Q1. How exactly synchronization is achieved using semaphore in our assignment?

Ans. The main process creates two semaphores, consumer and producer, and passes them as arguments to the processes (consumer and producer) it creates. Both the processes producer and consumer are using system calls wait and signal along with semaphores which allows them to maintain synchronization amongst themselves in order to avoid a race like condition to access the common shared variable. The semaphore produced is used to signal the consumer process to wait using through the WAIT keyword while the producer process is producing values, and similarly consumer signals producer to wait while it is consuming values.

Output of the semaphore synchronization is as follows:

xsh $ prodcons gft

The current input is =gft and is an invalid input . Please provide only an integer as input or an integer greater than 0.

xsh $ prodcons fd45

The current input is =fd45 and is an invalid input . Please provide only an integer as input or an integer greater than 0.

xsh $ prodcons 67hgfr

The current input is =67hgfr and is an invalid input . Please provide only an integer as input or an integer greater than 0.

xsh $ prodcons 0

The current input is =0 and is an invalid input . Please provide only an integer as input which is greater than 0.

xsh $ prodcons -34

The current input is =-34 and is an invalid input . Please provide only an integer as input or an integer greater than 0.

xsh $ prodcons 3

The value of n consumed is =3

xsh $ The value of n produced is =1

The value of n consumed is =1

The value of n produced is =2

The value of n consumed is =2

The value of n produced is =3

xsh $ prodcons --help

Usage: prodcons

Description:

n is used as a global variable by both producer and consumer to create & use N values.

Parameter: Two semaphores for producer and consumer, and a single integer for the value of n are passed as parameters.

Default: A default value of 2000 is used when no parameter is passed.

Q2. Can the above synchronization be achieved with just one semaphore? Why or why not?

Ans. No, Ideally the above synchronization cannot be achieved using just one semaphore as both producer and consumer require individual semaphores to invoke wait and signal system calls. In Xinu, all processes execute concurrently. That is, execution of a given process continues independent of other processes unless a programmer explicitly controls interactions among processes . But upon executing the code with a single semaphore the output was found to be in sync .

Name of all functions used in the producer-consumer problem are as follows:

* shellcmd xsh\_prodcons(int nargs, char \*args[])
* void producer(int count)
* void consumer(int count)

Note the tasks of the group members who developed the various components.

**Task Member User ID**

Created prodcons.h pandeyh

Created xsh\_prodcons.c sshalabh

Created produce.c pandeyh

Created consume.c sshalabh

Made entries in shprototpes.h in include folder sshalabh

Made entries in shell.c in include folder pandeyh

Creation of Test Cases pandeyh

Testing and Execution of code sshalabh

Bug Fixing sshalabh

Report Creation pandeyh