Assignment 2

Assignment no 2(a)

Information: One case (out of three cases explore own-self for the other two, theory will be covered in class) for handling signal is: a process executes a particular (user) function on receipt of the signal.

Problem: Write a C program by using signal() system call to handle the reception of SIGINT signal by executing a particular(user) function, which function is responsible for creating a child process by using fork( ) system call and then you have to display the PROCESS ID and PARENT PROCESS ID from the parent process as well as from the child process.

/* You need to put proper explanatory comment in your program to demonstrate the purpose and why you have used the C statements and system calls */

Hints:

* For generating, SIGINT (SIGINT is a keyboard interrupt signal) signal, you have to press Ctrl+C. So, by default pressing Ctrl+C in a running program leads to the termination of the running process. But, your program should provide a way to handle the keyboard interrupt through signal() system call by executing user defined function as mentioned above.
* To know more about, see **signal(2)** man page (command: "*man 2 signal*") and refer W.R. Steven Book,Vol-2.

Assignment no 2(b)

Write a C program which will take the Process ID and signal ID as input to demonstrate the use of kill( ) system call.

/* You need to put proper explanatory comment in your program to demonstrate the purpose and why you have used the system calls */

Hints:

* For demonstrating so, you modify the assignment 2(a) to handle each and every signal (as much possible as). Run command "***kill –l***", to know about signal type and ID. Now run the modified assignment 2(a).
* Again from another terminal, run the assignment 2(b) which will take the Process ID of the modified assignment 2(a) and any valid signal value as input.
* Your signal handler function of the modified assignment 2(b) should be able to display the signal ID of the generated signal.
* To know more about kill( ), see **kill(2)** man page and for signal type, value, action and comment, see **signal(7)** man page and refer to W.R. Steven Book,Vol-2**.**

Assignment no 2(c)

Write a C program to create a user level thread using system call pthread_create( ) and assign the thread to display the "HELLO WORLD" . Use pthread_exit() in your program (if possible) for terminating the thread.

/* You need to put proper explanatory comment in your program to demonstrate the purpose and why you have used the system calls */

Hints:

* To know more about pthread_create( ), see pthread_create (3) man page and to know more about pthread_exit( ), see pthread_exit (3).