****

**UE18CS305 – Operating Systems Laboratory**

**WEEK 7: Process Synchronization** **Due** **Date: 23/08/2021**

**OBJECTIVE: Process Synchronization**

**PROGRAM 1:** Write a C Program to simulate race condition in Producer Consumer Problem Implement a main program that creates two threads: producer and consumer threads which execute producer and consumer functions respectively. The producer should produce an item and update the buffer. The consumer should consume an item and update the buffer. You can use bounded buffer and both the producer and consumer threads can be infinite loops. Show how race condition occurs between producer and consumer without mutual exclusion.

**PROGRAM 2:** Write a C program to implement Producer Consumer problem using Mutex.

**(Do NOT use POSIX Semaphores in this program)**

Extend Program 1 by using mutex to enclose the critical sections in both producer and consumer so that only one of them can update the buffer at a time and prevent race condition as shown in sample outputs below. Also, consumer should wait if buffer is empty and producer should signal when the buffer has at least one item. You can use unbounded buffer so that producer does not have to wait for buffer availability. Both the producer and consumer threads can be infinite loops and each can randomly sleep to let the other proceed. The output should be the number of items in buffer along with the consumer/producer that is updating the buffer

**Sample Outputs:** Your output can vary depending upon your implementation and the manner in which the threads are scheduled on your system.

****