

# UNIX Operating System Job Interview Questions And Answers



**Interview Questions Answers**

**<https://interviewquestionsanswers.org/>**

## About Interview Questions Answers

**Interview Questions Answers . ORG** is an interview preparation guide of thousands of Job Interview Questions And Answers, Job Interviews are always stressful even for job seekers who have gone on countless interviews. The best way to reduce the stress is to be prepared for your job interview. Take the time to review the standard interview questions you will most likely be asked. These interview questions and answers on UNIX Operating System will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts.

If you find any **question or answer** is incorrect or incomplete then you can **submit your question or answer** directly with out any registration or login at our website. You just need to visit [UNIX Operating System Interview Questions And Answers](#) to add your answer click on the *Submit Your Answer* links on the website; with each question to post your answer, if you want to ask any question then you will have a link *Submit Your Question*; that's will add your question in UNIX Operating System category. To ensure quality, each submission is checked by our team, before it becomes live. This [UNIX Operating System Interview preparation PDF](#) was generated at **Wednesday 29th November, 2023**

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.  
[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter for latest Jobs and interview preparation guides.  
<https://twitter.com/InterviewQA>

If you need any further assistance or have queries regarding this document or its material or any of other inquiry, please do not hesitate to contact us.

Best Of Luck.

**Interview Questions Answers.ORG Team**  
<https://InterviewQuestionsAnswers.ORG/>  
[Support@InterviewQuestionsAnswers.ORG](mailto:Support@InterviewQuestionsAnswers.ORG)



# UNIX Operating System Interview Questions And Answers Guide.

## Question - 1:

What Happens when you execute a command in UNIX OS?

### Ans:

When you enter 'ls' command to look at the contents of your current working directory, UNIX does a series of things to create an environment for ls and then run it: The shell has UNIX perform a fork. This creates a new process that the shell will use to run the ls program. The shell has UNIX perform an exec of the ls program. This replaces the shell program and data with the program and data for ls and then starts running that new program. The ls program is loaded into the new process context, replacing the text and data of the shell. The ls program performs its task, listing the contents of the current directory.

[View All Answers](#)

## Question - 2:

What Happens when you execute a program?

### Ans:

When you execute a program on your UNIX system, the system creates a special environment for that program. This environment contains everything needed for the system to run the program as if no other program were running on the system. Each process has process context, which is everything that is unique about the state of the program you are currently running. Every time you execute a program the UNIX system does a fork, which performs a series of operations to create a process context and then execute your program in that context. The steps include the following:

Allocate a slot in the process table, a list of currently running programs kept by UNIX.

Assign a unique process identifier (PID) to the process.

Copy the context of the parent, the process that requested the spawning of the new process.

Return the new PID to the parent process. This enables the parent process to examine or control the process directly. After the fork is complete, UNIX runs your program.

[View All Answers](#)

## Question - 3:

What are the process states in Unix?

### Ans:

As a process executes it changes state according to its circumstances. Unix processes have the following states:

Running : The process is either running or it is ready to run .

Waiting : The process is waiting for an event or for a resource.

Stopped : The process has been stopped, usually by receiving a signal.

Zombie : The process is dead but has not been removed from the process table.

[View All Answers](#)

## Question - 4:

What is a zombie?

### Ans:

When a program forks and the child finishes before the parent, the kernel still keeps some of its information about the child in case the parent might need it - for example, the parent may need to check the child's exit status. To be able to get this information, the parent calls `wait()`; In the interval between the child terminating and the parent calling `wait()`, the child is said to be a 'zombie' (If you do `ps`, the child will have a 'Z' in its status field to indicate this.)

[View All Answers](#)

## Question - 5:

How can a parent and child process communicate?

### Ans:

A parent and child can communicate through any of the normal inter-process communication schemes (pipes, sockets, message queues, shared memory), but also have some special ways to communicate that take advantage of their relationship as a parent and child. One of the most obvious is that the parent can get the exit status of the child.



[View All Answers](#)

### Question - 6:

How can you get/set an environment variable from a program in UNIX?

**Ans:**  
Getting the value of an environment variable is done by using ``getenv()'``. Setting the value of an environment variable is done by using ``putenv()'``.

[View All Answers](#)

### Question - 7:

List the system calls used for process management in UNIX?

**Ans:**  
System calls Description  
`fork()` To create a new process  
`exec()` To execute a new program in a process  
`wait()` To wait until a created process completes its execution  
`exit()` To exit from a process execution  
`getpid()` To get a process identifier of the current process  
`getppid()` To get parent process identifier  
`nice()` To bias the existing priority of a process  
`brk()` To increase/decrease the data segment size of a process

[View All Answers](#)

### Question - 8:

Predict the output of the following program code?

