National University of Computer and Emerging Sciences



**Laboratory Manual**

***(Operating Systems)***

|  |  |
| --- | --- |
| Semester | Spring 2018 |

Department of Computer Science

FAST-NU, Lahore

**Question # 1**

Producer has to write N values in buffer, when producer has written N values, consumer has to read all N values. Following 2 threads shared that buffer. Both run simultaneously. Do synchronization with given variables.

Sem1 = N , Sem2 = 1, Mutex = 1 (2 semaphores and 1 is mutex)

Char buffer[n] (shared variable)

**Producer** **Consumer**

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

writeBuffer(buffer) readBuffer(buffer)

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_

**Question # 2**

There are exactly 3 threads generate string a, b and c in an arbitrary order. In an absence of any synchronization mechanism there will be no order in generation of a, b and c. In the form of regular expression the string (a | b | c)\* {\* means many times a character an occur, | means or, so different combinations can be aaaaaaa… , bbbbbbbbbbb… , ccccc… }. Synchronize threads using semaphore in such a way that your printed string will be (cba)\* {\* means many times cba can occur, so different combinations will be cbacbacbacb….}.

Note you are not allowed to add or delete any cout statement

|  |  |  |
| --- | --- | --- |
| //thread 1  While(1)  {    Cout << ‘a’;  } | //thread 1  While(1)  {  Cout << ‘b’;  } | //thread 1  While(1)  {  Cout << ‘c’;  } |