Maharshi Shah

Dr. Valentina Korzhova

Operating Systems

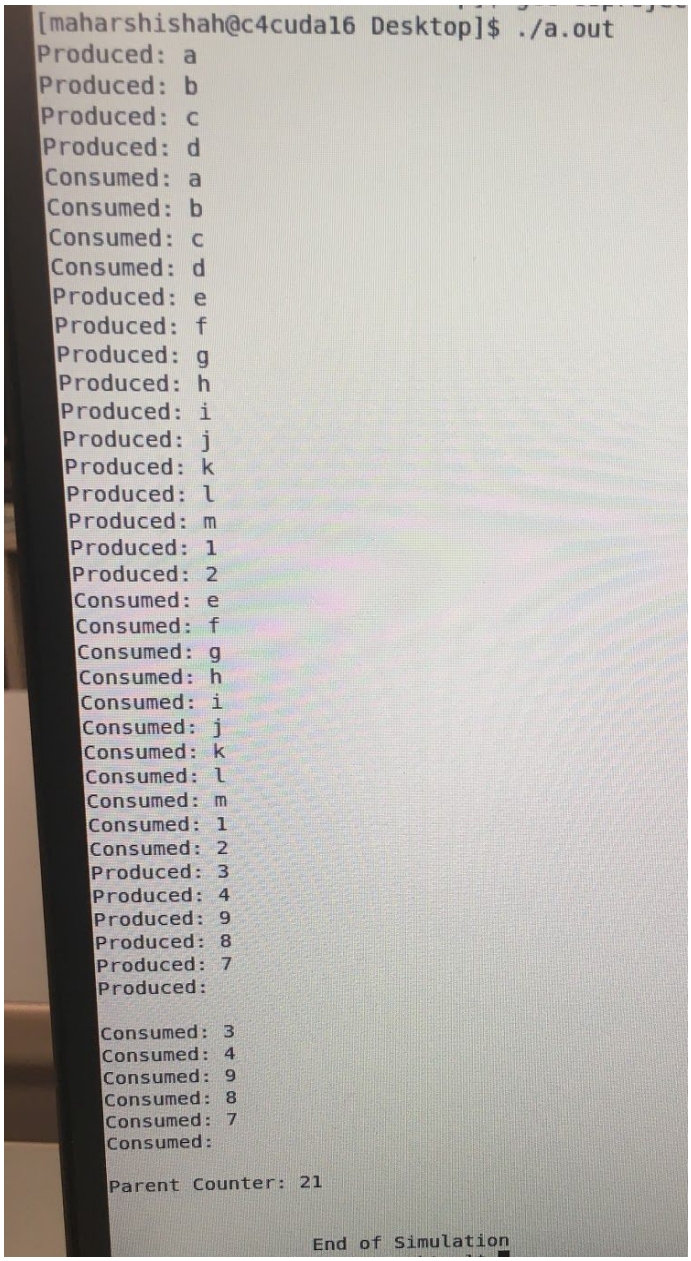
October 14, 2018

Bounded Buffer - Project 3

The purpose of this project was to understand the use of sempahores to protect a bounded buffer with circular implementation. The bounded-buffer problem, more commonly known as the producer-consumer problem deals with the synchronization of processes. Two threads were created namely, producer and consumer. The producer thread "produces" data and tries to insert that data into a slot with 'n' number of total slots. In this case, n is equal to 15. The consumer thread "consumes" the data inserted and makes one slot empty in the buffer. Semaphores help synchronize these two threads by releasing and locking and not allowing more than one process or thread to run in its critical section. A text file containing characters will be read by the c code and placed onto the buffer with 15 positions. The code sets the index back to 0 when 15 characters have been consumed.

The following page shows the images after running the code developed. In order to run the code on the C4 machines, following are the steps:

- gcc osp3.c -lpthread -lrt.c **To compile**
- ./a.out **To run the code (More information in the readme file)**



CONCLUSION

A total of 21 characters are read from the mytest.dat file that the c code operates on. The consumer has been designed in such a way that it waits until one character has been filled in the buffer. The empty semaphore is incremented by 1 as the consumer function has just removed

data. The code does take considerable time to run when the text file contains 150 characters. However, this project was important to understand the producer consumer problem better as it protected data and synchronized threads when accessing the same abstract data structure like a circular bounded buffer.