

Linux Debugging Interview Questions And Answers Guide.



Global Guideline.

<https://globalguideline.com/>



Linux Debugging Job Interview Preparation Guide.

Question # 1

Which one of the following string will print first by this program?

```
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
    printf("Googlen");
    pthread_exit("Bye");
}
int main()
{
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
        perror("pthread_create");
    printf("Linuxn");
    if(pthread_join(pt,&res_t) != 0)
        perror("pthread_join");
    return 0;
}
```

- a) Linux
- b) Google
- c) it can not be predicted
- d) none of the mentioned

Answer:-

- b) Google

Explanation:It depends upon the scheduler.

Output:

```
[root@localhost Google]# gcc -o san san.c -lpthread
[root@localhost Google]# ./san
Google
Linux
[root@localhost threads]#
```

[Read More Answers.](#)

Question # 2

What is the output of this program?

```
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
    int ret;
    ret = pthread_exit("Bye");
    printf("%dn",ret);
}
int main()
{
    pthread_t pt;
    void *res_t;
    if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
        perror("pthread_create");
    if(pthread_join(pt,&res_t) != 0)
        perror("pthread_join");
}
```



- ```
 return 0;
}
a) 0
b) 1
c) -1
d) none of the mentioned
```

**Answer:-**

- d) none of the mentioned

Explanation:

The function pthread\_exit() does not return any value. Hence this program will give an error.

Output:

```
[root@localhost Google]# gcc -o san san.c -lpthread
san.c: In function 'fun_t':
san.c:8:6: error: void value not ignored as it ought to be
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 3

What is the output of this program no 1?

```
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
 printf("googlen");
 pthread_exit("Bye");
}
int main()
{
 pthread_t pt;
 void *res_t;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 return 0;
}
```

- a) this program will print the string "google"  
b) this program will print nothing  
c) segmentation fault  
d) run time error

**Answer:-**

- b) this program will print nothing

Explanation:The pthread\_join() function waits for the thread to terminate.

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
[root@localhost google]# ./san
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 4

What is the output of this program no 2?

```
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
 printf("%dn",a);
 pthread_exit("Bye");
}
int main()
{
 int a;
 pthread_t pt;
 void *res_t;
 a = 10;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 if(pthread_join(pt,&res_t) != 0)
 perror("pthread_join");
 return 0;
}
```

- a) 10  
b) 0  
c) -1  
d) none of the mentioned

**Answer:-**



d) none of the mentioned

Explanation:

Each thread has its own stack so local variables are not shared among thread. Hence this program will give an error.

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
san.c: In function 'fun_t':
san.c:7:16: error: 'a' undeclared (first use in this function)
san.c:7:16: note: each undeclared identifier is reported only once for each function it appears in
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 5

What is the output of this program no 3?

```
#include<stdio.h>
#include<pthread.h>

int a;
void *fun_t(void *arg);
void *fun_t(void *arg)
{
 printf("%dn",a);
 pthread_exit("Bye");
}
int main()
{
 pthread_t pt;
 void *res_t;
 a = 10;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 if(pthread_join(pt,&res_t) != 0)
 perror("pthread_join");
 return 0;
}
a) 10
b) 0
c) -1
d) none of the mentioned
```

Answer:-

a) 10

Explanation:

Thread of the same process shares the global variables.

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
[root@localhost google]# ./san
10
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 6

What is the output of this program no 4?

```
#include<stdio.h>
#include<pthread.h>

int a;
void *fun_t(void *arg);
void *fun_t(void *arg)
{
 a = 20;
 pthread_exit("Bye");
}
int main()
{
 pthread_t pt;
 void *res_t;
 a = 10;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 if(pthread_join(pt,&res_t) != 0)
 perror("pthread_join");
 printf("%dn",a);
 return 0;
}
a) 10
b) 20
c) segmentation fault
d) none of the mentioned
```

Answer:-



b) 20

Explanation:

In this program the value of variable "a" is changed by the thread "fun\_t".

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
```

```
[root@localhost google]# ./san
```

```
20
```

```
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 7

Which one of the following statement is not true about this program?

```
#include<stdio.h>
```

```
#include<pthread.h>
```

```
void *fun_t(void *arg);
```

```
void *fun_t(void *arg)
```

```
{
```

```
 printf("%dn",getpid());
```

```
 pthread_exit("Bye");
```

```
}
```

```
int main()
```

```
{
```

```
 pthread_t pt;
```

```
 void *res_t;
```

```
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
```

```
 perror("pthread_create");
```

```
 if(pthread_join(pt,&res_t) != 0)
```

```
 perror("pthread_join");
```

```
 printf("%dn",getpid());
```

```
 return 0;
```

```
}
```

a) both printf statements will print the same value

b) both printf statements will print the different values

c) this program will print nothing

d) none of the mentioned

**Answer:-**

a) both printf statements will print the same value

Explanation:

All the threads of the same process have same PID.

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
```

```
[root@localhost google]# ./san
```

```
12981
```

```
12981
```

```
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 8

What is the output of this program no 6?

