
Documentation

The producer and consumer processes share a buffer of fixed size to facilitate an Inter Process Communication environment in an operating system. The producer writes data into the buffer while the consumer reads data from the buffer, one at a time. The process is controlled to ensure the producer does not write data into the buffer when it is full. Similarly, the consumer cannot read from an empty buffer. At an instance when the producer attempts to place information into the buffer when it is full, it squanders CPU cycles. The equivalent is valid for the consumer when it attempts to take from an empty buffer. It's better that they go on rest in these cases so the scheduler can plan another cycle.

The above process can be simulated by a program that utilizes semaphores in threads and mutexes. The mutex is for providing mutual exclusion when both programs are at critical sections. The producer produces an item then waits or sleeps when there are no empty slots in the buffer. It will then signal the consumer to wake it that there are items in the buffer to be read. Similarly, the consumer will wait or sleep until there are no full slots in the buffer. Then it signals or wakes the producer to inform it of empty slots in the buffer to be filled. The process continues for a defined number of items in the buffer and CPU cycles.

The following commands are used to run the programs.

```
$ gcc producer.c -pthread -lrt -o producer
$ gcc consumer.c -pthread -lrt -o consumer
$ ./producer & ./consumer &
```