WEEK 6 – LAB 6 - UE15CS305 – Introduction to Operating System (IOS) Laboratory Problem Statements with Solutions

Write a program to implement classical inter process communication problem (Producer Consumer) using Semaphores.

Background:

Chapter 3 : Read Shared Memory Systems and the producer consumer problem.
Chapter 4 : Read the section on pthreads and how to create threads.
Chapter Read 6.5 and 6.6 to learn about semaphores.
Refer man pages for pthread, semaphores

Implement a main program that creates two threads: producer and consumer threads which executes 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 need to use a semaphore (sem_t) and enclose the critical sections in both producer and consumer so that only one of them can update the buffer at a time. Also, consumer should wait if buffer is empty and producer should signal when the buffer has at least one item. You can assume it is 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 should also 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.

The structure of your program:

```
producer thread function()
        produce an item
        update the buffer

consumer thread function()
        if buffer is empty, wait for producer
        update the buffer
        :
main function()
        initialize the semaphores
        create consumer and producer threads
```

// interprocess communication
// gcc ipc.c -o ipc.exe –lpthread