BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI,
K. K. BIRLA Goa Campus,
I SEMESTER 2021 – 2022
Advanced Operating Systems (CS G623)
Assignment 2, Due date: 04/10/2021 (11:59 P.M)
Marks: 15M [5%]

Instructions: Please upload your completed assignment in Quanta.

This is an individual assignment. Please see section 4b of handout for Malpractice regulations.

The programming assignments will be graded according to the following criteria

- *Completeness; does your program implement the whole assignment?*
- *Correctness; does your program provide the right output?*
- *Efficiency; have you chosen appropriate algorithms and data structures for the problem?*
- *Programming style (including documentation and program organization); is the program well designed and easy to understand?*
- *Individual Viva.*

Submit your assignment as a tar.gz file.

DO NOT FORGET to submit a README file (text only) with following contents.

General README instructions

1. *Your name. If you interacted significantly with others, indicate this as well.*
2. *A list of all files in the directory and a short description of each.*
3. *HOW TO COMPILE your program (You need to submit MAKEFILE for each question).*
4. *HOW TO USE (execute) your program.*
5. *A description of the structure of your program.*
6. *In case you have not completed the assignment, you should mention in significant detail:*
    a. *What you have and have not done,*
    b. *Why you did not manage to complete your assignment (e.g., greatest difficulties)*
7. *This will allow us to give you partial credit for the things you have completed.*
8. *Document any bugs in your program that you know of. Run-time errors will cost you less penalty if you document them and you show that you know their cause. Also describe what you would have done to correct them, if you had more time to work on your project.*

NB: Late submission will not be entertained.

Question #1

Write a program to create a process tree as shown in Figure #1. The input value through command line includes how many children a process has to spawn if process id is odd and if the process id is even. The third argument specifies the level of the tree. Figure shows the resultant tree if the command line argument is 2 (even pid), 1 (odd pid), and 3(level). Print the process identifiers in in-order traversal form. The program should create simultaneously executing processes in the same level as quickly as possible before child's execution starts. Each process should print its process identifier and its parent process identifier by using the function getpid( ) and getppid( ) respectively. Your program should not create an orphan process while execution. The parent process should also print the exit status of its child processes.

Output of a run:

./question1 1 2 3

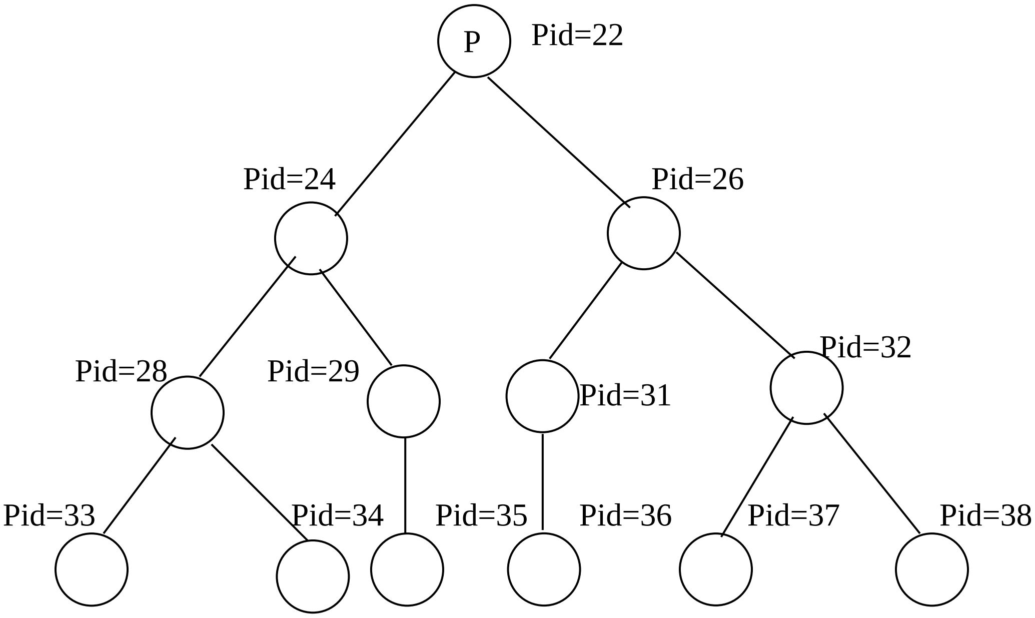

Figure #1