[Lab-6: Process Synchronization]

Reference:
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/4_Threads.html
https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/5_Synchronization.html

1. **Semaphore** is an integer variable that is shared between threads. This variable is used to solve the critical section problem and to achieve process synchronization in the multiprocessing environment. It is an integer variable which is accessed or modified by using two atomic operations: wait() and signal(). In C program the corresponding operations are sem_wait() and sem_post( ).

   Wait: The wait operation decrements the value of its argument S, if it is positive. If S is negative or zero, then no operation is performed.
   Signal: The signal operation increments the value of its argument S.

   Write a C program using threads that, one to increment the value of a shared variable and second to decrement the value of the shared variable. Both the threads make use of ***semaphore*** *variable so that only one of the threads is executing in its critical section.*

2. To simulate producer-consumer problem using semaphores.

   DESCRIPTION: Producer consumer problem is a synchronization problem. There is a fixed size buffer where the producer produces items and that is consumed by a consumer process. One solution to the producer-consumer problem uses shared memory. To allow producer and consumer processes to run concurrently, there must be available a buffer of items that can be filled by the producer and emptied by the consumer. This buffer will reside in a region of memory that is shared by the producer and consumer processes. The producer and consumer must be synchronized, so that the consumer does not try to consume an item that has not yet been produced.

   Write a C program to ***simulate producer-consumer problem*** *using semaphores. Create buffers for both producer and consumers. The producer should generate tasks and fill up the buffers and the consumer should consume and free up the buffer. Exit the process when the buffer is emptied.*

Submission time: On or before 23:59 hours 27/11/2021.