Linux Assignment – 2_1

1. Try to create as many children as possible from a process. After creating as many children as possible, clean-up the children using waitpid() before calling exit(), in the parent process !!!
   When you are creating as many processes using fork(), check whether fork() returns -1 ; this means, system is no longer able to create further processes – once you get this error , you must break the infinite while loop for creating processes and enter the loop for cleaning up all the children processes using waitpid() system call API !!!
   You must print the total no. of possible processes created and also the error message generated by the system to the terminal screen.
   What are your observations ?

2. On your systems, after fork, child executes first. Use sched_yield() system call to switch to the parent and execute the parent first.

Linux Assignment - 2_1..................

3. 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 sys_p_server.c for a sample signal handler that is used to clean up zombie children processes In the signal handler of the parent process !!!

Linux Assignment - 2_1.................

4. write a program and do the following :
   using sigaction API, handle SIGINT, SIGTERM, SIGQUIT, SIGSTOP, SIGTSTP and SIGKILL. you are free to what you want to do in your respective handler.
   Test your processes and see if they behave as expected.
5. Try and block as many signals as possible in your process and test whether they are blocked. observe and comment on the result.
6. 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.