**Ans:**  

```
main()
{
fork(); fork(); fork();
printf("Hello World!");
}
```

**Answer:**  
"Hello World" will be printed 8 times.  
**Explanation:**  
 $2^n$  times where n is the number of calls to `fork()`

[View All Answers](#)

### Question - 9:

Predict the output of the following program code in UNIX?

**Ans:**  

```
main()
{
fork();
printf("Hello World!");
}
```

**Answer:**  
Hello World!Hello World!  
**Explanation:**  
The fork creates a child that is a duplicate of the parent process. The child begins from the `fork()`. All the statements after the call to `fork()` will be executed twice. (once by the parent process and other by child). The statement before `fork()` is executed only by the parent process.

[View All Answers](#)

### Question - 10:

Explain UNIX `fork()` system call.

**Ans:**  
The ``fork()'`` used to create a new process from an existing process. The new process is called the child process, and the existing process is called the parent. We can tell which is which by checking the return value from ``fork()'``. The parent gets the child's pid returned to him, but the child gets 0 returned to him.

[View All Answers](#)

### Question - 11:

What are various IDs associated with a process in UNIX OS?

**Ans:**  
Unix identifies each process with a unique integer called ProcessID. The process that executes the request for creation of a process is called the 'parent process' whose PID is 'Parent Process ID'. Every process is associated with a particular user called the 'owner' who has privileges over the process. The identification for the user is 'UserID'. Owner is the user who executes the process. Process also has 'Effective User ID' which determines the access privileges for accessing resources like files.  
`getpid()` -process id  
`getppid()` -parent process id  
`getuid()` -user id  
`geteuid()` -effective user id



[View All Answers](#)

### Question - 12:

Brief about the initial process sequence while the UNIX system boots up.

#### Ans:

While booting, special process called the 'swapper' or 'scheduler' is created with Process-ID 0. The swapper manages memory allocation for processes and influences CPU allocation. The swapper inturn creates 3 children:

the process dispatcher,  
vhand and  
dbflush

with IDs 1,2 and 3 respectively.

This is done by executing the file /etc/init. Process dispatcher gives birth to the shell. Unix keeps track of all the processes in an internal data structure called the Process Table (listing command is ps -el).

[View All Answers](#)

### Question - 13:

What is a UNIX OS shell?

#### Ans:

A shell is an interactive user interface to an operating system services that allows an user to enter commands as character strings or through a graphical user interface. The shell converts them to system calls to the OS or forks off a process to execute the command. System call results and other information from the OS are presented to the user through an interactive interface. Commonly used shells are sh,csh,ks etc.

[View All Answers](#)

### Question - 14:

How does the inode map to data block of a file in UNIX OS?

#### Ans:

Inode has 13 block addresses. The first 10 are direct block addresses of the first 10 data blocks in the file. The 11th address points to a one-level index block. The 12th address points to a two-level (double in-direction) index block. The 13th address points to a three-level(triple in-direction)index block. This provides a very large maximum file size with efficient access to large files, but also small files are accessed directly in one disk read.

[View All Answers](#)

### Question - 15:

Discuss the mount and unmount system calls in UNIX OS?

#### Ans:

The privileged mount system call is used to attach a file system to a directory of another file system; the unmount system call detaches a file system. When you mount another file system on to your directory, you are essentially splicing one directory tree onto a branch in another directory tree. The first argument to mount call is the mount point, that is , a directory in the current file naming system. The second argument is the file system to mount to that point. When you insert a cdrom to your unix system's drive, the file system in the cdrom automatically mounts to /dev/cdrom in your system.

[View All Answers](#)

### Question - 16:

How do you create special files like named pipes and device files in UNIX OS?

#### Ans:

The system call mknod creates special files in the following sequence.

1. kernel assigns new inode,
2. sets the file type to indicate that the file is a pipe, directory or special file,
3. If it is a device file, it makes the other entries like major, minor device numbers.

For example:

If the device is a disk, major device number refers to the disk controller and minor device number is the disk.

[View All Answers](#)

### Question - 17:

What is a FIFO in UNIX OS?

#### Ans:

FIFO are otherwise called as 'named pipes'. FIFO (first-in-first-out) is a special file which is said to be data transient. Once data is read from named pipe, it cannot be read again. Also, data can be read only in the order written. It is used in interprocess communication where a process writes to one end of the pipe (producer) and the other reads from the other end (consumer).

[View All Answers](#)

### Question - 18:

What are links and symbolic links in UNIX OS file system?

#### Ans:

A link is a second name (not a file) for a file. Links can be used to assign more than one name to a file, but cannot be used to assign a directory more than one name or link filenames on different computers.

