IT308: Operating System (Winter 2012)
Programming Assignment #2
Inter Process Communication System (IPC System)
Due: February 24@17hrs.

Objective: Implement an IPC system that lets processes communicate with one another by using *inter-process message* according to the following schemes:

A process wishing to send a message indicate the name of the process to which the message should be delivered and text of the message. The IPC controller delivers this message to the destination process immediately if that process is waiting for a message; otherwise, the message remains *pending*.

A process wishing to receive a message **does not** indicate the name of the process from which it wishes to receive a message. The **IPC** controller should block the process if no messages sent to it are still pending. Otherwise, the **IPC** controller delivers it the earliest that was sent to it which is still pending, i.e. the **IPC** controller gives it messages sent by other process in an **FCFS** manner.

You are required to code the **IPC** controller and *four* user processes as per the following specification:

1. Messages are of standard length only.

2. **IPC** controller has limited amount of buffer space available, say, enough buffer space to accommodate 20 messages. The IPC controller must make the best possible use of this space.

3. Messages will handed over to a receiver process in an **FCFS** manner.

4. Operation of each user process is governed by a separate commands file. The user process will read the next command in the file, which will be either a *send* or a *receive* command with relevant details, and act on it. You may assume your own format for the commands.

5. After performing a send or receive request made by a process, the **IPC** controller will print report line –

   Message sent/received: <Sender id>, <receiver id>, <message>

6. **IPC controller will detect deadlocks if they arise.** (A deadlock situation for a group of processes is situation in which each of the process in the group is blocked for an event that cannot occur. This way, processes that are in deadlock will remain blocked indefinitely.)

Points/Suggestions for implementation

1. Interpret the word ‘process’ in the above specification as a ‘thread’
2. Send/Receive actions performed by a user process are dictated by a commands file.
3. Each *send* or *receive* command will amount to a call to the **IPC**, made by the user process. Identity of the process making the call will be required by the **IPC**. You must make appropriate provision for it.
4. The process of *send* and *receive* commands for each process should be so chosen as to demonstrate working of the system in different situations. During viva with TA, you will be required to demonstrate for a given situation.
5. You should shut down the **IPC** controller and the processes after some fixed number of messages have been exchanged.
6. **Your implementation must have appropriate synchronization and minimum concurrence. You must think of all possible deadlock situations.**
7. It would be useful to
    a. Analyze the problem specification and understand working of the IPC controller.
    b. Analyze the synchronization requirements.
    c. Decide how the synchronization should be performed.
    d. Code your solution and test it.
8. For this programming assignment on Posix threads (pthreads) in the concurrent programming assignment pl. refer following URL https://computing.llnl.gov/tutorials/pthreads

Deliverable:

You should hand your program, README, and Makefile as a part of this assignment electronically. You should include a copy of the code. These should be clearly labeled (commented) as to what they do. You **must** answer the **step 7** in reference to your code.

You will upload your deliverable with to FTP server. Details of ftp server are given below. You can start to upload deliverable 2 days in prior to deadline, and not early to make ftp server usage more consistent.

There is four folder in IT_308_2012 (This is share folder) batch1, batch2, batch3, batch4 each folder 32 student. Must have your folder under relevant batch with name *your.sid_pa2* for final evaluation. \\10.100.56.24\IT_308_2012