

OS Assignment 9

Written by Akilesh B, CS13B1042

October 29, 2015

Objective:

To implement the bounded buffer producer-consumer problem using semaphores.

Input:

Input is a text file which consists of capacity, producer threads, consumer threads, producer loops, consumer loops, average sleep values.

How I implemented?

The first step is to create the consumer and producer threads. The consumer threads keeps consuming the produced items in the buffer and the producers keep producing in the buffer. In the producer action that is performed by the producer threads, we first check if the buffer is full and wait until its not full. There is also be a possibility of race condition between the producer which is solved using the pthread mutex. When a producer reaches the critical section we can be sure that no other thread is accessing the buffer. That is done through semaphores in the following way:

```
sem_wait(&empty);
pthread_mutex_lock(&buffer_mutex);
/* Produce and insert in buffer */
pthread_mutex_unlock(&buffer_mutex);
```

Now the semaphore and the mutex locks are modified. Similarly, in consumer threads, the semaphore is waited if the buffer is empty and once it has a value , there will be a race condition and the mutex is used to solve it.

Explanation:

For very low values of ratio (producer delay : consumer delay), consumer takes more time than producers. This is because consumer consumes only after producer produces. This adds to the delays for consumer. The same doesn't hold true for high ratio, it is because they almost converge.

Graph:

