CSE Project 4: Producer-Consumer Problem

Wang Ziyi
515030910578
wzy_sjtu@foxmail.com

December 14, 2017

Abstract

This is the report for my CSE course project 4: Producer-Consumer Problem in SJTU. In this project, I completed the code for solving the producer-consumer problem(also known as bounded-buffer problem) using Pthread with applications of semaphores and mutex and so on.

1 Project Description

In this project, the consumer requires data from the buffer, and the producer produces data and put it into the buffer. We have to be very careful in our case that we dont over fill the buffer or remove something that isnt there; in c this will produce a segmentation fault. Here we use semaphores to synchronize among the threads in the concurrent programming.

2 Implementation

The textbook provided some part of the codes needed. Using the APIs in Pthread like pthread_createc and the semaphores full and empty, we are able to create a synchronization rule among the producers and consumers. The mutex lock should be locked up when the data in the buffer is being manipulated. The detailed implementation process can be found in the source code.

3 Problems and Solutions

This project is based on the Pthread API and the use of semaphore so before implementing it, I looked up some documents on it and learned about some

of its important functions and their usage. After understanding them, the programming process is quite smooth.
This project is a advanced problem based on the previous project 3. So this time I am much more proficient in coding.

4 Source Code

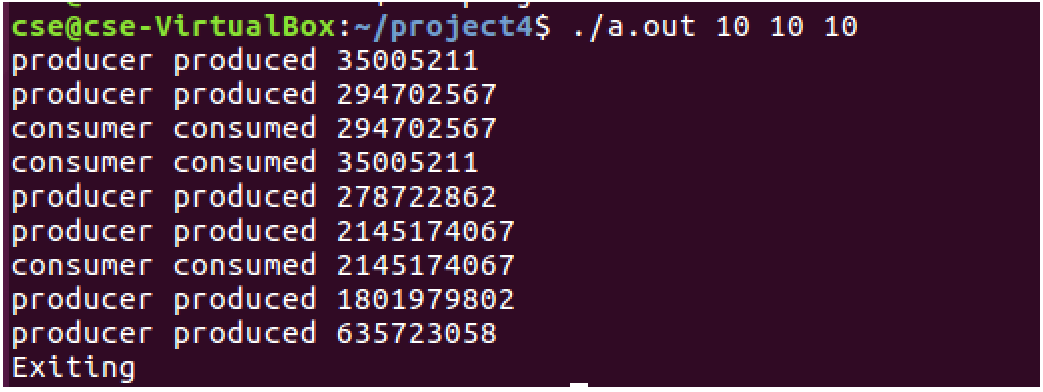
See the file **producer-consumer.c**.

5 Conclusion

In conclusion, this project checked our ability about coding and using threads and semaphores. Enough understanding on the bounded-buffer solution to the producer-consumer problem is also very important. I really like solving a real problem like this, which makes me feel a sense of achievement.

6 Test

This is a snapshot for running the program on Linux:



```
cse@cse-VirtualBox:~/project4$ ./a.out 10 10 10
producer produced 35005211
producer produced 294702567
consumer consumed 294702567
consumer consumed 35005211
producer produced 278722862
producer produced 2145174067
consumer consumed 2145174067
producer produced 1801979802
producer produced 635723058
Exiting
```