

# Unix IPC 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 IPC 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 IPC 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 IPC category. To ensure quality, each submission is checked by our team, before it becomes live. This [Unix IPC 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>



## Unix IPC Interview Questions And Answers Guide.

### Question - 1:

Explain a zombie?

### Ans:

A system that has been taken over using Remote Control Software. Zombies are often used to send spam or to attack remote servers with an overwhelming amount of traffic (a Distributed Denial of Service Attack).

[View All Answers](#)

### Question - 2:

What is Daemon?

### Ans:

The processes like vhand, bdfush, sched are housed in kernel file or /unix system which are known as daemons. These files run in the background without users request. These are created when the system boots up and remains active till it shut down or hang. These are not linked to any user or any terminal. We can't kill a daemon.

[View All Answers](#)

### Question - 3:

What Happens when we execute a Unix command?

### Ans:

When command is given then unix os will fork the shell i.e will create a new process and will execute the command using exec command...something like suppose you gave command "ls" in the shell then...

```
fork();
```

```
exec(ls);
```

It will give you the result and after this the child process will die.

[View All Answers](#)

### Question - 4:

What is an advantage of executing a process in background?

### Ans:

There are two main advantages:

1. you can use console to execute your command.
2. More importantly if one is connected through remote console and as there is no controlling terminal attached to background process even if your terminal gets disconnected the process continues (with help of nohup in shell script or by using setsid in C)

[View All Answers](#)

### Question - 5:

How do you execute one program from within another?

### Ans:

by calling system call fork() to create child process which



handles the other.

[View All Answers](#)

### Question - 6:

What is IPC? What are the various schemes available?

#### Ans:

Inter Process Communication. IPC is used to pass information between two or more processes.

Schemes are pipes, shared memory & semaphore.

Below are the different IPC methods

1. Semaphores
2. FIFO's (Also called Named Pipes)
3. Message Queues
4. Shared Memory

[View All Answers](#)

### Question - 7:

Please Describe the initial process sequence while the system boots up?

#### Ans:

- Hardware Boot
- Read boot block (block 0) and load to memory
- Transfer control to kernel
- Kernel initialization
- Mount the root file system on / and create environment for process 0
- Process 0 invokes process 1 (init process)
- init process invokes process getty
- getty invokes login

[View All Answers](#)

### Question - 8:

What are the system calls used for process management:

#### Ans:

The system calls for process management are

- Fork() > create child process
- Vfork() > create child process (copy-on-write only)
- exec() > exec to do a different task
- wait() > wait for a process to complete execution
- kill() > to send a signal to a process
- signal() > to handle a signal
- sigaction() > handle signal
- exit() > exit from a process execution
- \_exit() > same as exit() but with out clean up and many more .....

[View All Answers](#)

### Question - 9:

How to write the program on full-duplex communication on bidirectional(e.g using two pipes)?

#### Ans:

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/ipc.h>
#include<sys/types.h>
int main()
{
    int p1[2],p2[2],i;
    char buf[5];
    pipe(p1);
    pipe(p2);
    if(fork()==0)
    {
        printf("
this is child(the input text is text)
");
        close(p1[0]);
        close(p2[1]);
        write(p1[1],"text",5);
        read(p2[0],buf,5);
        write(1,buf,5);
    }
    else
    {
```



