pipes in Unix

A pipe is a unidirectional mechanism of interprocess communication. In a Unix command line, if a pipe is being used, the first process is assumed to be writing to stdout and the second is assumed to be reading from stdin.

```
$who | sort | lpr
```

This will create 3 processes with 2 pipes in between them.



## 7. In Shell scripting, how do you separate the grep and egrep?

egrep is an extended version of grep, in addition to searching using regular expressions, egrep can use extended regular expressions for searching.

```
$ ls | grep '.env|.json'
```

This will check whether any file has extension '.env|.json' literally

```
$ ls | egrep '.env|.json'
```

But in this case, it checks whether any file has extension '.env' or '.json' since '|' this will be considered as a metacharacter.

## 8. What is the fork() system call?

This is used to create another process that duplicates the entire process structure and address space. The newly created process is called the child process and the one from which it got replicated is called the parent process. This was used to achieve parallelism before threads. The fork() system call takes no arguments and returns an integer.

- 0 means that the child process is created successfully, and 0 refers to the child process within the parent process.
- -1 means that the system was unable to create another process.
- default positive integer > 0 is returned to the parent process which represents the process ID of the child process.

## 9. What is meant by the term Super User?

The Super User is a user with access to all files and commands within the system. In general, this superuser login is to access root and it is secured with the root password.

## 10. What do chmod, chown, chgrp commands do?

These are file management commands. These are used for:

- chmod - It changes the permission set of a file
- chown - It changes the ownership of the file.
- chgrp - It changes the group of the file.

```
$ chmod g+w testfile
```



It changes permission for a user group to write.

```
$ chown sroy8091 testfile
```

It changes the owner of testfile to sroy8091.

```
$ chgrp moderators testfile
```

It changes a group of testfile to moderators.

## 11. Can you name the important standard streams in the UNIX shell scripting?

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

## 12. What is the 'nohup' in UNIX?

nohup is a special command to run a process in the background even when a user logs off from the system. It also helps in creating daemon processes or some cleanup script of logs.

# Intermediate Unix Interview Questions

## 13. Differentiate between swapping and paging?

**Swapping:** In the case of swapping the main memory is loaded with the complete process and thus only the process less than the main memory size is executable. This is fairly easy to implement but not very efficient.

**Paging:** On the other hand in the case of paging instead of loading the whole process into the main memory only required memory pages are loaded. This is complex to implement but allows us to run a number of processes simultaneously.

## 14. What is a daemon?

A daemon is a background job or process responsible for a certain task or set of tasks. This is a fundamental concept to the Unix OS. The Unix kernel consists of numerous system daemons that are responsible for memory management, file system management, printer jobs, network connections, and several other services.

## 15. Can you explain the method of changing file access

There are three components of a file permissions

- **Owner permissions** – This determines what actions the owner of the file can perform.
- **Group permissions** – This determines what actions a group member(of the group that a file belongs to) can perform.
- **Other (world) permissions** – This is for everyone else.

```
$ ls -l /home/sroy8091
-rwxr-xr-- 1 sroy8091 users 1024 Mar 23 00:10 myfile
drwxr-xr-- 1 sroy8091 users 1024 Mar 23 00:10 mydir
```

r, w, and x represent read, write and execute respectively.

