

Linux Shutdown & Startup Interview Questions And Answers Guide.



Global Guideline.

<https://globalguideline.com/>



Linux Shutdown & Startup Job Interview Preparation Guide.

Question # 1

Single user mode shell runs as:

- a) Admin user
- b) Root user
- c) normal user
- d) Log user

Answer:-

- b) Root user

[Read More Answers.](#)

Question # 2

Which is the only partition mounted in Single user mode?

- a) boot
- b) usr
- c) root
- d) tmp

Answer:-

- c) root

[Read More Answers.](#)

Question # 3

Which daemon manages the physical memory by moving process from physical memory to swap space when more physical memory is needed?

- a) Sched daemon
- b) Swap daemon
- c) Init daemon
- d) Process daemon

Answer:-

- b) Swap daemon

[Read More Answers.](#)

Question # 4

The process id of init process is:

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

Answer:-

- c) 1

[Read More Answers.](#)

Question # 5

The shell used for Single user mode shell is:

- a) bash
- b) Csh
- c) ksh
- d) sh

Answer:-

- d) sh

[Read More Answers.](#)



Question # 6

Bootstrapping is also known as:

- a) Quick boot
- b) Cold boot
- c) Hot boot
- d) Fast boot

Answer:-

- b) Cold boot

[Read More Answers.](#)

Question # 7

At the end of kernel bootstrap, which process is started?

- a) /etc/init
- b) /etc/sched
- c) /etc/swap
- d) /etc/kernel

Answer:-

- a) /etc/init

[Read More Answers.](#)

Question # 8

The process of starting up a computer is known as:

- a) Boot Loading
- b) Boot Record
- c) Boot Strapping
- d) Booting

Answer:-

- c) Boot Strapping

[Read More Answers.](#)

Question # 9

In this program the allocated memory block can store

```
<pre lang="c" line="1" cssfile="hk1_style">
#include<stdio.h>
#include<stdlib.h>
```

```
int main()
{
    int *ptr;
    ptr = malloc(10);
    return 0;
}
```

- a) int
- b) char
- c) float
- d) all of the mentioned

Answer:-

- d) all of the mentioned

Explanation:

When the malloc() is used without typecasting the default type is void*.

[Read More Answers.](#)

Question # 10

Please tell me output of this program?

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

```
int main()
{
    int *ptr;
    ptr = (int *)calloc(1,sizeof(int));
    if (ptr != 0)
        printf("%dn", *ptr);
    return 0;
}
```

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

Answer:-

- a) 0

Explanation:



The memory allocated by calloc() contains 0 until process does not make any change to it.

Output:

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

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

```
0
```

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

[Read More Answers.](#)

Question # 11

Program given below will allocate the memory of ___ bytes for pointer "ptr".

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

int main()
{
    int *ptr;
    ptr = (int*)malloc(sizeof(int)*4);
    ptr = realloc(ptr,sizeof(int)*2);
    return 0;
}
```

- a) 2
- b) 4
- c) 8
- d) none of the mentioned

Answer:-

c) 8

Explanation:

We can also use the realloc() to make memory block smaller.

[Read More Answers.](#)

Question # 12

In which condition this program will print the string "google"?

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

```
int main()
{
    int *ptr;
    ptr = (int *)malloc(sizeof(int)*10);
    if (ptr == NULL)
        printf("googlen");
    return 0;
}
```

- a) if the memory could not be allocated to the pointer "ptr"
- b) if the memory has been allocated to the pointer "ptr" successfully
- c) it will never print
- d) none of the mentioned

Answer:-

a) if the memory could not be allocated to the pointer "ptr"

Explanation:

The malloc() returns NULL when the memory is not allocated.

[Read More Answers.](#)

Question # 13

What is the output of this program?

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

```
int main()
{
    int ret;
    int *ptr;
    ptr = (int *)malloc(sizeof(int)*10);
    free(ptr);
    free(ptr);
    return 0;
}
```

- a) it will print nothing
- b) it will give segmentation fault
- c) undefined behaviour
- d) none of the mentioned

Answer:-

c) undefined behaviour

Explanation:

If the free() has already called before, undefined behaviour occurs.

Output:



