**OS ASSIGNMENT 2**

1. Write a program that creates a child process using fork (). The child prints its parent’s name, Parent ID and Own ID while the parent waits for the signal from the child process. Parent sets an alarm 10 seconds after the Child termination.
2. Write a program that creates a child process using fork (). The child displays its ID and parent ID. The Parent displays its own ID and its child ID.

The output should be of the form:

I am parent my ID is=1234 and my Child ID is=4567

I am Child my ID is=4567 and my Parent ID is=1234

1. Write a C++ program in which a parent process creates a child process using fork () system call. The child takes a number as input and checks if it is a palindrome number or not and then prints a message accordingly. While the parent process waits for the child process to terminate. After the child exits, the parent displays the exit status returned by the child and calls exit ().
2. Write a program that spawns “n” no. of children from the parent process, then waits for their completion and behaves differently after the completion of each child process (Also display the exit codes and IDs of each child on their completion in the parent process).

“n” is a number that will be taken as input from the user in the parent process.

1. Write a program that uses fork () to create a child process. The child then uses exec () to copy two files whose names should entered by the user as input. While the parent waits for the child to terminate and call exit () after child termination.
2. Write a program for inter-process communication among three processes using pipes. In which, one process is Parent which has two child processes. Parent process gets a number from the user. First child reads the number from the pipe and calculates the factorial. Similarly, the second child reads the number from pipe and calculates the square of that number. In the end, the Parent process read the output from the pipe and display it.
3. Write a program that spawns “n” (a number input from user) no. of children from the parent process. Parent creates a process group and adds child processes in the group in a way that every new child process is assigned a new group ID. Then parent waits for all the child processes to finish and prints appropriate messages (with group and process IDs) to display the program flow.