```
#include<stdio.h>
```

```
#include<pthread.h>
```

```
#include<fcntl.h>
```

```
int fd;
```

```
void *fun_t(void *arg);
```

```
void *fun_t(void *arg)
```

```
{
```

```
 char buff[10];
```

```
 int count;
```

```
 count = read(fd,buff,10);
```

```
 printf("%dn",count);
```

```
 pthread_exit("Bye");
```

```
}
```

```
int main()
```

```
{
```

```
 pthread_t pt;
```

```
 void *res_t;
```

```
 fd = open("san.c",O_RDONLY);
```

```
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
```

```
 perror("pthread_create");
```

```
 if(pthread_join(pt,&res_t) != 0)
```

```
 perror("pthread_join");
```

```
 return 0;
```

```
}
```

a) 10

b) 0

c) -1

d) segmentation fault



### Answer:-

a) 10

Explanation:

Open file descriptors can be shared between threads of the same process

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
[root@localhost google]# ./san
10
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 9

What is the output of this program no 7?

```
#include<stdio.h>
#include<pthread.h>
#include<fcntl.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
 pthread_exit("Bye");
 printf("googlen");
}
int main()
{
 pthread_t pt;
 void *res_t;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 if(pthread_join(pt,&res_t) != 0)
 perror("pthread_join");
 printf("%sn",res_t);
 return 0;
}
```

a) google

b) Bye

c) segmentation fault

d) run time error

### Answer:-

b) Bye

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
[root@localhost google]# ./san
Bye
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 10

What is the output of this program no 8?

```
#include<stdio.h>
#include<pthread.h>

void *fun_t(void *arg);
void *fun_t(void *arg)
{
 sleep(1);
}
int main()
{
 pthread_t pt;
 void *res_t;
 if(pthread_create(&pt,NULL,fun_t,NULL) != 0)
 perror("pthread_create");
 if(pthread_join(pt,&res_t) != 0)
 perror("pthread_join");
 printf("%sn",res_t);
 return 0;
}
```

a) this process will pause

b) segmentation fault

c) run time error

d) none of the mentioned

### Answer:-

b) segmentation fault

Explanation:

This program is trying to print the return value of the thread, but pthread\_exit() function is not present in the thread.

Output:

```
[root@localhost google]# gcc -o san san.c -lpthread
```



```
[root@localhost google]# ./san
Segmentation fault (core dumped)
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 11

For debugging with GDB, the file "google" can be created with the command:

- a) gcc -g -o google google.c
- b) gcc -g google.c
- c) gdb google
- d) none of the mentioned

**Answer:-**

- a) gcc -g -o google google.c

[Read More Answers.](#)

### Question # 12

For debugging with GDB, the compiled program can be run by the command

- a) run
- b) execute
- c) ./
- d) none of the mentioned

**Answer:-**

- a) run

[Read More Answers.](#)

### Question # 13

In GDB, breakpoints can be set by the command:

- a) break
- b) b
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 14

GDB stands for:

- a) GNU debugger
- b) general debugging breakpoint
- c) general debugger
- d) none of the mentioned

**Answer:-**

- a) GNU debugger

[Read More Answers.](#)

### Question # 15

GDB can be used for:

- a) c language
- b) c++ language
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 16

The command "gdb google"

- a) will start debugging for the file "google" if the file is compiled with -g option with GCC
- b) will create executable for debugging
- c) will provide all errors present in the file "google"
- d) none of the mentioned

**Answer:-**

- a) will start debugging for the file "google" if the file is compiled with -g option with GCC

[Read More Answers.](#)

### Question # 17

In debugging with GDB, break points can be set to:



- a) any line
- b) any function
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 18

In GDB debugging, we can proceed to the next break-point with command:

- a) next
- b) continue
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- b) continue

[Read More Answers.](#)

### Question # 19

At the time of debugging with GDB, if we just press ENTER:

- a) GDB will repeat the same command you just gave it
- b) GDB will do nothing
- c) GDB will exit
- d) none of the mentioned

**Answer:-**

- a) GDB will repeat the same command you just gave it

[Read More Answers.](#)

### Question # 20

To print the value of a variable while debugging with GDB, \_\_\_\_\_ command can be used.

- a) printf
- b) print
- c) show
- d) none of the mentioned

**Answer:-**

- b) print

[Read More Answers.](#)

### Question # 21

Which GDB command prints the value of a variable in hex.

