

Assignment – II
Operating Systems Laboratory

1. Create two child processes: X and Y.
 - a. Each child process performs 10 iterations. The child process just displays its name/id and the current iteration number, and sleeps for some random amount of time. Adjust the sleep duration of the processes to have a different output (i.e. another interleaving of processes' traces).
 - b. X is not allowed to start iteration i before process Y has terminated its own iteration $(i-1)$. Modify the program to implement this synchronization using a semaphore.
2. Write program/s that implement the following using IPC mechanisms. Choose an appropriate IPC mechanism for each.

Your choice is restricted to Pipes, FIFOs, and Message Queues. Do not use shared memory and semaphores. Cover at least two IPC mechanisms.

 - a. Telephonic conversation (between a caller and a receiver)
 - b. Conference call (first caller calls 2nd and then sets up calls with others (at least two))
3. Write a program for p -producer c -consumer problem with both p and $c \geq 1$. A shared circular buffer that can hold 25 integers is to be used. Each of the producer processes stores the numbers 1 to 60 in the buffer one by one and then exits. Each of the consumer processes reads the numbers from the buffer and adds them to a shared variable TOTAL (initialized to 0). Though any consumer process can read any of the numbers in the buffer, the only constraint is: every number written by some producer should be read *exactly once* by *exactly one* of the consumers.

The program will read in the values of p and c from the user, and fork p producers and c consumers. After all the producers and consumers have finished (the consumers exit after all the data produced by all the producers have been read), the parent process prints the value of TOTAL. Test the program with different values of p and c and different number of items produced by the p producers. (Hint: Any producer can produce any number of item in a particular execution of the program.)
4. Write a program that implements the following using proper synchronization mechanism.

Readers-Writers Problem

“A database is shared among some processes, a few of which have the right to write also. Any writer has priority over readers. The system will belong to the writer/s whenever any writer is ready.”

Last date of submission of Assignment II: For respective groups: 16th and 17th October, 2017
And Lab Exam (2nd): For respective groups: 23th and 24th October, 2017