

QUIZ 7 – Signals

Q1. Write the code snippet for all the possible ways by which signals can be ignored.

Q2. Look at the code below. What will be the output when Control+C is pressed during infinite while loop.

```
handler1() {printf("Handler1 is executed by %d\n", getpid());}
handler2() {printf("Handler2 is executed by %d\n", getpid());}
main()
{
    signal(SIGINT, handler1);
    fork();
    while(1);
}
```

Q3. Look at the code below. What will be the output when Control+C is pressed during infinite while loop.

```
handler1() {printf("Handler1 is executed by %d\n", getpid());}
handler2() {printf("Handler2 is executed by %d\n", getpid());}
main()
{
    signal(SIGINT, handler1);
    if (fork() == 0)
        signal(SIGINT, handler2);
    while(1);
}
```

Q4. Let's say parent process (pid=10) has 2 children (pid=11 and pid=12). Write a single C code for implementing the following scenarios. Note: Use Alarm for all the waiting instead of using sleep.

a) Parent to suspend the second child (pid=12) between 2 to 5 seconds of program execution time.

b) Parent kills the first child (pid=11) at 4th second.

c) Parent send signal of Control+C to both the live processes (parent pid=10 and second child pid=12) at 7th second. Detect the death of the second child using SIGCHLD signal handler.

Q5. Write a C code for the following situations.

a) Parent process has one child process and both are using a single pipe for bidirectional communication using SIGUSR1 and SIGUSR2.

b) When a process (parent or child) wants to send data to the other process (child or parent) it sends SIGUSR1. Upon receipt of SIGUSR1 a process knows that other process is going to use the pipe to send data so it will wait for data from pipe by issuing read() in SIGUSR1 handler.

c) When a process (parent or child) finished sending data it will send SIGUSR2 signal to the other process (child or parent) indicating transmission is complete.

You can test your code by sending one line of file1.txt from parent to child and one line of file2.txt from child to parent in alternately.