Producer-consumer problem

Producer-Consumer Problem :- The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer.

Producer-Consumer problem also known as the **bounded-buffer Problem** is a multiprocess Synchronization problem.

Producer : The producer's job is to generate a piece of data, Put it into the buffer and start again.

Consumer : The consumer is consuming the data (i.e removing it from the buffer) one piece at a time.

If the buffer is empty, then a consumer should not try to access the data item from it.

Similarly, a producer should not produce any data item if the buffer is full.

Counter Counts the data items in the buffer. or to track whether the buffer is empty or full. Counter is shared b/w 2 processes and updated by both.

How it works ?

→ Counter value is checked by Consumer before Consuming it.

→ If Counter is 1 or greater than 1 then it starts executing the process and updates the Counter.

→ -lly Produces checks the buffer for the value of Counter ~~fo~~ for adding data.

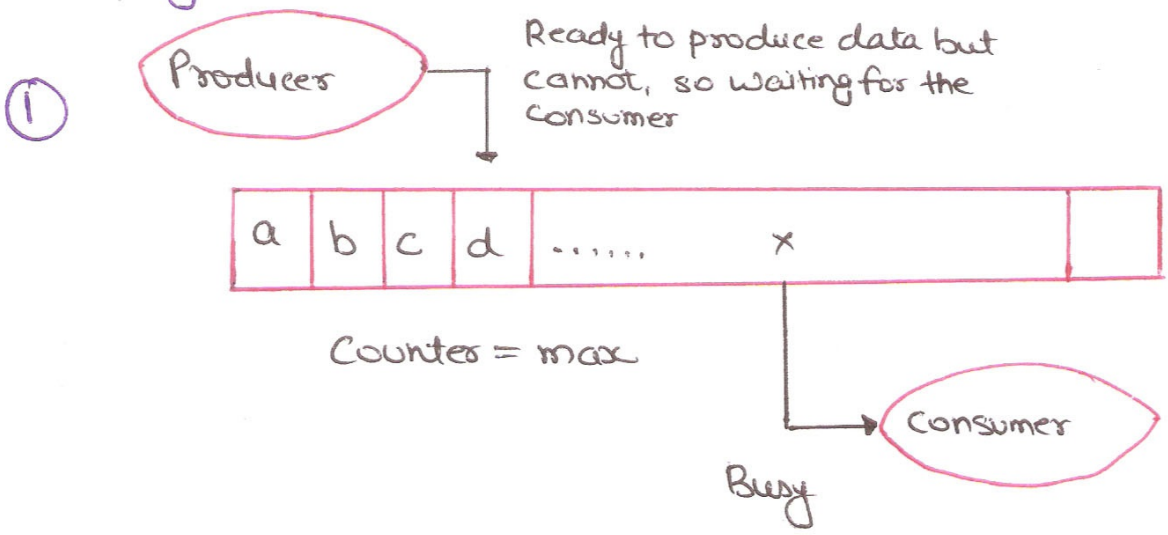
→ If the Counter is less than its maximum value, it means that there is some space in the buffer.

It starts executing for producing the data item and updates the

Counter by incrementing it by one.

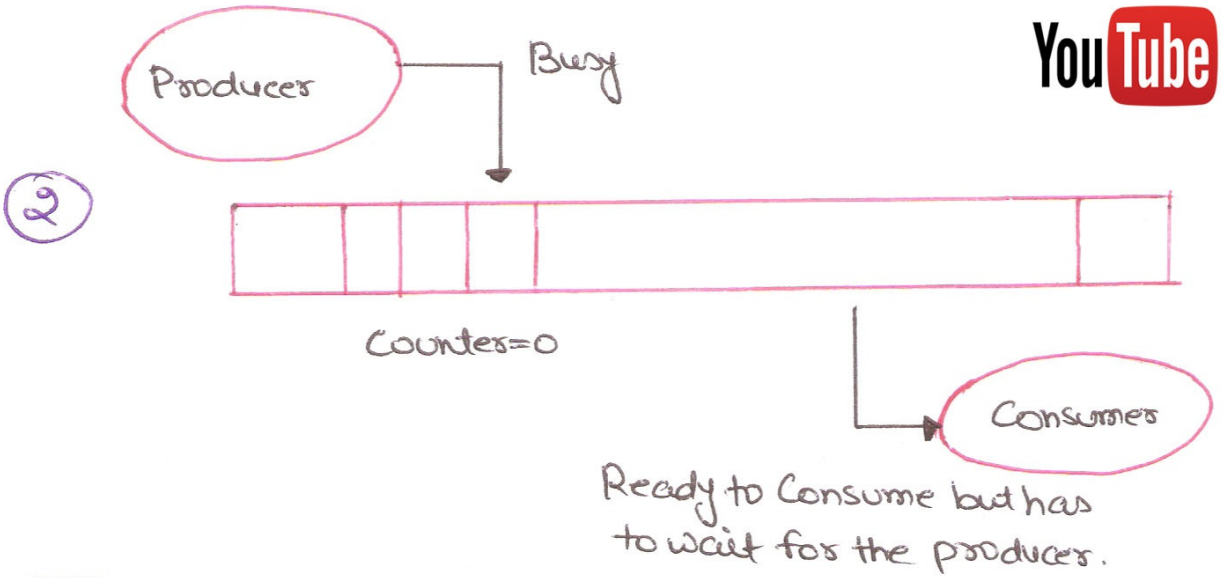
Let max = maximum size of the buffer.

If buffer is full then Counter = max and Consumer is busy executing other instructions or has not been allotted its time Slice yet.



Producer-Consumer problem : Buffer is full.

In this buffer is full so producer has to wait until consumer set counter by decrementing its value by 1.



In this Situation, buffer is empty, that is Counter = 0, and the Producer is busy executing Other instructions or has not been

allotted its time slice yet. At this consumer is ready to consume an item from the buffer.

Consumer waits until Counter = 1

When the buffer is empty and producer busy in filling data items in buffer in while Consumer goes to SLEEP.

When the Counter goes to 1 then System generates WAKEUP Calls to make Consumer to wakeup & start executing it.