Linux Assignment - 3

1. Write a program and do the following :
   using sigaction API, handle SIGINT, SIGTERM, SIGQUIT, SIGSTOP, SIGTSTP and SIGKILL. Install dummy handlers for the mentioned signals Using sigaction() system call API !!!

   Test your processes using signals and see if they behave as expected.

2. Try and mask as many signals as possible in your process and test whether they are masked and are not disturbed by signals – you may use sigsuspend() to block your code and test your code .
   observe and comment on the result.

3. Try to kill init process (with pid 1) from your command line(using kill command) or using kill() system call inside one of your processes.
   what is the result ? comment on the same.

Linux Assignment 3

4. launch a pdf viewer and use kill command to generate a SIGSTOP signal to the pdf viewer's process – what happens ??

   Next, using kill command, generate a SIGCONT signal to the same process. Observe what happens and comment !!!

5. modify example2.c such that sigsuspend() system call APIs are replaced with sigtimedwait() - set the timeout field to NULL – meaning, wait for ever !!!

   Test the working of the new code – does it work as expected – if so, what Is difference with respect to the code using sigsuspend() !!! refer to lecture notes and also manual page of sigtimewaid(), for more details !!!

Linux Assignment – 3 …..........

6. Create 5 children from a common parent and you must clean-up all the children in an explicit SIGCHLD handler installed in the parent process. Please note that parent process should not make any wait() or waitpid() calls in the main body of the process; You must use waitpid()(not wait()) in the signal signal handler; as discussed in the class, you must strictly follow the rules for writing the signal handler

   Also, you must ensure that sigaction() used to install the signal handler with all signals masked in the signal handler.

   Also,you must ensure that the parent must not exit until all the children terminated(ZOMBIE_STATE) and are cleaned-up(DEAD_STATE)
   You are free to use sigsuspend() system call in the main body of the process to wait until the above clean-up is completed. Refer to msg_server.c for a sample signal handler that is used to clean up zombie children processes In the signal handler of the parent process !!!

Note: you can use the sample signal handler, but do not just blindly copy it -
In addition, you must sigsuspend() at appropriate point in main()
of parent process !!!