

MODULE 3 ASSESSMENTS

1. Which signals are triggered, when the following actions are performed.

- a. user press ctrl+C
Answer: SIGINT signal (2)
- b. kill() system call is invoked
Answer: SIGTERM signal (15)
- c. CPU tried to execute an illegal instruction
Answer: SIGILL signal (4)
- d. When the program access the unassigned memory
Answer: SIGSEGV signal (11)

2. List the gdb command for the following operations

- a. To run the current executable file
Answer: >run or >run arg
- b. To create breakpoints at
Answer: >break <line_number> or >break <function_name>
- c. To resume execution once after breakpoint
Answer: >continue
- d. To clear break point created for a function
Answer: >clear [line number] or >clear [FUNCTION_NAME]
- e. Print the parameters of the function in the backtrace
Answer: >backtrace
>frame <frame_number>
>info args

3. Guess the output for the following program.

```
#include<stdio.h>
#include<unistd.h>
int main() {
    if (fork() && (!fork())) {
        if (fork() || fork()) {
            fork();
        }
    }
    printf("2 ");
    return 0;
}
```

Answer: 2 2 2 2 2 2 2

MODULE 3 ASSESSMENTS

4. Guess the output for the following program.

```
#include<stdio.h>
#include<unistd.h>
int main() {
    if (fork()) {
        if (!fork()) {
            fork();
            printf("1 ");
        }
        else {
            printf("2 ");
        }
    }
    else {
        printf("3 ");
    }
    printf("4 ");
    return 0;
}
```

Answer: 2 4,
2 4 3 4,
3 4 2 4 1 4 1 4,
2 4 1 4 1 4 3 4,
2 4 3 4 1 4 1 4,
2 4 1 4 3 4 1 4 ,etc.

This program provides different outputs on execution because of it's concurrent nature of process execution.Hence,we get a variety of outputs.

6. How to avoid Race conditions and deadlocks?

Answer:

To avoid race conditions

- Use Synchronization Mechanisms like locks,mutexes and semaphores.
- Protect critical sections of code using locks and mutexes.

To avoid deadlocks

- Avoid circular dependencies
- Use time out mechanisms
- Implement deadlock prevention mechanisms like banker's algorithm.

7. What is the difference between exec and fork ?

Answer:

EXEC()

- The exec() command is used to replace the current process image with a new process.when exec() is called,the new program starts executing from its main() function or entry point.

FORK()

- The fork() is used to create a new process by duplicating the calling process.The new process is referred to as the child process, and the original process is referred to as the parent process.

MODULE 3 ASSESSMENTS

8. What is the difference between process and threads.

Answer:

Process

- A process is a program in execution. It has its own memory space, which includes data and resources.
- Processes are isolated from each other and they cannot directly access each others memory space.
- Inter-process communication mechanisms such as pipes and shared memory are used for communication between processes.
- Processes are heavyweight in terms of resource consumption.

Thread

- A thread is a smallest unit of execution within a process. Multiple threads can exist within a single process and they can share same memory space.
- Threads within the same process can communicate directly with each other through shared memory.
- Threads share same resources of a process which can lead to synchronization problems.
- Threads are more efficient in terms of resource consumption as they share resources.