- a) print/x
- b) print/h
- c) print/e
- d) none of the mentioned

**Answer:-**

- a) print/x

[Read More Answers.](#)

### Question # 22

Which GDB command interrupts the program whenever the value of a variable is modified and prints the value old and new values of the variable?

- a) watch
- b) show
- c) trace
- d) none of the mentioned

**Answer:-**

- a) watch

[Read More Answers.](#)

### Question # 23

Which GDB command produces a stack trace of the function calls that lead to a segmentation fault?

- a) trace
- b) backtrace
- c) forwardtrace
- d) none of the mentioned

**Answer:-**

- b) backtrace

[Read More Answers.](#)





### Question # 24

The specific break point can be deleted by \_\_\_\_\_ command in GDB.

- a) delete
- b) del
- c) remove
- d) none of the mentioned

**Answer:-**

- a) delete

[Read More Answers.](#)

### Question # 25

The "step" command of GDB:

- a) executes the current line of the program
- b) stops the next statement to be executed
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 26

Which GDB command reloads the debugging information?

- a) file
- b) reload
- c) debug
- d) none of the mentioned

**Answer:-**

- a) file

[Read More Answers.](#)

### Question # 27

GDB can be used:

- a) to find out the memory leakages
- b) to get the result of a particular expression in a program
- c) to find the reason of segmentation fault
- d) all of the mentioned

**Answer:-**

- d) all of the mentioned

[Read More Answers.](#)

### Question # 28

Which GDB command can be used to put a break-point at the beginning of the program?

- a) b main
- b) b start
- c) break
- d) none of the mentioned

**Answer:-**

- a) b main

[Read More Answers.](#)

### Question # 29

To put the breakpoint at the current line \_\_\_\_\_ command can be used?

- a) b
- b) break
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 30

We can list all the break-point in GDB by the command:

- a) info break
- b) break all
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**



a) info break

[Read More Answers.](#)

### Question # 31

Which one of the following is not true about GDB?

- a) quit command is used to exit the GDB
- b) kill command is used to stop execution in GDB
- c) if the execution is stopped by kill command then it can not be started again
- d) none of the mentioned

**Answer:-**

- c) if the execution is stopped by kill command then it can not be started again

[Read More Answers.](#)

### Question # 32

While debugging with GDB:

- a) variables can be print
- b) variables can be modify
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 33

GDB command "frame" is used:

- a) to change the stack frames
- b) to check the stack frames only
- c) it is not a valid command
- d) none of the mentioned

**Answer:-**

- a) to change the stack frames

[Read More Answers.](#)

### Question # 34

The GDB command "info local"

- a) displays the list of local variables
- b) value of local values for the current stack frame
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 35

If we have multiple source files, then during the debugging with GDB:

- a) breakpoint can not be set
- b) break point can be set by "break" command with a filename
- c) break point can be set only to makefile
- d) none of the mentioned

**Answer:-**

- b) break point can be set by "break" command with a filename

[Read More Answers.](#)

### Question # 36

What is temporary breakpoint?

- a) it stops the program once
- b) it is removed after one execution of the program
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- c) both (a) and (b)

[Read More Answers.](#)

### Question # 37

In GDB breakpoints can be skipped by the command:

- a) ignore
- b) reject



- c) skip
- d) none of the mentioned

**Answer:-**

- a) ignore

[Read More Answers.](#)

### Question # 38

Which GDB command is used to examine the memory?

- a) x
- b) y
- c) z
- d) none of the mentioned

**Answer:-**

- a) x

[Read More Answers.](#)

### Question # 39

Which one of the following is not true about the GDB?

- a) info register is used to see that what is in the processor registers
- b) processor registers can not be accessed by GDB
- c) first 32 bits of the variable can not be examined
- d) none of the mentioned

**Answer:-**

- c) first 32 bits of the variable can not be examined

[Read More Answers.](#)

### Question # 40

Assemble code of the program can be displayed in GDB by the command:

- a) disassemble
- b) assemble
- c) assembly
- d) none of the mentioned

**Answer:-**

- a) disassemble

[Read More Answers.](#)

### Question # 41

The execution of the program in GDB can be affected by:

- a) arguments
- b) working directory
- c) environment
- d) all of the mentioned

**Answer:-**

- d) all of the mentioned

[Read More Answers.](#)

### Question # 42

What is the output of this program no 9?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 struct sockaddr_in addr;
 int fd;
 fd = socket(AF_INET,SOCK_STREAM,0);
 printf("%dn",fd);
 return 0;
}
```

- a) -1
- b) 3
- c) error
- d) none of the mentioned

**Answer:-**

- c) error

Explanation:

The header file netinet/in.h is required to use the structure sockaddr\_in.

Output:

