# Assignment – III

1. Synchronization using semaphores: Semaphore is a synchronization tool, that has two operations, wait( ) and signal( ). These two operations are atomic in nature. These two operations are used to maintain the control of the critical section in our program. Wait operation asks the process to wait, until it executes the critical section, in our case would be accessing the buffer, and then it signals the producer, indicating that the wait is now done and they can take access of the critical section, which then accesses (consumes) the values from the common buffer. The consumer calls wait on producer when it is accessing the critical section and then signals the producer. This process switches back and forth, this is how the synchronization is achieved using semaphores.

2. No. We need two semaphores to maintain and indicate the exclusive access of the critical section to one and another, as explained above.

3. Github link with the corresponding branch: https://github.iu.edu/srameshv/OS-FA16/tree/vm_gnumake

4. Function written are:

   a. shellcmd xsh_prodcons(int nargs, char *args[ ])
   b. void producer(int count)
   c. void consumer(int count)

5. Team Contribution:

   a. Raghuveer worked on writing the semaphores, wait and signal sections in xsh_prodcons.c , prodcons.h, producer.c and consumer.c
   b. Sruthi Ramesh Vani contributed in debugging and writing the report.
      Both of us sat together and discussed while working on the assignment.