**PRODUCER CONSUMER**

* + Producer puts items in buffer
  + Consumer takes items out of buffer
  + NB:

-if the buffer is full the producer waits

-if the buffer is empty not allowed call the consumer

Because you will get deadlock .it will lock the mutex denning access to cpu for all threads .the consumer will set the mutex to 0 causing it to wait & because the mutex is 0 the producer cant write to the buffer so you get deadlock .

|

>> that’s why we check if buffer is empty before we run consumer

* Producer consumer problem is also known as bounded buffer problem. In this problem we have two processes, producer and consumer, who share a fixed size buffer. Producer work is to produce data or items and put in buffer. Consumer work is to remove data from buffer and consume it. We have to make sure that producer do not produce data when buffer is full and consumer do not remove data when buffer is empty
* The producer should go to sleep when buffer is full. Next time when consumer removes data it notifies the producer and producer starts producing data again. The consumer should go to sleep when buffer is empty. Next time when producer add data it notifies the consumer and consumer starts consuming data. This solution can be achieved using semaphores.