```
[root@localhost google]# gcc -o san san.c
```



```
san.c: In function 'main':
san.c:7:21: error: storage size of 'addr' isn't known
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 43

What is the output of this program?

```
#include<stdio.h>
#include<stdlib.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd_server, fd_client, len, len_client;
 struct sockaddr_in add_server, add_client;
 char buff[10];
 fd_server = socket(AF_INET,SOCK_STREAM,0);
 if (fd_server == -1){
 perror("fd_sock");
 exit(1);
 }
 len = sizeof(add_server);
 len_client = sizeof(add_client);
 if(bind(fd_server,(struct sockaddr*)&add_server,len) != 0)
 perror("bind");
 fd_client = accept(fd_server,(struct sockaddr*)&add_client,len_client);
 if(fd_client == -1)
 perror("accept");
 read(fd_client,buff,10);
 return 0;
}
```

- a) segmentation fault
- b) error at the time of compilation
- c) syntax error
- d) none of the mentioned

**Answer:-**

- b) error at the time of compilation

Explanation:

The third argument of the accept is the type of pointer.

Output:

```
[root@localhost google]# gcc -o san san.c
san.c: In function 'main':
san.c:26:39: warning: passing argument 3 of 'accept' makes pointer from integer without a cast [enabled by default]
/usr/include/sys/socket.h:214:12: note: expected 'socklen_t * __restrict__' but argument is of type 'int'
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 44

This program can send the request to

```
#include<stdio.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd_client,fd, len;
 struct sockaddr_in add_server;
 fd_client = socket(AF_INET,SOCK_STREAM,0);
 if (fd_client == -1){
 perror("fd_sock");
 exit(1);
 }
 add_server.sin_family = AF_INET;
 add_server.sin_port = ntohs(4001);
 add_server.sin_addr.s_addr = inet_addr("193.39.0.4");
 len = sizeof(add_server);
 fd = connect(fd_client,(struct sockaddr*)&add_server,len);
 if(fd == -1)
 perror("connect");
 return 0;
}
```

- a) the system having IP address 193.39.0.4
- b) any system present in the network
- c) any system of the private network
- d) none of the mentioned



### Answer:-

a) the system having IP address 193.39.0.4

Explanation:

The IP address is mentioned in the proper element of the structure sockaddr\_in

[Read More Answers.](#)

### Question # 45

This program is valid for

```
#include<stdio.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>
```

```
int main()
{
 int fd_client,fd, len;
 struct sockaddr_in add_server;
 fd_client = socket(AF_INET,SOCK_STREAM,0);
 if (fd_client == -1){
 perror("fd_sock");
 exit(1);
 }
 add_server.sin_family = AF_INET;
 add_server.sin_port = ntohs(4001);
 add_server.sin_addr.s_addr = inet_addr("144.29.8.2");
 len = sizeof(add_server);
 fd = connect(fd_client,(struct sockaddr*)&add_server,len);
 return 0;
}
```

- a) IPv4
- b) IPv6
- c) both (a) and (b)
- d) none of the mentioned

### Answer:-

a) IPv4

[Read More Answers.](#)

### Question # 46

What is the output of this program?

```
#include<stdio.h>
#include<sys/socket.h>
int main()
{
 int ret;
 ret = shutdown(0,0);
 printf("%dn",ret);
 return 0;
}
```

- a) 0
- b) -1
- c) can not be determined
- d) none of the mentioned

### Answer:-

b) -1

Explanation:

The shutdown() is used to close a socket and the first argument in shutdown() is socket.

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
-1
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 47

What is the problem with this server program?

```
#include<stdio.h>
#include<stdlib.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>
```

```
int main()
{
 int fd_server, fd_client, len;
 struct sockaddr_in add_server;
 fd_server = socket(AF_INET,SOCK_STREAM,0);
```



```
if (fd_server == -1){
 perror("fd_sock");
 exit(1);
}
add_server.sin_family = AF_INET;
add_server.sin_port = htons(4001);
add_server.sin_addr.s_addr = inet_addr("122.23.1.1");
len = sizeof(add_server);
if(bind(fd_server,(struct sockaddr*)&add_server,len) != 0)
 perror("bind");
if(listen(fd_server,5) != 0)
 perror("listen");
fd_client = accept(fd_server,(struct sockaddr*)&add_server,&len);
if(fd_client == -1)
 perror("accept");
return 0;
}
```

- a) it can not accept the request of any client
- b) it will give the segmentation fault
- c) there is no problem with this program
- d) none of the mentioned

**Answer:-**

- a) it can not accept the request of any client

[Read More Answers.](#)

**Question # 48**

What is the output of this program no 10?

