

Final Syllabus Start

Chapter - 5

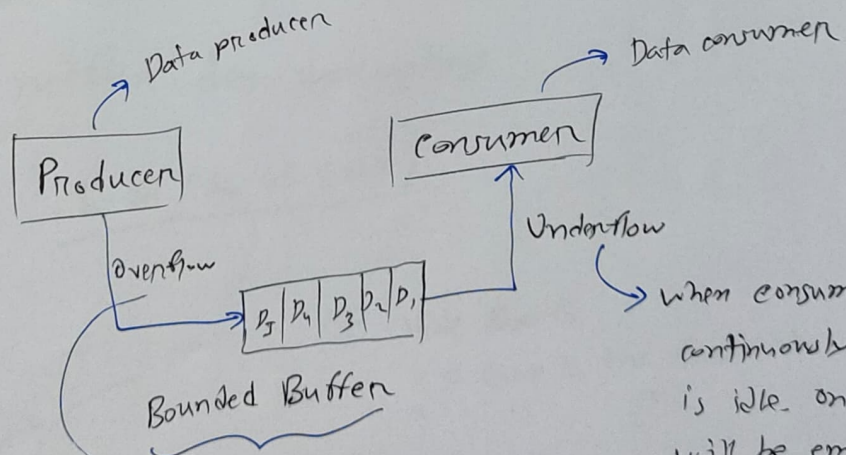
Process Synchronization

⊗ Cooperating Process:

- that can affect or be affected by other processes executing in the system.
- can share resource by any available way.

⊗ Bounded Buffer or Producer Consumer Problem

- multi-process synchronization problem.



✗ if data overflow then, bounded buffer problem, happens when producer run continuously but consumer is idle.

When consumer run continuously and producer is idle, once buffer will be empty and consumer will need garbage.

⇒ When,
buffer full ⇒ Producer should not run
buffer empty ⇒ Consumer should not run } We need to make ~~st~~ sure of that

⊛ Solution:

- need inter-process communication
 - by using semaphores
- deadlock can appear if both processes are waiting to be awaked.

⇒ When,

buffer full ⇒ Producer need to sleep or discard data

~~and~~ consumer will notify to fill again

buffer empty ⇒ Consumer need to sleep

- producer will ~~not~~ notify to remove data.

- Use system call named, sleep & wakeup

5-6 ***

→ Problem!

Let's say, Buffer size is full.

So, when producer check the condition $(itemCount == BUFFER_SIZE)$

It will find, its true. And the ~~at~~ same time producer

get interrupted by another process.

Before resume of producer if consumer run, then

the condition $(itemCount == BUFFER_SIZE - 1)$ will be true and consumer will try to wakeup the producer but, producer is not on sleep, so doesn't work.

Now, if producer resume and executed `sleep()`

it will be on sleep forever.

Because, ~~can~~ condition on consumer will not be true

after that. and after removing all item, consumer will also go to sleep.

⇒ Deadlock problem
- race condition

Page - 7 - Done