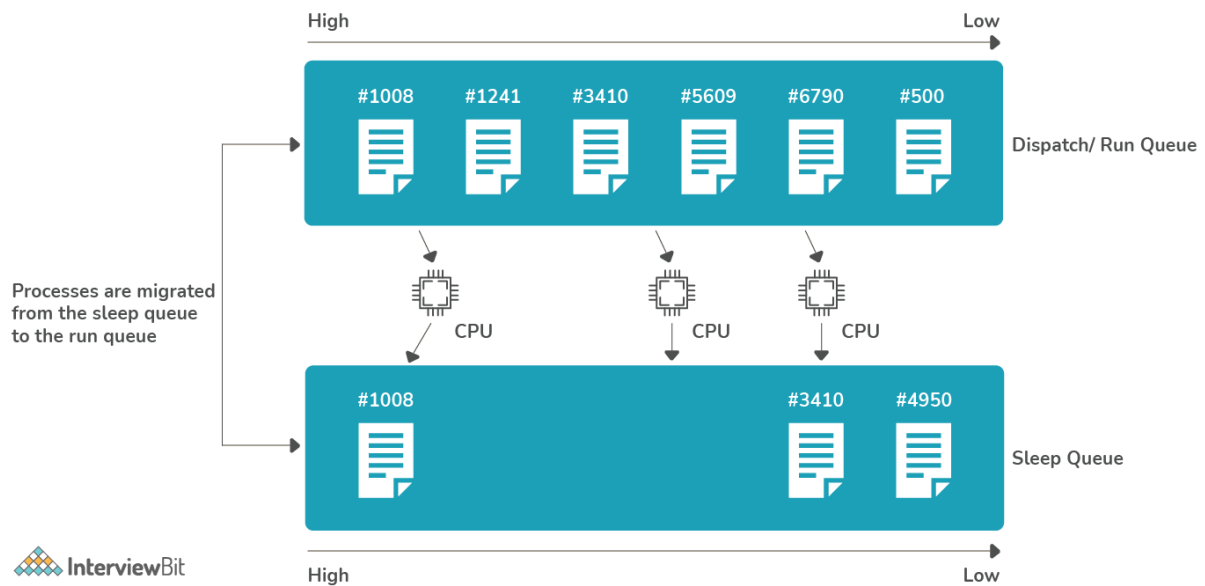
To change the file permission we need to run chmod command.

```
$ ls -l testfile
-rwxrwxr-- 1 sroy8091 users 1024 Mar 23 00:10 testfile
$ chmod o+wx testfile
$ ls -l testfile
-rwxrwxrwx 1 sroy8091 users 1024 Mar 23 00:10 testfile
```

Here we updated the permission for other users, from only read to read, write and execute.

## 16. Explain the process model of Unix?

Processes are scheduled using a priority-based preemptive round-robin scheduling algorithm. This ensures that there is no CPU starvation and CPU saturation. Based on the completion of the time slice or wait by any I/O, processes can be preempted from the run queue and moved to the sleep queue. While scheduling we might face deadlock on any resources hence, user applications need to provide deadlock detection to ensure that user processes do not deadlock on resources.



## 17. Explain the term filter.

A program that takes input from the standard input, and displays results to the standard output by performing some actions on it. The most popular example of a Unix filter is the **grep** command. This is used to search for a pattern in files(standard input) and print them out in the output screen(standard output).

## 18. What can you tell about shell variables?

The shell has a few sets of predefined internal variables with the support of creating new ones as well. Some of these are environment variables whereas others are local variables. These can be accessed by different processes. The name of a variable can contain only letters(A-z), numbers(0-9), or the underscore character ( \_).

```
NAME="Sumit Roy"
$ echo $NAME
$ unset NAME
```

## 19. What do you know about the MBR?

It stands for Master Boot Record and is a small program that is executed when the computer is booting up in order to find the OS and load it into memory. During the first stage of the system boot-up process, the BIOS (basic input-output system) searches for the MBR and then loads it into memory, after which the MBR takes over.

## 20. Explain the file system in UNIX

These are a fundamental component of the file system:

- Boot lock
- Inode lock
- Data Block
- Super Block

The UNIX file system has many different file system types such as ufs (UNIX filesystem), NFS (Network file system), vxfs (Veritas file system), and cdfs (CD-ROM filesystem). Among these, the ufs file system type is the most standard UNIX file system. Reads and writes to a ufs file system are done in blocks depending on the size of the file system block size. Block sizes can range from 1 KB to 8 KB depending on the file system type selected.

## 21. In shell scripting, what is the significance of the Shebang line?

It gives information about the location of where the engine is placed. The engine is the one that executes the given script. It is placed at the top of the script.

## 22. Can you enlist some commonly used network commands?

Some commonly used networking commands in Unix are:

- **telnet:** This is used for remote login and for communication with another hostname.
- **ping:** This is used for checking network connectivity.
- **hostname:** This gives the IP address and domain name.
- **nslookup:** This performs a DNS query.
- **xtracroute:** This is used to determine the number of hops and response time required to reach the network host.
- **netstat:** This provides information about system and ports, routing tables, interface statistics, etc.
- **tcpdump:** This provides info about both incoming and outgoing network traffic.