```
#include<stdio.h>
#include<stdlib.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd_server, fd_client, len, len_client;
 struct sockaddr_in add_server;
 fd_server = socket(AF_INET,SOCK_STREAM,0);
 fd_client = accept(fd_server,(struct sockaddr*)&add_server,&len);
 if(fd_client == -1)
 perror("accept");
 if(listen(fd_server,5) != 0)
 perror("listen");
 return 0;
}
```

- a) syntax error
- b) error at the time of compilation
- c) segmentation fault
- d) none of the mentioned

**Answer:-**

- d) none of the mentioned

Explanation:

The listen() must always be used before accept().

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
accept: Invalid argument
[root@localhost google]#
```

[Read More Answers.](#)

**Question # 49**

What is the output of this program no 11?

```
#include<stdio.h>
#include<stdlib.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd_server, fd_client, len, len_client;
 struct sockaddr_in add_server;
 fd_server = socket(AF_INET,SOCK_STREAM,0);
 close(fd_server);
 perror("accept");
 if(listen(fd_server,5) != 0)
 perror("listen");
 fd_client = accept(fd_server,(struct sockaddr*)&add_server,&len);
}
```



```
 if(fd_client == -1)
 return 0;
}
```

- a) syntax error
- b) error at the time of compilation
- c) segmentation fault
- d) none of the mentioned

**Answer:-**

- d) none of the mentioned

Explanation:

The program will not work properly because the file descriptor is not available in the for listen() and accept().

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
accept: Success
listen: Bad file descriptor
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 50

On which system call, this program (process) waits until the server responds?

```
#include<stdio.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>
```

```
int main()
{
 int fd_client,fd, len;
 struct sockaddr_in add_server;
 fd_client = socket(AF_INET,SOCK_STREAM,0);
 if (fd_client == -1){
 perror("fd_sock");
 exit(1);
 }
 add_server.sin_family = AF_INET;
 add_server.sin_port = ntohs(4001);
 add_server.sin_addr.s_addr = inet_addr("127.0.0.1");
 len = sizeof(add_server);
 fd = connect(fd_client,(struct sockaddr*)&add_server,len);
 if(fd == -1)
 perror("connect");
 write(fd,"Hellon",6);
 return 0;
}
```

- a) socket()
- b) connect()
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- a) socket()

[Read More Answers.](#)

### Question # 51

What is the the response of this server for this client if both programs are running on the same system?

```
/*This is server.c*/
#include<stdio.h>
#include<stdlib.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>
```

```
int main()
{
 int fd_server, fd_client, len, len_client;
 struct sockaddr_in add_server, add_client;
 char buff[10];
 fd_server = socket(AF_INET,SOCK_STREAM,0);
 if (fd_server == -1){
 perror("fd_sock");
 exit(1);
 }
 add_server.sin_family = AF_INET;
 add_server.sin_port = htons(4001);
 add_server.sin_addr.s_addr = inet_addr("127.0.0.1");
 len = sizeof(add_server);
 len = sizeof(add_client);
 if(bind(fd_server,(struct sockaddr*)&add_server,len) != 0)
```



```
perror("bind");
if(listen(fd_server,5) != 0)
 perror("listen");
fd_client = accept(fd_server,(struct sockaddr*)&add_client,&len_client);
if(fd_client == -1)
 perror("accept");
read(fd_client,buff,10);
return 0;
}
/*This is the client.c*/
#include<stdio.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd_client,fd, len;
 struct sockaddr_in add_server;
 fd_client = socket(AF_INET,SOCK_STREAM,0);
 if (fd_client == -1){
 perror("fd_sock");
 exit(1);
 }
 add_server.sin_family = AF_INET;
 add_server.sin_port = ntohs(4001);
 add_server.sin_addr.s_addr = inet_addr("127.0.0.1");
 len = sizeof(add_server);
 fd = connect(fd_client,(struct sockaddr*)&add_server,len);
 if(fd == -1)
 perror("connect");
 write(fd,"Hellon",6);
 return 0;
}
```

- a) the server will write back to the client whatever the client will write to the server
- b) the client server communication will not work
- c) the response can not be determined
- d) none of the mentioned

### Answer:-

- a) the server will write back to the client whatever the client will write to the server

Explanation:

The loopback address is used as IP address in both the programs.

[Read More Answers.](#)

### Question # 52

What is the output of this program no 12?

```
#include<stdio.h>

int main()
{
 int fd_socket;
 fd_socket = socket(AF_UNIX,SOCK_STREAM,0);
 printf("%dn",fd_socket);
 return 0;
}
```

- a) -1
- b) 0
- c) any integer value
- d) none of the mentioned

### Answer:-

- d) none of the mentioned

Explanation:

To use socket(), the header files sys/types.h and sys/socket.h are required.

