CS 474 – Project

Bounded Buffer/Producer-Consumer Problem

Antonio Maldonado
Computer Science
New Mexico State University
Las Cruces, New Mexico
m21tony@nmsu.edu

Description

This program will teach us about how to use semaphores to protect a limited size resource. A circular buffer with 15 positions for characters will be used to communicate information between threads, a producer and a consumer.

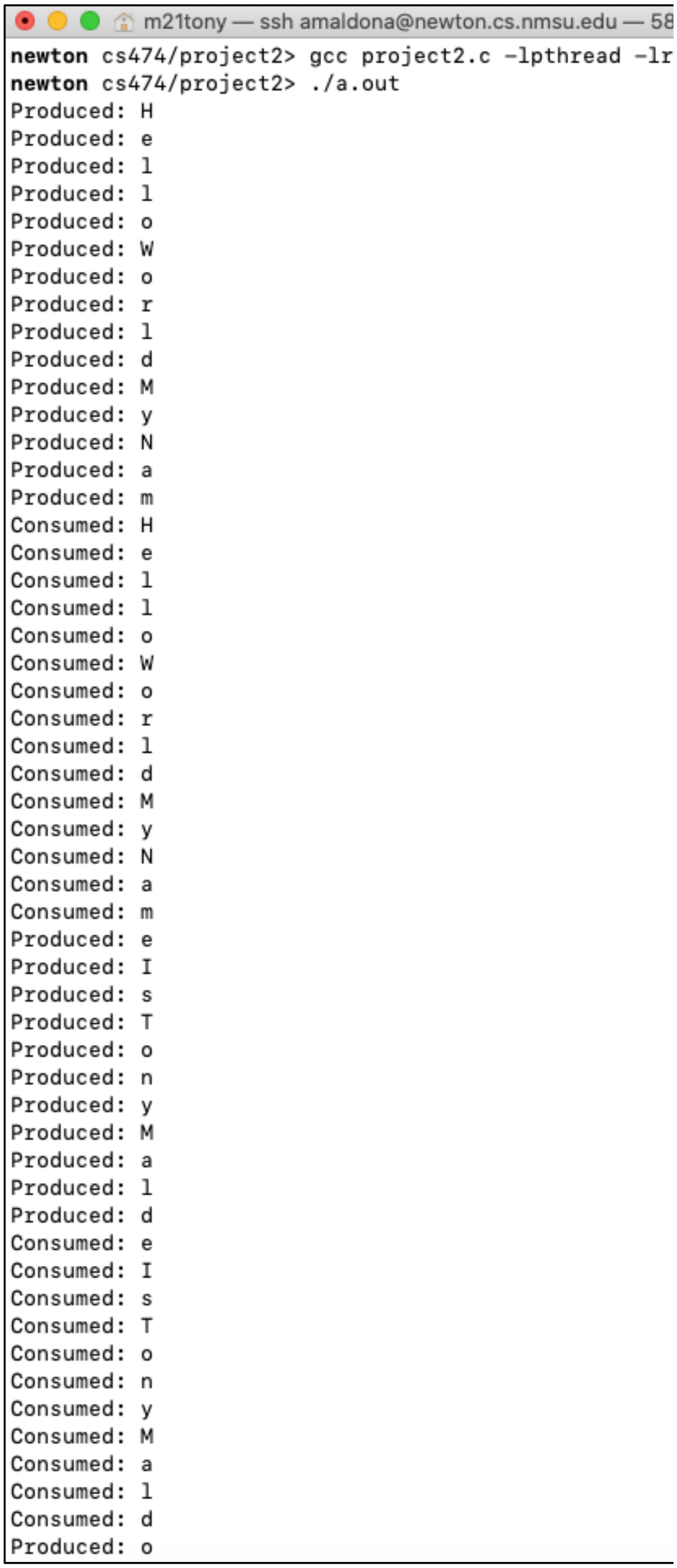
The producer will read characters until the end of file and print to screen and inform the consumer when to consume the buffer. The producer does this by placing a special character into the shared buffer - * - then the consumer knows to start consuming. As the consumer reads the characters, it will print to the screen.

Conclusion

A total of 35 characters were successfully read from ‘mytest.dat’ file. The producer read the characters one by one and placed them into the buffer until 15 chars were reached and then the consumer thread consumed the chars and outputted to the screen. The consumer was implemented in a manner that it’s supposed to be slower than the producer, hence there is a one second sleep in the thread between “reads” from the shared memory. So, the program can take a significant amount of time to run fully for larger test sets, up to 150 characters. This project was certainly important to understand producer consumer problem and to help visualize thread processes.
See the results below.

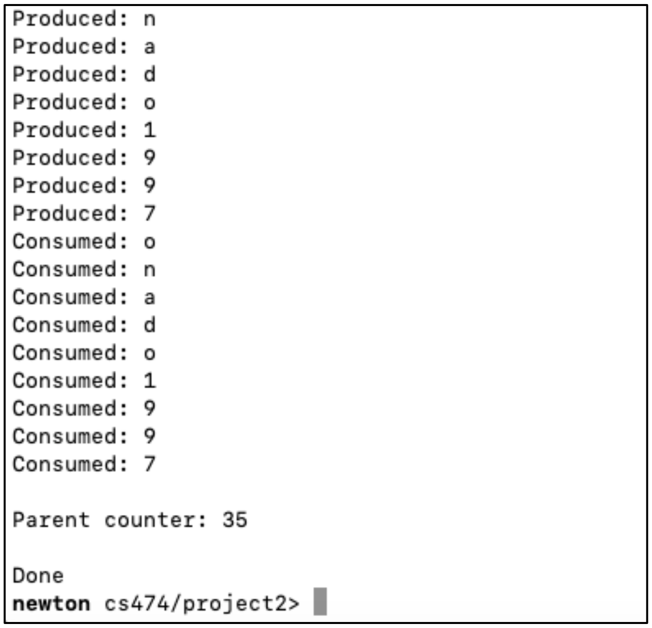
Output: Screenshots (below)

Output: Screenshots



```
newton cs474/project2> gcc project2.c -lpthread -lr
newton cs474/project2> ./a.out
Produced: H
Produced: e
Produced: l
Produced: l
Produced: o
Produced: W
Produced: o
Produced: r
Produced: l
Produced: d
Produced: M
Produced: y
Produced: N
Produced: a
Produced: m
Consumed: H
Consumed: e
Consumed: l
Consumed: l
Consumed: o
Consumed: W
Consumed: o
Consumed: r
Consumed: l
Consumed: d
Consumed: M
Consumed: y
Consumed: N
Consumed: a
Consumed: m
Produced: e
Produced: I
Produced: s
Produced: T
Produced: o
Produced: n
Produced: y
Produced: M
Produced: a
Produced: l
Produced: d
Consumed: e
Consumed: I
Consumed: s
Consumed: T
Consumed: o
Consumed: n
Consumed: y
Consumed: M
Consumed: a
Consumed: l
Consumed: d
Produced: o
```

(continued screenshot)



```
Produced: n
Produced: a
Produced: d
Produced: o
Produced: 1
Produced: 9
Produced: 9
Produced: 7
Consumed: o
Consumed: n
Consumed: a
Consumed: d
Consumed: o
Consumed: 1
Consumed: 9
Consumed: 9
Consumed: 7

Parent counter: 35

Done
newton cs474/project2>
```