```
[root@localhost google]# gcc -o san san.c
[root@localhost google]# ./san
*** glibc detected *** ./san: double free or corruption (fasttop): 0x08f1b008 ***
===== Backtrace: =====
/lib/libc.so.6[0x4a6489f2]
./san[0x8048425]
/lib/libc.so.6(__libc_start_main+0xf3)[0x4a5e96b3]
./san[0x8048361]
===== Memory map: =====
08048000-08049000 r-xp 00000000 fd:01 394194 /home/google/san
08049000-0804a000 rw-p 00000000 fd:01 394194 /home/google/san
08f1b000-08f3c000 rw-p 00000000 00:00 0 [heap]
4a5ab000-4a5cc000 r-xp 00000000 fd:01 785334 /lib/ld-2.14.90.so
4a5cc000-4a5cd000 r-p 00020000 fd:01 785334 /lib/ld-2.14.90.so
4a5cd000-4a5ce000 rw-p 00021000 fd:01 785334 /lib/ld-2.14.90.so
4a5d0000-4a77a000 r-xp 00000000 fd:01 789110 /lib/libc-2.14.90.so
4a77a000-4a77b000 -p 001aa000 fd:01 789110 /lib/libc-2.14.90.so
4a77b000-4a77d000 r-p 001aa000 fd:01 789110 /lib/libc-2.14.90.so
4a77d000-4a77e000 rw-p 001ac000 fd:01 789110 /lib/libc-2.14.90.so
4a77e000-4a781000 rw-p 00000000 00:00 0
4a7e0000-4a7fc000 r-xp 00000000 fd:01 789128 /lib/libgcc_s-4.6.2-20111027.so.1
4a7fc000-4a7fd000 rw-p 0001b000 fd:01 789128 /lib/libgcc_s-4.6.2-20111027.so.1
b76f4000-b76f5000 rw-p 00000000 00:00 0
b770d000-b770f000 rw-p 00000000 00:00 0
b770f000-b7710000 r-xp 00000000 00:00 0 [vdso]
bfc0a000-bfc2b000 rw-p 00000000 00:00 0 [stack]
Aborted (core dumped)
[root@localhost google]#
```

[Read More Answers.](#)

Question # 14

What is the output of this program?

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

int main()
{
    int *ptr1;
    while(1){
        ptr1 = malloc(1024*1024);
        if(ptr1 == 0)
            break;
        sleep(1);
        printf("googlen");
        free(ptr1);
    }
    return 0;
}
```

- a) it will print "google" until the process has been stoped by any signal
- b) it will print nothing
- c) segmentation fault
- d) none of the mentioned

Answer:-

- a) it will print "google" until the process has been stoped by any signal

Explanation:

None.

Output:

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

[Read More Answers.](#)

Question # 15

What is the output of this program?

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

int main()
{
    int *ptr1, *ptr2;
    ptr1 = malloc(4);
    *ptr1 = 10;
```



```
*ptr2 = free(ptr1);  
printf("%dn",*ptr2);  
return 0;  
}
```

- a) 10
- b) it will print the address stored in ptr1
- c) it will print the address stored in ptr2
- d) it will give an error

Answer:-

- d) it will give an error

Explanation:

The free() function returns no value.

Output:

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

[Read More Answers.](#)

Question # 16

Do you have any idea what is the output of this program?

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

```
int main()  
{  
    int *ptr;  
    *ptr = 10;  
    *ptr = 20;  
    printf("%dn",*ptr);  
    return 0;  
}
```

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

Answer:-

- c) segmentation fault

Explanation:

The segmentation fault occurs because memory for the pointer has not been allocated in this program.

Output:

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

[Read More Answers.](#)

Question # 17

What is the output of this program?

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

```
int main()  
{  
    int *ptr;  
    ptr = (int *)calloc(1,sizeof(int));  
    *ptr = 10;  
    printf("%dn",*ptr);  
    return 0;  
}
```

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

Answer:-

- d) none of the mentioned

Explanation:

This program will give an error because calloc() requires the header file stdlib.h.

Output:

```
[root@localhost google]# gcc -o san san.c  
san.c: In function 'main':  
san.c:6:15: warning: incompatible implicit declaration of built-in function 'calloc' [enabled by default]  
[root@localhost google]#
```

[Read More Answers.](#)

Question # 18

On Linux, initrd is a file:



- a) containing root file-system required during bootup
- b) Contains only scripts to be executed during bootup
- c) Contains root-file system and drivers required to be preloaded during bootup
- d) None of the above

Answer:-

- c) Contains root-file system and drivers required to be preloaded during bootup

[Read More Answers.](#)

Question # 19

Which of the following is not a valid run-level?

- a) S
- b) 0
- c) 8
- d) 1

Answer:-

- c) 8

[Read More Answers.](#)

Question # 20

Which file is read by init to get the default run-level?

- a) /etc/profile
- b) /etc/init
- c) /etc/boot
- d) /etc/inittab

Answer:-

- d) /etc/inittab

[Read More Answers.](#)

Question # 21

Which is loaded into memory when system is booted?

- a) Kernel
- b) Shell
- c) Commands
- d) Script

Answer:-

- a) Kernel

[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

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