Output:

```
[root@localhost google]# gcc -o san san.c
san.c: In function 'main':
san.c:6:21: error: 'AF_UNIX' undeclared (first use in this function)
san.c:6:21: note: each undeclared identifier is reported only once for each function it appears in
san.c:6:29: error: 'SOCK_STREAM' undeclared (first use in this function)
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 53

In this program, the third argument of the socket() is used for \_\_\_\_\_ potocol.

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
int main()
```





```
{
 int fd_socket;
 if(socket(AF_UNIX,SOCK_STREAM,0) == -1)
 perror("socket");
 return 0;
}
```

- a) TCP/IP
- b) UDP
- c) both (a) and (b)
- d) none of mentioned

**Answer:-**

- a) TCP/IP

[Read More Answers.](#)

### Question # 54

By this program the socket "san\_sock" will create

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>
```

```
int main()
{
 struct sockaddr_un add_server;
 int fd_server;
 fd_server = socket(AF_UNIX,SOCK_STREAM,0);
 if(fd_server == -1)
 perror("socket");
 add_server.sun_family = AF_UNIX;
 strcpy(add_server.sun_path,"san_sock");
 if(bind(fd_server,(struct sockaddr*)&add_server,sizeof(add_server)) != 0)
 perror("bind");
 return 0;
}
```

- a) in the /tmp directory
- b) in the /usr directory
- c) in the present working directory
- d) none of the mentioned

**Answer:-**

- c) in the present working directory

Output:

```
[root@localhost google]# ls
san.c
```

```
[root@localhost google]# gcc -o san san.c
```

```
[root@localhost google]# ./san
```

```
[root@localhost google]# ls
```

```
san san.c san_sock
```

```
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 55

What is the length of the queue for pending connections in this program?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>
```

```
int main()
{
 struct sockaddr_un add_server;
 int fd_server;
 fd_server = socket(AF_UNIX,SOCK_STREAM,0);
 if(fd_server == -1)
 perror("socket");
 add_server.sun_family = AF_UNIX;
 strcpy(add_server.sun_path,"server_sock2");
 if(bind(fd_server,(struct sockaddr*)&add_server,sizeof(add_server)) != 0)
 perror("bind");
 if(listen(fd_server,3) != 0)
 perror("listen");
 return 0;
}
```

- a) 0
- b) 1
- c) 2
- d) 3

**Answer:-**



d) 3

Explanation:

The second argument of listen() specifies the length for the queue for pending connections.

[Read More Answers.](#)

### Question # 56

What is the output of the program no 13?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>

int main()
{
 struct sockaddr_un add_server, add_client;
 int fd_server, fd_client;
 int len;
 char ch;
 fd_server = socket(AF_UNIX,SOCK_STREAM,0);
 if(fd_server == -1)
 perror("socket");
 add_server.sun_family = AF_UNIX;
 strcpy(add_server.sun_path,"san_sock");
 if(bind(fd_server,(struct sockaddr*)&add_server,sizeof(add_server)) != 0)
 perror("bind");
 if(listen(fd_server,3) != 0)
 perror("listen");
 len = sizeof(add_client);
 fd_client = accept(fd_server,(struct sockaddr*)&add_client,&len);
 printf("googlen");
 return 0;
}
```

- a) the program will print the string "google"
- b) the process will remain block
- c) segmentation fault
- d) none of the mentioned

**Answer:-**

- b) the process will remain block

Explanation:

There is no pending request in the queue for listening socket "san\_sock".

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
^Z
[4]+ Stopped ./san
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 57

What is the output of this program?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>

int main()
{
 int fd;
 fd = socket(AF_UNIX,SOCK_STREAM,0);
 printf("%dn",fd);
 return 0;
}
```

- a) 0
- b) 1
- c) 2
- d) 3

**Answer:-**

- d) 3

Explanation:

The socket() returns the lowest available file descriptor and in this program i.e. 3.

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
3
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 58



What is the output of this program no 14?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>
#include<errno.h>

int main()
{
 struct sockaddr_un addr;
 int fd;
 fd = socket(AF_UNIX,SOCK_STREAM,0);
 if (fd == -1)
 perror("socket");
 addr.sun_family = AF_UNIX;
 strcpy(addr.sun_path,"san_sock");
 if (bind(4,(struct sockaddr*)&addr,sizeof(addr)) == -1)
 printf("Sanfoudnryn");
 return 0;
}
```

- a) this program will print the string "google"
- b) this program will not print the string "google"
- c) segmentation fault
- d) none of the mentioned

**Answer:-**

- a) this program will print the string "google"

Explanation:

The first argument of the bind() is not a valid file descriptor in this program.