```
printf("
this is parent(the output text is text)
");
close(p2[0]);
close(p1[1]);
read(p1[0],buf,5)
for(i=0;i<4;i++)
buf[i]=to upper(buf[i]);
write(p2[1],buf,5)
}
}
```

[View All Answers](#)

### Question - 10:

What is semaphore?

#### Ans:

a semaphor is nothing but a term used in unix for a variable which acts as a counter.for instance there may be times when two processes try to access the same file simultaneously.in this event we must control the access of the when the other process is accessing this is done by assigning value to a semaphore. the value of the semaphore is initialized by the 1st process when the file is in access by it.when the 2nd process try to access the file it checks the value of the semaphore and if it finds the value as initialized it does not access the file.

[View All Answers](#)

### Question - 11:

Explain What is the process id for kernel process?

#### Ans:

The Process id are sequentially numbered and the first process id created is 'PID 0', which loads the data structures and resources of Kernel which also forks the next process "INIT" which has the process id '1'.

[View All Answers](#)

### Question - 12:

What is the process id for kernel process?

#### Ans:

zero(0)

[View All Answers](#)

### Question - 13:

What is the very first process created by kernel?

#### Ans:

The first process in Unix is INIT.

[View All Answers](#)

### Question - 14:

Explain Linking across directories?

#### Ans:

It's possible through both symbolic and hard links. However, hard links are only possible if both source and destination belong to the same volume.

[View All Answers](#)

### Question - 15:

What is ln(linking)?

#### Ans:

Linking is also two types.

1>Hard link.

2>soft link.

For hard :-

ln <source> <destination>

for softlink :-

ln -s <source> <destination>

[View All Answers](#)

**Question - 16:**

What is fork()?

**Ans:**

Fork system call is used to create a new process. Fork is called once and returns twice. It will return 0 to the newly created process (child process) and process id of child to the calling process (parent process).

The child process gets copy of parents data, stack and heap segment. The code segment is common for both. Both the processes will resume execution from the command next to fork. Scheduler decide which process to run first.

[View All Answers](#)

**Question - 17:**

Max relax-able permission value with out giving write permission to others?

**Ans:**

```
chmod 755 <filename>
```

[View All Answers](#)

**Question - 18:**

What is i-node numbers?

**Ans:**

File is identified by the I-node number in linux. Kernel uses inode number to access the file. It is unique for the particular filesystem. Inode contains all the information about the file like , file size, access permissions, time stamp , uid , gid , pointers to data blocks..etc.

[View All Answers](#)

**Question - 19:**

What is the condition required for dead lock in Unix system?

**Ans:**

deadlock occurs because of

- 1.no preemption
- 2.circular waiting i.e.. when A process is waiting for the resources which are held by the process B, which is waiting for the resources engaged by process A.

[View All Answers](#)

**Question - 20:**

Tell me set-user-id is related to (in unix)?

**Ans:**

setuid short for set user ID upon execution is Unix access rights flag that allow users to run an executable with the permissions of the executable's owner. This is often used to allow users on a computer system to run programs with temporarily elevated privileges in order to perform a specific task. While the assumed user id privileges provided are not always elevated, at a minimum that is specific. setuid and setgid are needed for tasks that require higher privileges than those which a common user has, such as changing his or her login password. Some of the tasks that require elevated privileges may not immediately be obvious, though - such as the ping command, which must send and listen for control packets on a network interface.

[View All Answers](#)

**Question - 21:**

Explain the initial process sequence while the system boots up?

**Ans:**

While booting, special process called the 'swapper' or 'scheduler' is created by the Process-ID 0. The swapper manages memory allocation for processes and influences CPU allocation. The swapper in turn creates 3 children: the process dispatcher, vhand, etc...

[View All Answers](#)

**Question - 22:**



What are various IDs associated with a process?

**Ans:**

Real User id, Effective User Id, saved user id.  
Real Group id, Effective Group Id, saved Group id.

[View All Answers](#)

**Question - 23:**

Please Explain fork() system call?

**Ans:**

fork is a system call used for creating child processes of a parent process. it returns the process id of the created child process. after that pid (process id) is checked if it is negative, it means no child process is created, pid==0 implies the id of the newly created process and pid>0 is the id of child process given to the parent process. the statements following fork system call are executed by both the parent and child process. and one more thing, the parent and child process have the exact copy of address space but it exist separately for the two processes.

[View All Answers](#)

**Question - 24:**

Predict the output of the following program code

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

**Ans:**

prints Hello world Hello world  
All the statements after the call to fork() will be executed twice

[View All Answers](#)

**Question - 25:**

What are the system calls used for process management?

**Ans:**

This should not allow to increase process priority.

[View All Answers](#)

**Question - 26:**

How to get or set an environment variable from a program?

**Ans:**

to get environment variable execute getenv();  
to set environment variable execute setenv();

[View All Answers](#)

# OS Unix Most Popular & Related Interview Guides

- 1 : [Unix General Interview Questions and Answers.](#)
- 2 : [Unix System Calls Interview Questions and Answers.](#)
- 3 : [Unix Threads 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)