## 23. Explain a path in UNIX and different types of pathnames.

A Path is the unique location of a file/directory and a way to access it within the hierarchy of directories. There are basically two types of pathnames that are used in Unix.

- **Absolute Pathname:** The complete path specifying the location of a file/directory from the very start of the actual file system (root directory).  
Ex- /usr/local/Cellar/mysql/bin
- **Relative Pathname:** The path from the current working directory where the user is i.e. the present working directory (pwd).  
Ex- If current directory is /usr/local/Cellar the the relative path for bin is ./mysql/bin

## 24. Explain Superblock in UNIX.

A SuperBlock is a program that contains records of specific file systems such as the block size, the empty and the filled blocks and their respective counts, the size of the block groups, the disk block map, the size and location of the inode tables, and usage information. There are basically two types of superblocks:

- **Default superblock:** It is a fixed offset from the beginning of the system's disk partition.
- **Redundant superblock:** There are multiple copies of superblocks of a default one and are referenced when the default superblock is affected by a system crash or some errors.

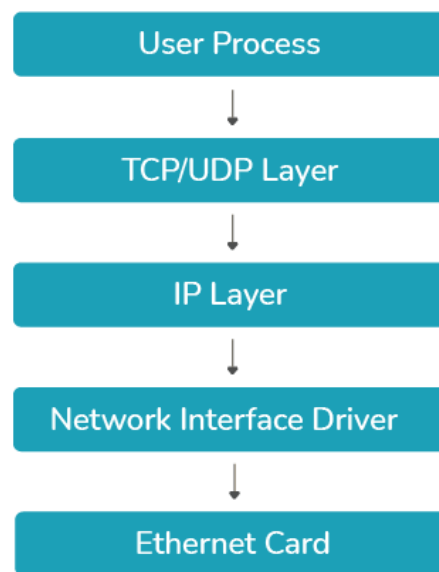
## 25. Enlist some file manipulation commands in UNIX.

These are few file manipulation commands:

- **cat filename** - Displays contents of the file.
- **cp source destination** - Copy the source file into the destination.
- **mv old\_name new\_name** - Move/rename.
- **rm filename** - Remove/delete filename.
- **touch filename** - creating/changing modification time.
- **ln [-s] old\_name new\_name** - Creating a soft link on an old name.
- **ls -F** - Displays information about the file type.
- **ls -ltr** - This will display in long format sorted by modified time with oldest first.

## 26. Explain networking stack and protocol.

The default networking stack protocol for the UNIX kernel is TCP/IP. TCP/IP works in two parts: TCP(Transmission Control Protocol) is responsible for the virtual circuit layer that provides bidirectional data transmission, IP(Internet Protocol) is responsible for packet transmission between the user processes and the networking layer.



## 27. Explain the alias mechanism.

To reduce the time of repeatedly typing long commands, an alias is used to assign another name to that command. Basically, it is a short form of a larger command.

```
alias gco='git checkout -b'
```

Here, we added an alias for the *checkout* command of git which is frequently used by devs making it less painful every time typing a long command.

## 28. What is a wildcard and how is it used?

A wildcard is a symbol that takes the place of an unknown character or set of characters. The asterisk(\*) and question mark(?) are wildcard characters usually used. The wildcards are majorly used in searching files with unknown characters in the filename (or directory).

## 29. What are system calls and library functions in terms of Unix commands?

**System calls:** System calls are an interface to the kernel itself that further requests the operating system to perform tasks on behalf of user programs. Whenever a system call is invoked within the operating system, the application program performs a context switch from user space to kernel space. These are not portable.