Output:

```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
Sanfoudnry
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 59

What this program is not able to connect with any client program?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>
```

```
int main()
{
 struct sockaddr_un add_server, add_client;
 int fd_server, fd_client;
 int len;
 char ch;
 fd_server = socket(AF_UNIX,SOCK_STREAM,0);
 if(fd_server == -1)
 perror("socket");
 add_server.sun_family = AF_UNIX;
 strcpy(add_server.sun_path,"san_sock");
 if(bind(fd_server,(struct sockaddr*)&add_server,sizeof(add_server)) != 0)
 perror("bind");
 len = sizeof(add_client);
 fd_client = accept(fd_server,(struct sockaddr*)&add_client,&len);
 printf("googlen");
 return 0;
}
```

- a) the listen() is missing
- b) the connect() is missing
- c) the read() and write() are missing
- d) none of the mentioned

**Answer:-**

- a) the listen() is missing

[Read More Answers.](#)

### Question # 60

What is the output of this program no 15?

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/un.h>
#include<sys/socket.h>
```

```
int main()
{
 struct sockaddr_un add_server, add_client;
```



```
int fd_server, fd_client;
int len;
char ch;
fd_server = socket(AF_UNIX,SOCK_STREAM,0);
if(fd_server == -1)
 perror("socket");
add_server.sun_family = AF_UNIX;
strcpy(add_server.sun_path,"san_sock");
if(bind(fd_server,(struct sockaddr*)&add_server,sizeof(add_server)) != 0)
 perror("bind");
len = sizeof(add_client);
fd_client = connect(fd_server,(struct sockaddr*)&add_client,&len);
printf("googlen");
return 0;
}
a) this program will print the string "google"
b) segmentation fault
c) error
d) none of the mentioned
```

**Answer:-**

c) error

Explanation:

The syntax of the connect() is wrong. connect() should be used in client program only.

Output:

```
[root@localhost google]# gcc -o san san.c
```

```
san.c: In function 'main':
```

```
san.c:20:46: warning: passing argument 3 of 'connect' makes integer from pointer without a cast [enabled by default]
```

```
/usr/include/sys/socket.h:129:12: note: expected 'socklen_t' but argument is of type 'int *'
```

```
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 61

What is the output of this program no 16?

```
#include<stdio.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<sys/socket.h>
#include<errno.h>
```

```
int main()
{
 struct sockaddr_in addr;
 int fd;
 fd = socket(AF_UNIX,SOCK_STREAM,0);
 if (fd == -1)
 perror("socket");
 addr.sun_family = AF_UNIX;
 strcpy(addr.sun_path,"san_sock");
 if (bind(4,(struct sockaddr*)&addr,sizeof(addr)) == -1)
 printf("Sanfoudnryn");
 return 0;
}
```

a) error

b) "google"

c) segmentation fault

d) none of the mentioned

**Answer:-**

a) error

Explanation:

The structure used for AF\_UNIX is sockaddr\_un.

Output:

```
[root@localhost google]# gcc -o san san.c
```

```
san.c: In function 'main':
```

```
san.c:14:6: error: 'struct sockaddr_in' has no member named 'sun_family'
```

```
san.c:15:2: warning: incompatible implicit declaration of built-in function 'strcpy' [enabled by default]
```

```
san.c:15:13: error: 'struct sockaddr_in' has no member named 'sun_path'
```

```
[root@localhost google]#
```

[Read More Answers.](#)

### Question # 62

While debugging with GDB, arguments to the program can be specified by the arguments of \_\_\_\_\_ command.

a) run

b) gdb

c) make

d) none of the mentioned

**Answer:-**

a) run



[Read More Answers.](#)

### Question # 63

Inside GDB, a program may stop because of

- a) a signal
- b) a breakpoint
- c) step command
- d) all of the mentioned

**Answer:-**

- d) all of the mentioned

[Read More Answers.](#)

### Question # 64

Which one of the following is a special breakpoint that stops the program when the value of an expression changes in GDB?

- a) watchpoint
- b) catchpoint
- c) getpoint
- d) none of the mentioned

**Answer:-**

- a) watchpoint

[Read More Answers.](#)

### Question # 65

With the list command, by default GDB prints the \_\_\_\_ source lines.

- a) 20
- b) 10
- c) all
- d) none of the mentioned

**Answer:-**

- b) 10

[Read More Answers.](#)

### Question # 66

In GDB which one of the following allows us to specify a variable in terms of the file or function where it is defined?

