

Advanced Operating Systems (CSCI-B536) Assignment 3

Hitender Prakash (hprakash@iu.edu)

The assignment 3 is regarding using the semaphore for synchronization in the producer-consumer program (prodcons) that we created in the second assignment. We used semcreate(), wait() and signal() methods to achieve that.

Questions:

1. How exactly synchronization is achieved using semaphore in our assignment?

We created two semaphores consumed and produced. Semaphore consumed and produced are initiated with semcreate(1) and semcreate(0) respectively. When we call producer we obtain a lock through one semaphore (consumed) by calling wait(consumed) and while the producer is done with incrementing the shared variable n, it calls signal (produced). Now consumer call the wait () on produced, prints the value of n and call signal () on consumed so that it becomes ready to be locked by producer again. In this way both producer and consumer works in synchronization.

2. Can the above synchronization be achieved with just one semaphore? Why or why not?

The given setup does not work with one semaphore. If we initiate our semaphore with semcreate(0), then no process producer or consumer is able to obtain the lock as value of the semaphore is already 0. In this case nothing is produced or consumed.

If we initialize semaphore with semcreate(1) then the first process which is called is able to lock the semaphore and do its function. When this process release the lock, its value again becomes 1 and its ready to be taken by any of the two processes, producer or consumer. So it totally depend on the OS process scheduler to call any one of the process. Even if producer is scheduled again, it will run and produce another value without getting the previous value consumed by consumer. In this case we get the result like initially producer runs for a while and produce many values, then after that consumer start running and consume last value many times.

Team and tasks done by team members: We are two team members (Hitender Prakash and Dhaval Niphade). We have done this assignment individually and not collaborated. Both will be submitting separate commit hash and report.

Sources and credits:

1. Operating System Design: The XINU Approach, 2E.
2. Function template are same as given in the assignment instructions at <https://github.iu.edu/SOIC-Operating-Systems/AOS-FA16/wiki/Assignment-2>
3. The code for handling of '-help' in xsh_prodcons.c is inspired from and implemented in the same way as in xsh_date.c file in the xinu source code.