

## Threads Assignment 1

1. repeat the producer-consumer problem of process model using threads; use the process address space's heap area for buffer slots; use unnamed posix semaphores(thread semphores) for counting semaphores and mutexes for protecting any critical sections.  
You must repeat the producer – consumer assignment (last problem of assignment 3).
2. repeat the producer-consumer problem of process model using threads; use the process address space's heap-area for buffer slots; use condition variables in the place of counting semaphores and mutexes for protecting any critical sections. You must repeat the producer – consumer assignment (last problem of assignment 3).
3. prove practically that error checking type of mutexes do what they claim ;
  - a) that is they avoid dead-lock in the same thread, if the thread locks a mutex that it has already locked
  - b) only the thread that locks can unlock the mutex;
  - c) verify that fast/normal mutex type does not support a) and b).

## Threads assignment 1 ...continued

4. Write a multi-threaded server application that does the following :  
Look into client-server sample code that was discussed in the class – meaning, `sys_p_server_fifo.c`. Use the sample code to write a server that will handle multiple clients concurrently using a new thread for each client's new request – also, write several clients that will communicate with the server process as discussed in the class. In this assignment, there will be one named fifo used by all clients for sending request messages to the server. This is known as server's named fifo. In addition, there will be one named fifo for each client for receiving its requested data. Server's named fifo will be created by the server. Each client will create their own named fifo for receiving their requested data. In addition, you must take care of necessary signal handling. In addition, you must test using multiple clients requesting service from the server simultaneously – it is better to use a multi processor configuration for this testing – say, test with 10 clients at a time.

Threads assignment - 1 ...continued

5. Write a multi-threaded server application that does the following :

Note: in this case, you must use a server message queue instead of server fifo – otherwise, the same as previous assignment

Look into client-server sample code that was discussed in the class – meaning, `sys_p_server_fifo.c`. Use the sample code to write a server that will handle multiple clients concurrently using a new thread for each client's new request – also, write several clients that will communicate with the server process as discussed in the class. In this assignment, there will be one message queue used by all clients for sending request messages to the server. This is known as server's message queue. In addition, there will be one named fifo for each client for receiving its requested data. Server message queue will be created by the server. Each client will create their own named fifo for receiving their requested data. In addition, you must take care of necessary signal handling. In addition, you must test using multiple clients requesting service from the server simultaneously – it is better to use a multi processor configuration for this testing – say, test with 10 clients at a time.

---