A shared Protected Circular Queue and Communication between threads

Karshan Arjun

COP4600: Operating Systems University of South Florida, USA

Abstract

The purpose of this project is to understand the producer and consumer problem. The use of a circular queue and the combinations between thread will help create an efficient system.

Design

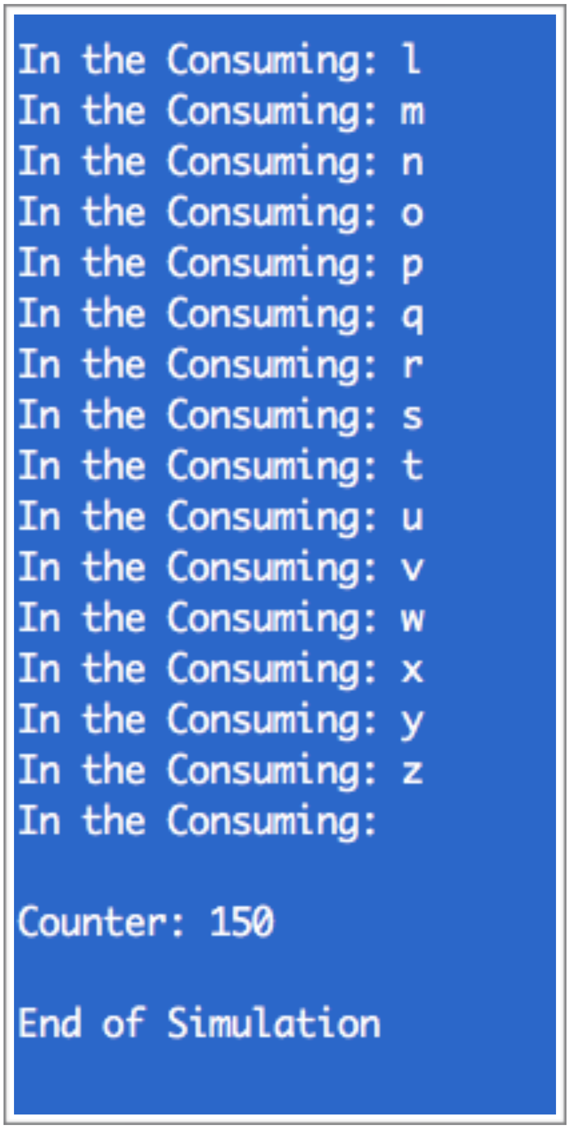
The start of the program was to use a buffer size of 150 to hold the amount of characters needed for this project. Buffer structure were used for data handling on the buffer and also for the counter. Two functions were created that would implement a circular queue.

The consumer and producer would be the main functions on this program. In the producer function the a loop with take in the empty and critical item as they produce 15 item at a time. Then post the characters with sem_post() function. The consumer will then take the 15 charters and then output them and also sem_post() the characters. This will keep on going until all the characters in mytest.dat file is produced and consumed.

In the main function share memory is implemented and the sem_init() are used to created the handling of the objects. Pthread is used as well to create threads.

Results

With a one second wait the consumer will out each item in the sem_post(). The counter is used to determine how many items were counted.



```
In the Consuming: l
In the Consuming: m
In the Consuming: n
In the Consuming: o
In the Consuming: p
In the Consuming: q
In the Consuming: r
In the Consuming: s
In the Consuming: t
In the Consuming: u
In the Consuming: v
In the Consuming: w
In the Consuming: x
In the Consuming: y
In the Consuming: z
In the Consuming:

Counter: 150

End of Simulation
```

Performance:

Because of the one second wait the time for the all the characters to print out will be 150 seconds. Using the time linux commands, real time took 0m28.01s, sys time took 0m0.002s and user took 0m001s. Overall great timing.