- a) ::
- b) @
- b) \$
- d) none of the mentioned

**Answer:-**

- a) ::

[Read More Answers.](#)

### Question # 67

If we want to print the value of a variable in hexadecimal, we have to use "print" command with the option \_\_\_\_ in GDB.

- a) x
- b) h
- c) hex
- d) none of the mentioned

**Answer:-**

- a) x

[Read More Answers.](#)

### Question # 68

Which one of the following variables is used within GDB to hold on to a value and refer to it later?

- a) convenience variables
- b) environment variables
- c) temporary variables
- d) none of the mentioned

**Answer:-**

- a) convenience variables

[Read More Answers.](#)

### Question # 69

In GDB, we can refer to machine register contents, in expressions, as variables with names starting with:

- a) \$
- b) #
- c) !



d) none of the mentioned

**Answer:-**

a) \$

[Read More Answers.](#)

### **Question # 70**

In GDB hardware-dependent information about the floating point unit can be displayed by the command

- a) info float
- b) display float
- c) show float
- d) none of the mentioned

**Answer:-**

a) info float

[Read More Answers.](#)

### **Question # 71**

The result of an expression can be assigned to an environment variable with the command:

- a) assign
- b) set
- c) env
- d) none of the mentioned

**Answer:-**

b) set

[Read More Answers.](#)

### **Question # 72**

Which one of the following GDB command deletes any break-point at the next instruction to be executed in the selected stack frame?

- a) clear
- b) delete
- c) disable
- d) none of the mentioned

**Answer:-**

a) clear

[Read More Answers.](#)

### **Question # 73**

Which one of the following GDB command allows to move from one stack frame to another without printing the frame?

- a) select-frame
- b) frame
- c) frame move
- d) none of the mentioned

**Answer:-**

a) select-frame

[Read More Answers.](#)

### **Question # 74**

The GDB text user interface uses the \_\_\_\_ library to show the source file.

- a) curses
- b) YUI
- c) JUI
- d) none of the mentioned

**Answer:-**

a) curses

[Read More Answers.](#)

### **Question # 75**

By default the GDB automatically executes the command from its:

- a) init files
- b) start files
- c) begin files
- d) none of the mentioned

**Answer:-**

a) init files

[Read More Answers.](#)

### **Question # 76**



The user can define a command for GDB with the command:

- a) define
- b) command
- c) assign
- d) none of the mentioned

**Answer:-**

- a) define

[Read More Answers.](#)

### Question # 77

The GDB command "show output-radix"

- a) sets the default base for numeric display
- b) displays the current default base for numeric display
- c) both (a) and (b)
- d) none of the mentioned

**Answer:-**

- b) displays the current default base for numeric display

[Read More Answers.](#)

### Question # 78

The command "show commands" of GDB

- a) displays the last 10 commands in the command history
- b) displays all commands of the command history
- c) displays all the commands available in GDB
- d) none of the mentioned

**Answer:-**

- a) displays the last 10 commands in the command history

[Read More Answers.](#)

### Question # 79

In GDB, a trace-point can be set by the command

- a) trace
- b) set
- c) break trace
- d) none of the mentioned

**Answer:-**

- a) trace

[Read More Answers.](#)

### Question # 80

Which one of the following command saves the command history of GDB in a file?

- a) history
- b) set history
- c) set history save on
- d) none of the mentioned

**Answer:-**

- c) set history save on

[Read More Answers.](#)

## Operating System Linux Most Popular Interview Topics.

- 1 : [Linux OS Frequently Asked Interview Questions and Answers Guide.](#)
- 2 : [Linux Commands Frequently Asked Interview Questions and Answers Guide.](#)
- 3 : [Linux IPC Frequently Asked Interview Questions and Answers Guide.](#)
- 4 : [Linux General Frequently Asked Interview Questions and Answers Guide.](#)
- 5 : [Linux System Calls Frequently Asked Interview Questions and Answers Guide.](#)
- 6 : [Linux Device Drivers Frequently Asked Interview Questions and Answers Guide.](#)
- 7 : [Linux Socket Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 8 : [Linux Threads Frequently Asked Interview Questions and Answers Guide.](#)
- 9 : [Makefile Frequently Asked Interview Questions and Answers Guide.](#)
- 10 : [Awk Programming Frequently Asked Interview Questions and Answers Guide.](#)



## About Global Guideline.

**Global Guideline** is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of [Interview Questions with Answers](#) and much more. Learn the most common technologies at Global Guideline. We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

\* This PDF was generated from <https://GlobalGuideline.com> at **November 29th, 2023**

\* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and [e-mail us](#) we will fix it.

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

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

Best Of Luck.

Global Guideline Team  
<https://GlobalGuideline.com>  
[Info@globalguideline.com](mailto:Info@globalguideline.com)