COS3043
System Fundamentals

Lab 6

Task

Design, develop and implement a process with a producer thread and a consumer thread which make use of bounded buffer (size can be prefixed at a suitable value) for communication. Use any suitable synchronization construct.

Basic Concepts

Producer-Consumer Problem

Producer-consumer problem (also known as the bounded-buffer problem) is a classical example of a multiprocessor synchronization problem.

The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer. The producer's job is to generate a piece of data, put it into the buffer and start again. At the same time the consumer is consuming the data (i.e., removing it from the buffer) one piece at a time.

The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

Producer

Producer corresponds to produce the item and stores in the buffer. After storing the item in the buffer, it has to increment the counter also it has to notice the consumer to consume the item.

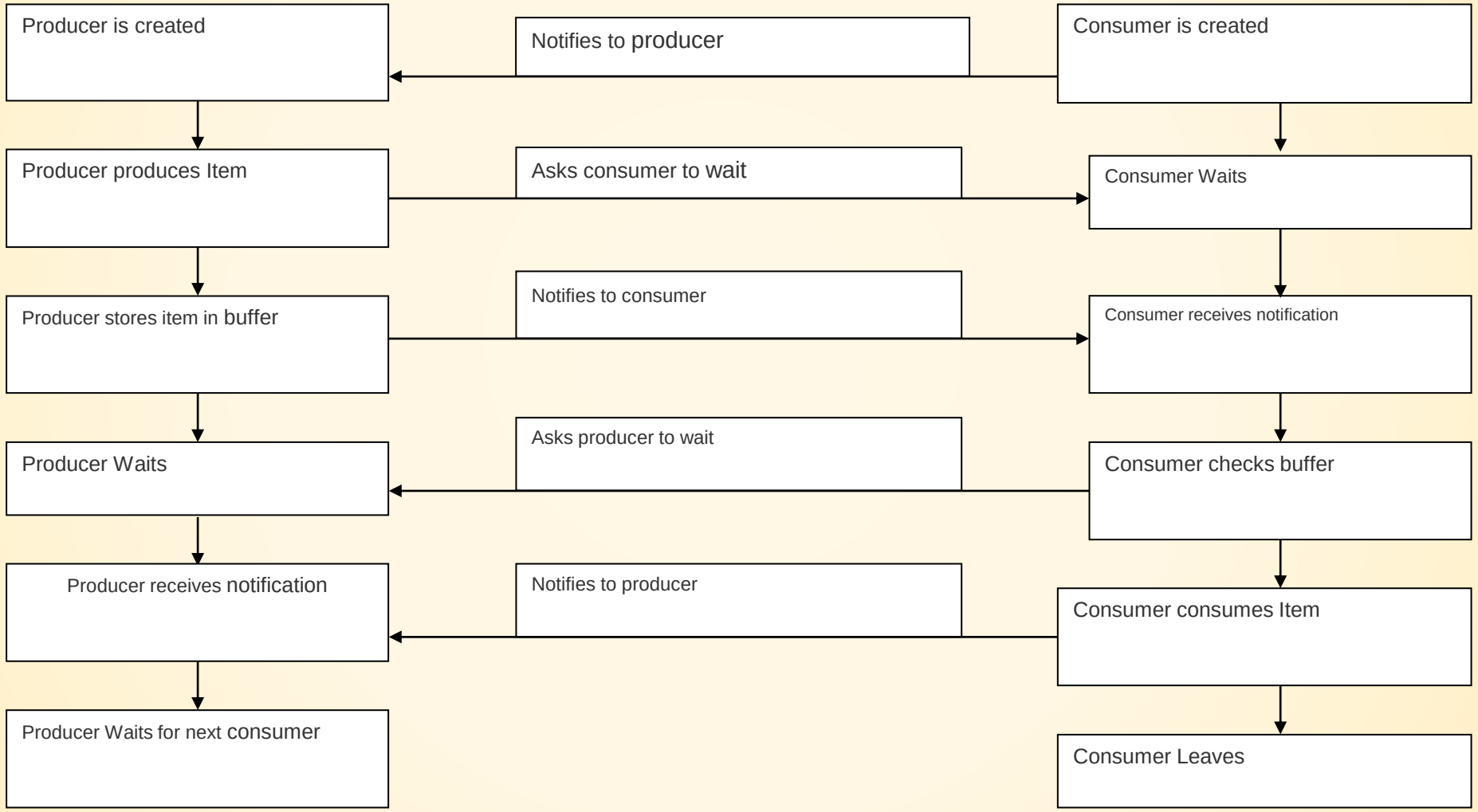
Consumer

Consumer corresponds to consume the item available in the buffer. After consuming the item in the buffer, it has to decrement the counter also it has to notice the consumer to consume the item.

Bounded-Buffer

It works as the interface between the Producer and the Consumer. It allows the producer to store the item in the buffer (shared space), also allows the consumer to consume the item from the buffer (shared space).

Flowchart


Producer is created
Notifies to producer
Consumer is created
Producer produces Item
Asks consumer to wait
Consumer Waits
Producer stores item in buffer
Notifies to consumer
Consumer receives notification
Producer Waits
Asks producer to wait
Consumer checks buffer
Producer receives notification
Notifies to producer
Consumer consumes Item
Producer Waits for next consumer
Consumer Leaves

Algorithm

Producer process:

```
//item nextProduced;
while (buffer_size = =max_size) {
        /* do nothing */
        wait( );
 }
Add item into buffer;
notify( );
```

Consumer process:

```
//item to be Consumed;
while (buffer_size = = 0) {
        /* do nothing */
        wait( );
}
Remove from item buffer;
notify( );
```