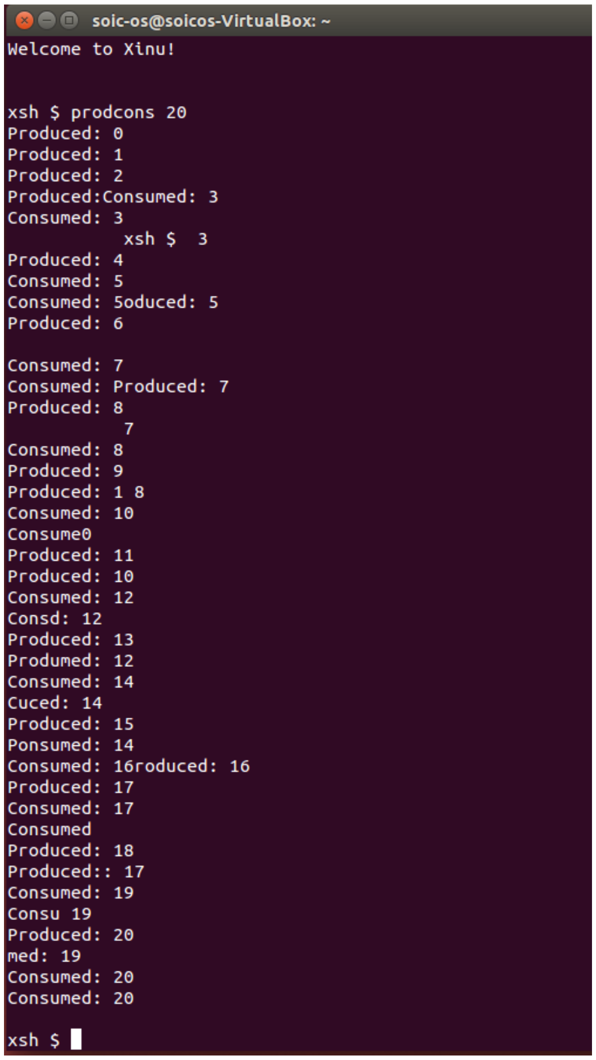
Advanced Operating Systems - Assignment 2 (CSCI P536 Fall 2015)
Team members

Srivatsan Iyer (srriyer)
Pralhad Sapre (pssapre)

Does the program output any garbage? If yes why?

The program doesn't output garbage values of variables (since there is only one variable involved). However it outputs garbage printing and interleaved values. It often happens that the producer's output via *kprintf()* gets interrupted and the consumer starts its printing, resulting in a garbage message output.

Here is a sample



```
Welcome to Xinu!


xsh $ prodcons 20
Produced: 0
Produced: 1
Produced: 2
Produced:Consumed: 3
Consumed: 3
           xsh $  3
Produced: 4
Consumed: 5
Consumed: 5oduced: 5
Produced: 6

Consumed: 7
Consumed: Produced: 7
Produced: 8
           7
Consumed: 8
Produced: 9
Produced: 1 8
Consumed: 10
Consume0
Produced: 11
Produced: 10
Consumed: 12
Consd: 12
Produced: 13
Produmed: 12
Consumed: 14
Cuced: 14
Produced: 15
Ponsumed: 14
Consumed: 16roduced: 16
Produced: 17
Consumed: 17
Consumed
Produced: 18
Produced:: 17
Consumed: 19
Consu 19
Produced: 20
med: 19
Consumed: 20
Consumed: 20

xsh $ 
```

The reason for this is that the producer and consumer are not synchronized. There is no mechanism in place which provides exclusive access to either of these processes. That's why the output generated depends on how each of them are scheduled at runtime.

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

The consumption of the produced values totally depends on the runtime scheduling of each run of prodcons. We can't say with certainty that all produced values will be consumed because there is no mechanism like semaphore in place to synchronize the producer and consumer processes. It might happen with a stroke of luck that the consumer does consume all the produced values, but again it all runtime specific.

Please refer to the output in the previous section. Note that not all produced values are consumed.

Functions in the project and task distribution

The following important components were developed in this assignment.

prodcons.h file - contains the declarations of the global variable 'n' and producer() and consumer() functions. The variable 'n' is declared as an extern variable in this file. The definition appears in xsh_prodcons.c file. It is through this header file that this variable is available for use by files like produce.c and consume.c.

xsh_prodcons.c file - contains the code for the shell command prodcons. In addition to sanity check for input variables, this compilation unit is responsible for spawning off two threads – producer process and consumer process.

produce.c file - contains the producer() function which increments the value of global variable 'n' and prints the information. This is value which the consumer() function in consume.c file will access.

consume.c file - contains the consumer() function which prints the current value of 'n' and goes on in a loop till count.

Task Distribution

Srivatsan Iyer – Adding the shell command by adding xsh_prodcons.c, editing shell.c and shprototypes.h.

Pralhad Sapre – Added producer and consumer core logic in produce.c and consume.c respectively. Added sanity checking to xsh_prodcons.c.

Note: Since we pair-programmed, most commits were done from a single machine.