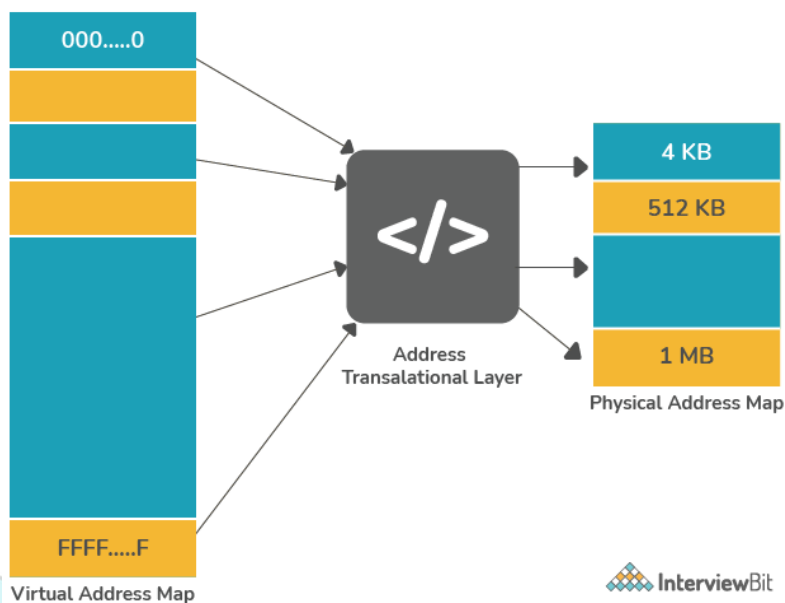
**Library functions:** These are functions that are not part of the kernel but are used by the different application programs. As compared to the system call it takes less time for execution and is portable and can perform certain tasks only in 'kernel mode'.

## Advanced Unix Interview Questions

### 30. Explain Virtual memory

Virtual Memory is a process or a system to address more memory than physically exists. The model consists of a set of memory pages, usually 4 KB each. There is a translation layer between virtual memory addresses and physical memory addresses. This translation layer is part of the kernel and is usually written in machine language(Assembly) to achieve optimal performance when translating and mapping addresses. Each time a user process accesses memory, the kernel uses the address translation layer to map the virtual memory address into a physical memory address. Virtual Addressing makes it possible to address larger amounts of memory than physically exist.





### 31. Explain the kill() system call and its return values?

kill() system call sends signals to any process which further takes suitable action according to the signal. It takes two arguments, first is PID, to which you want to send a signal and the signal you want to send is second. This method returns the following return values:

- 0 - means that the process exists with the given PID and the system allows sending signals to it.
- -1 and errno==ESRCH - means that the process/process group with specified PID does not exist
- -1 and errno==EPERM - means that the sender has no permission to send a signal to the target process.
- EINVAL - means an invalid signal was specified.

### 32. Name the various commands that are used for the user information in UNIX.

The various commands that are used for displaying the user information in Unix are:

- **id:** This displays the active user id with login and group.
- **last:** This displays the last login of the user in the system.
- **who:** This determines who is logged onto the system.
- **groupadd admin:** This is used to add group 'admin'.
- **usermod -a:** This is used to add an existing user to the group.

### 33. Explain mount and unmount commands.

**Mount:** The mount command mounts a storage device or file system onto an existing directory making it accessible to users.

**Unmount:** This unmount command detaches the mounted file system safely. It also informs the system of any pending read and write operations.

### 34. Can you explain a little bit about command substitution?

Command substitution is a method by which the command is substituted by its output. We can use this to assign the output to a variable.

```
$ DATE=`date`  
$ echo "Date is $DATE"  
Date is Thu May 27 03:59:57 IST 2021
```

### 35. What is the difference between ps -ef and ps -auxwww?

ps -ef will not list the processes with a very long command line while ps -auxwww will list those processes as well. This sometimes helps in debugging as we do not always want to list all the processes but if we are not getting any idea we should switch to the ps -auxwww command.

### 36. What is the Zombie process in UNIX? How do you find the Zombie process?

**Zombie processes** are those child processes that get finished before the parent process. So after finishing up, the process structure and address space are removed and freed back to the system but the entry in the process table still exists. To be able to get the status of the child process the parent calls wait(). In the interval between the child terminating and the parent calling wait(), the child is said to be a 'zombie'.

