

## Producer Consumer Problem

**Case II:**

When The processes are not synchronized and preemption happens among processes.

**Case:** Producer · I<sub>1</sub>, I<sub>2</sub> consumer · I<sub>1</sub>, I<sub>2</sub> Producer · I<sub>3</sub> (Terminated) consumer · I<sub>3</sub> (Terminated)

**Consumer**

Void Consumer (void)

```
{
  int itemC;
  while (true)
```

**BUFFER EMPTY**

```
  while (count == 0);
  itemC = Buffer[out];
  out = (out + 1) mod n;
```

```
  count = count - 1;
  process - item (itemC);
```

```
1. load Rc, m[count];
2. DECR Rc;
3. store m[count], Rc;
```

out = 0

0	x <sub>1</sub>
1	x <sub>2</sub>
2	x <sub>3</sub>
3	x <sub>4</sub>
4	
5	
6	
7	

itemP = x<sub>4</sub>

R<sub>p</sub> = 3

After executing I<sub>2</sub> in Producer, It got Preempted. That means, this process got interrupted and the increment of count did not fully happen!

int count = 0;

**Producer**

Void Producer (void)

```
{
  int itemP;
  while (true)
  {
    Produce - item (itemP);
    while (count == n) → Buffer full
    Buffer[in] = itemP;
    in = (in + 1) mod n;
    count = count + 1;
  }
```

```
1. load Rp, m[count];
2. INCR Rp;
3. STORE m[count], Rp;
```

R<sub>c</sub> = 3 2 | And again after executing the I<sub>2</sub> in consumer, It got Preempted. That means the process got interrupted.