NAME : MOITRISH MAITY DEPT : CSE YEAR : 3RD CLASS ROLL : 48

ASSIGNMENT ON SIGNALS

1. Use the kill command to find the number of signals available on your system and then write a program that ignores all of them.

CODE :

```
~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
~$
```

CODE:

```
#include <signal.h>
sigset_t mask;
sigfillset(&mask);
sigprocmask(SIG_SETMASK, &mask, NULL);
```

Questionnaire

1. **In what ways can a process behave when it receives a signal? What is special about the SIGSTOP and SIGKILL signals?**

Ans : Normally when a process receives a signal, the process gets terminated. And mainly to terminate or stop the process the signals are generated by keyboard, hardware, etc. But there are ways by which a process can be treated when it receives a signal.

Terminate: This is the most common way, because after receiving the signal of termination the process execution is terminated and the process is killed.

Ignore: When a process receives a signal, it will ignore the signal, so there is no impact to the process after receiving a signal.

Restore: The process will go back to its previous state after receiving the signal. Suppose a process is stopped and after receiving the signal it starts running. It means that a process is paused but when the signal is received it starts running.

Respond: A process receives a signal and after that a function is executed for a specific task. This can also be done for a process.

SIGSTOP and **SIGKILL** are the only two signals that cannot be caught, blocked, or ignored in a process.