Linux system programming assignment – 2

1. Create 5 children processes from a common parent and ensure that the parent terminates after cleaning all the terminated children using waitpid(). The waitpid() must be called after all the children are created and the parent has completed its work real work,if any.

2. Create 5 processes but not from the common parent. Meaning, each child creates a new process. clean-up the children using waitpid().

3. Create a child process and compile any of programs in the child process. Parent process must use waitpid() to collect the termination status of the child process and print a message accordingly to the user.

4. Create 5 processes from a common parent and ensure that the parent terminates after cleaning all the terminated children using waitpid(). The waitpid() must be called after all the children are created and the parent has completed its work real work,if any; in addition, you must compile 5 different program files in children processes to generate their respective object files ; the parent process must use waitpid() to collect the termination status of children processes and link all the object files to generate the final, linked program/application.

Linux system programming assignment – 2

5. use the example1.c sample program to find the following:
Note: first, you must understand the working of example1.c thoroughly – meaning, overall working of the program and individual system call APIs

a) without accessing the pages/page-frames, what is the largest size process that you cancreate and load in the system
b) how many such large processes can you create and load – again without accessing the pages/page-frames
c) now, signal the process to access all the pages/page-frames and find what is the largest process you can create and load
d) using the condition in c), how many such processes can be created and loaded