Assignment 3 - Report

How exactly synchronization is achieved using semaphore in our assignment?

The Synchronization is achieved with the help of two semaphore, produced and consumed. Below mentioned points will illustrate the synch process,

- The semaphores, Produced and consumed are initialized to 1, 0 respectively in xsh_prodcons.c file.
- The prodcons command invokes two new processes whose job will be to produce and consume values which is accomplished using a single global variable 'n'.
- In producer program, after the execution reaches inside the main loop, we make a call to wait (produced), availability accessing the critical section (i.e. modifying variable 'n'). This task is accomplished by checking the value of semaphore, if it positive then the process is allowed to enter the critical section else will be put back in ready queue waiting for the semaphore to be available. Wait (Produced) will decrement the value of produced, as we initialized the produced semaphore with a value of 1, the first execution will produce a value. After producing the value, producer process will make a call to signal (consumed) system call, as to increment the value of consumed to 1.
- In consumer process, Now the semaphore consumed's value is 1, The main loop starts with making a call to wait (Consumed) system call check on availability of critical section, Since the value of consumed is 1 at that point, execution will continue consuming 'n' and calling signal (Produced). Hence notifying the producer process about the availability of critical section.
- This process continues, producing value and consuming value alternatively with the help of semaphores, hence achieving synchronization.

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

Synchronization cannot be achieved using single semaphore because both the producer process and consumer process have to wait on a single semaphore, if both processes try to call wait at the same time then the execution can go to any of the two processes resulting in loss of synchronization. The event where two process access the same semaphore can result in deadlock or one of the two can be randomly picked by the scheduler.

Functions,

Prodcons command,

Syntax: Prodcons argument, Argument must be a valid integer value or be left blank. If blank the default value will be 2000

Options: –help – will display the help information about command.

Usage: Invokes two independent processes, Producer and consumer, which will work in synchronous manner to produce and consume values, communicated via a global variable and synchronization handled with the help of two semaphores.

Misc: All the validations and verifications have been handled such as argument validity, number of arguments and correctness of argument.

Producer,

The producer function will run the main loop for specified number of times given by the argument passed to prodcons command else will be run 2000 (default) number of times. The function will work in synch with consumer function by making use of a global variable 'n' and two semaphores, produced and consumed as described at the beginning of the document.

Consumer,

The consumer function will also run in the same fashions as that of producer, only difference will be that, it will consume values (decrement global variable 'n').

Miscellaneous,

All the procedure involved in developing a command are taken care of, such as creation and updation of prodcons.h, shprototype.c etc. Also the make file has been modified to accommodate the apps folder.

Efforts,

The functionality and logic was developed collaboratively.

Anuj,

- Modifying the xsh_prodcons.c
- Updating make file
- report

Chitesh,

- updating consumer and producer function