Advanced Operating Systems (CSCI-B536) Assignment 2

Hitender Prakash (hprakash@iu.edu)

The assignment 2 is focused on knowing about Xinu processes. We used two important Xinu systems calls create() and resume().

The two processes are created which shares a common variable, one process produce (write) some values and the other process tries to consume (read) these values with the help of shared variable.

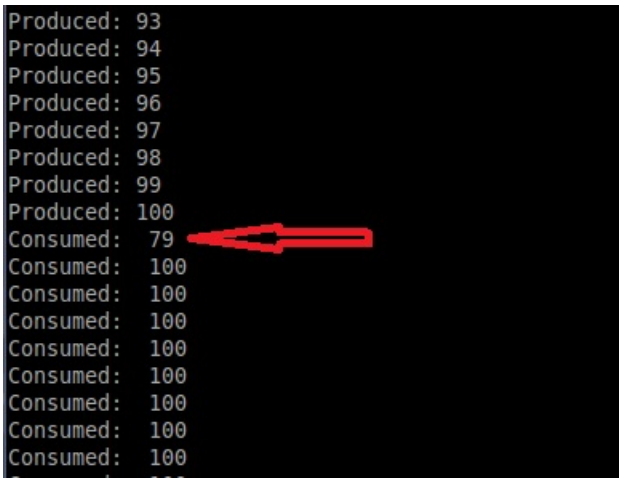
Questions:

1. Does your program output any garbage? If yes, why?

   The shared variable properly n was properly initialized (n=0) in the start of the program, hence did not get any garbage value. Also in the main shell code we have created and placed the producer thread earlier than consume. However if we do not initialize the n and call the consumer thread before produce, then we get garbage value printed. This happens simply because the n has neither been explicitly initialized nor it is being set by producer before being read by consumer.

2. Are all the produced values getting consumed? Check your program for a small count like 20

   No. Normally when the program is run for smaller count like 20 the producer finishes completely before consumer can start executing hence the consumer prints only the last value for every iteration in the loop.
   However if we run the program with larger count value (say 100) then producer prints the value from 1 to 100 and then consumer prints any intermediate value followed by last count values in the entire loop after that. Below is the screenshot for program run for count=100.



Functions used in the programs:

1. shellcmd xsh_prodcons(int , char *args[])
2. void consumer(int count)
3. void producer(int count)

Contribution of the team members:

For this assignment we have not collaborated in team. We will be submitting this assignment individually with different hash.

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.