LAB ASSIGNMENT 5: On Processes in Linux

Deadline: 6th September 2015, 5:00 pm

This assignment consists of two tasks. You have execute both these tasks on **Linux** in **C.**

Task 1: In this task, you have to develop a program called lab1Task5-<*xxxx*>.c, where <*xxxx*> is your roll number. This program will take as input two positive integers *max* and *num* from the user where *num* ≤ *max*. Assume that both *num* and *max* are small (preferably less than 50). Then, it creates two child processes *c1* and *c2*. Let the parent process be denoted as *p1*.

Process *c1* **randomly** chooses *num* numbers between 1 – *max*. It displays these numbers on the screen and then computes the **product** of these numbers which is also displayed on the screen. We denote the product as *prod*.

Along the same lines, process c2 choose *num* numbers between 1 – *max*. It displays these numbers on the screen and then computes the **sum** of these numbers which is also displayed on the screen. Let us denote the product as *sum*.

The parent process waits for both these children to terminate. It then finally computes the difference of *prod* and *sum* (which could also be negative). A sample input and output is as follows:

$ lab1Task5-<*xxxx*>.o
num: 3
max: 10

Output of c1: 7 8 9 prod: 504

Output of c2: 3 5 7 sum: 15

Output of p: 489

Note: Since the processes *c1* and *c2* are executing concurrently, the order of display of these processes can be different.

Task 2: Develop a program lab5Task2-<*xxxx*>.c which takes a valid *pid* and display the list of all its ancestors. If the *pid* is not valid, then it should output *pid* is not valid.
pstree is a shell command that takes a pid and displays the tree of processes. Use this shell command to achieve this task.

Deliverables: You have to submit the following

- A report describing your implementation
- Source codes of lab5Task1-<*xxxx*>.c and lab5Task2-<*xxxx*>.c

The report should describe in detail how you completed Tasks 1 and 2. Provide all CODE that you wrote in this report. Make sure that your report is technically sound and readable.

Submission deadline: 6th September 2015, 5:00 pm
Evaluation Criteria:

Task	**Report Description**	**Code Execution**	**Total**
Task 1	35	25	60
Task 2	25	15	4