Report(Assignment 3)

- **How exactly is sychronization achieved using semaphore in our assignnment?**

We achieve this using semaphores. The main process creates two semaphores, consumed and produced, and passes them as arguments to the processes it creates. A semaphore consists of an integer value that is initialized when the semaphore is created and a set of zero or more processes that are waiting on the semaphore. Then, we use System calls wait() and signal() to achieve synchronization.

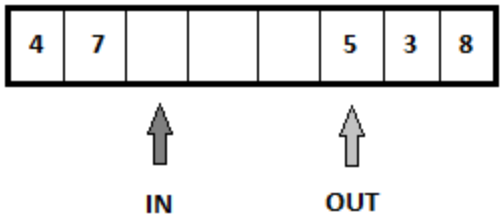
1) **Wait**() decreases the semaphore by 1 and adds the calling process to the process waiting queue if the result is negative
2) **Signal**(): Increases the count of semaphore and resumes execution of a waiting process.

In producer module, when we are producing an item we will invoke wait() call to consumer process to wait till the item is produced and then call signal() to mark the completion of item produced.
Same way in consumer module, we will make producer process wait till the item is consumed and then signal to producer after completion.
This way first an item is produced then only it can be consumed and vice versa.

- **Can the above sychronization be achieved with just one semaphore? Why or why not?**

  No, it cannot be synchronized using only one semaphore because:

  1) The buffer is limited and the two semaphore counts are not independent.



Citing above diagram, there are two pointers required to signify the locations from where the items are produced and consumed. Having two semaphores ensures that at any point of time, the system is aware of the size of the buffer, the size of the empty and full slots in the buffer.

Following is the work done on the assignment:

Producer Module: ShreeHarsha S (sridhash)

Consumer Module: Saurabh Tripathy (saurtrip)