Symbolic link 'is' a file that only contains the name of another file.Operation on the symbolic link is directed to the file pointed by the it.Both the limitations of links are eliminated in symbolic links.



Commands for linking files are:  
Link ln filename1 filename2  
Symbolic link ln -s filename1 filename2

[View All Answers](#)

### Question - 19:

How do you change File Access Permissions in UNIX OS?

#### Ans:

Every file has following attributes:  
owner's user ID ( 16 bit integer )  
owner's group ID ( 16 bit integer )  
File access mode word  
'r w x -r w x- r w x'  
(user permission-group permission-others permission)  
r-read, w-write, x-execute  
To change the access mode, we use chmod(filename,mode).

Example 1:

To change mode of myfile to 'rw-rw-r--' (ie. read, write permission for user - read,write permission for group - only read permission for others) we give the args as:  
chmod(myfile,0664) .

Each operation is represented by discrete values

'r' is 4

'w' is 2

'x' is 1

Therefore, for 'rw' the value is 6(4+2).

Example 2:

To change mode of myfile to 'rwxr--r--' we give the args as:  
chmod(myfile,0744).

[View All Answers](#)

### Question - 20:

What are the Unix system calls for I/O?

#### Ans:

open(pathname,flag,mode) - open file  
creat(pathname,mode) - create file  
close(filedes) - close an open file  
read(filedes,buffer,bytes) - read data from an open file  
write(filedes,buffer,bytes) - write data to an open file  
lseek(filedes,offset,from) - position an open file  
dup(filedes) - duplicate an existing file descriptor  
dup2(oldfd,newfd) - duplicate to a desired file descriptor  
fcntl(filedes,cmd,arg) - change properties of an open file  
ioctl(filedes,request,arg) - change the behaviour of an open file

The difference between fcntl and ioctl is that the former is intended for any open file, while the latter is for device-specific operations.

[View All Answers](#)

### Question - 21:

Brief about the directory representation in UNIX OS?

#### Ans:

A Unix directory is a file containing a correspondence between filenames and inodes. A directory is a special file that the kernel maintains. Only kernel modifies directories, but processes can read directories. The contents of a directory are a list of filename and inode number pairs. When new directories are created, kernel makes two entries named . (refers to the directory itself) and .. (refers to parent directory).  
System call for creating directory is mkdir (pathname, mode).

[View All Answers](#)

### Question - 22:

What is inode in UNIX OS?

#### Ans:

All UNIX files have its description stored in a structure called inode. The inode contains info about the file-size, its location, time of last access, time of last modification, permission and so on. Directories are also represented as files and have an associated inode. In addition to descriptions about the file, the inode contains pointers to the data blocks of the file. If the file is large, inode has indirect pointer to a block of pointers to additional data blocks (this further aggregates for larger files). A block is typically 8k.

Inode consists of the following fields:

File owner identifier  
File type  
File access permissions  
File access times  
Number of links  
File size  
Location of the file data

[View All Answers](#)

### Question - 23:



How are devices represented in UNIX OS?

**Ans:**

All devices are represented by files called special files that are located in /dev directory. Thus, device files and other files are named and accessed in the same way. A 'regular file' is just an ordinary data file in the disk. A 'block special file' represents a device with characteristics similar to a disk (data transfer in terms of blocks). A 'character special file' represents a device with characteristics similar to a keyboard (data transfer is by stream of bits in sequential order).

[View All Answers](#)

# Operating System Most Popular & Related Interview Guides

- 1 : [RTOS Interview Questions and Answers.](#)
- 2 : [Windows 7 Interview Questions and Answers.](#)
- 3 : [MAC Operating System Interview Questions and Answers.](#)
- 4 : [Disk Operating System \(DOS\) Interview Questions and Answers.](#)
- 5 : [Shell Scripting Interview Questions and Answers.](#)
- 6 : [Operating System \(OS\) Interview Questions and Answers.](#)
- 7 : [Solaris Interview Questions and Answers.](#)
- 8 : [Solaris Admin Interview Questions and Answers.](#)
- 9 : [VxWorks Interview Questions and Answers.](#)
- 10 : [OS Memory Management Interview Questions and Answers.](#)

Follow us on FaceBook

[www.facebook.com/InterviewQuestionsAnswers.Org](http://www.facebook.com/InterviewQuestionsAnswers.Org)

Follow us on Twitter

<https://twitter.com/InterviewQA>

For any inquiry please do not hesitate to contact us.

Interview Questions Answers.ORG Team

[https://InterviewQuestionsAnswers.ORG/  
support@InterviewQuestionsAnswers.ORG](https://InterviewQuestionsAnswers.ORG/support@InterviewQuestionsAnswers.ORG)