### 37. You have an IP address in your network, how will you find the hostname and vice versa?

**nslookup** command is used to query internet domain name servers which can be used to find hostname from an IP address and for reverse lookup also, Similarly **host** command can also be used.

### 38. Explain system bootup in UNIX.

It is the first thing when the system starts. It has majorly 5 phases

- Hardware - BIOS fired up and checks for the hardware connection
- Operating System Loader - The OS loader is located in the initial 512-byte block of the boot device known as the MBR
- Kernel - It initializes various components of the computer, operating system and each portion of software responsible for task, usually consider a driver
- root user-space process (init and inittab) - This defines what should be run when the /sbin/init program is instructed to enter a particular run-level, giving the administrator an easy way to establish an environment for some usage
- Boot scripts - For each managed service (mail, nfs server, cron, etc.), there is a single startup script located in a specific directory.

### 39. What are different classes of jobs?

The UNIX operating system job scheduler provides three different types of job classes

- **Time-share Job Class** - This is the default job class and has priority 0 through 59, with 59 being the highest priority. In this job class, each process is assigned a time slice where the time slice specifies the number of CPU clock ticks that a particular task can occupy. Once the process finishes its time slice, the process is placed on the sleep queue, and the priority decreases.
- **System Job Class** - This class is reserved for system daemon processes. The priority range for the system job class is between 60 and 99. These are of higher priority than the time-share job class. This is to ensure that the system can provide services to the time-share job class processes when system services such as memory, and/or file I/O are requested.
- **Real-Time Job Class** - This is the highest priority job class and has no time slice. Real-time process priorities range between 100 and 159. This ensures that critical processes always acquire the CPU as soon as the process is scheduled to run. Because there is no time slice the number of context switches is reduced significantly.

## 40. What are the various IDs in UNIX processes?

There are 3 ids associated with the Unix process:

- **PID** - Process Id, which is unique in the range of 0 to 3000.
- **PPID** - Parent process id
- **PGID** - Process group id, which is used to group processes together.

getppid() retrieves the Parent Process ID, getpid() retrieves the Process ID and getpgrp() retrieves the process group ID. The process also has a real user ID (the UID), an effective user ID (the EUID), a real user group ID (the GID), and an effective user group ID (the EGID).

## Conclusion

### 41. Conclusion

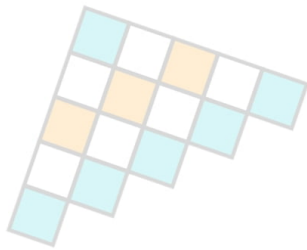
Unix is the basis of many modern operating systems in which the fundamental concept still carries on along with few [shell commands](#). Thus it is very important to have these concepts clear and these questions gave a very brief idea about different aspects of operating system concepts used in Unix.

**Recommended Tutorial:**

[Linux Interview Questions](#)

**References:**

[The Design of the UNIX Operating System](#) - Book



InterviewBit

# Links to More Interview Questions

---

[C Interview Questions](#)

[Php Interview Questions](#)

[C Sharp Interview Questions](#)

[Web Api Interview Questions](#)

[Hibernate Interview Questions](#)

[Node Js Interview Questions](#)

[Cpp Interview Questions](#)

[Oops Interview Questions](#)

[Devops Interview Questions](#)

[Machine Learning Interview Questions](#)

[Docker Interview Questions](#)

[Mysql Interview Questions](#)

[Css Interview Questions](#)

[Laravel Interview Questions](#)

[Asp Net Interview Questions](#)

[Django Interview Questions](#)

[Dot Net Interview Questions](#)

[Kubernetes Interview Questions](#)

[Operating System Interview Questions](#)

[React Native Interview Questions](#)

[Aws Interview Questions](#)

[Git Interview Questions](#)

[Java 8 Interview Questions](#)

[Mongodb Interview Questions](#)

[Dbms Interview Questions](#)

[Spring Boot Interview Questions](#)

[Power Bi Interview Questions](#)

[Pl Sql Interview Questions](#)

[Tableau Interview Questions](#)

[Linux Interview Questions](#)

[Ansible Interview Questions](#)

[Java Interview Questions](#)

[Jenkins Interview Questions](#)