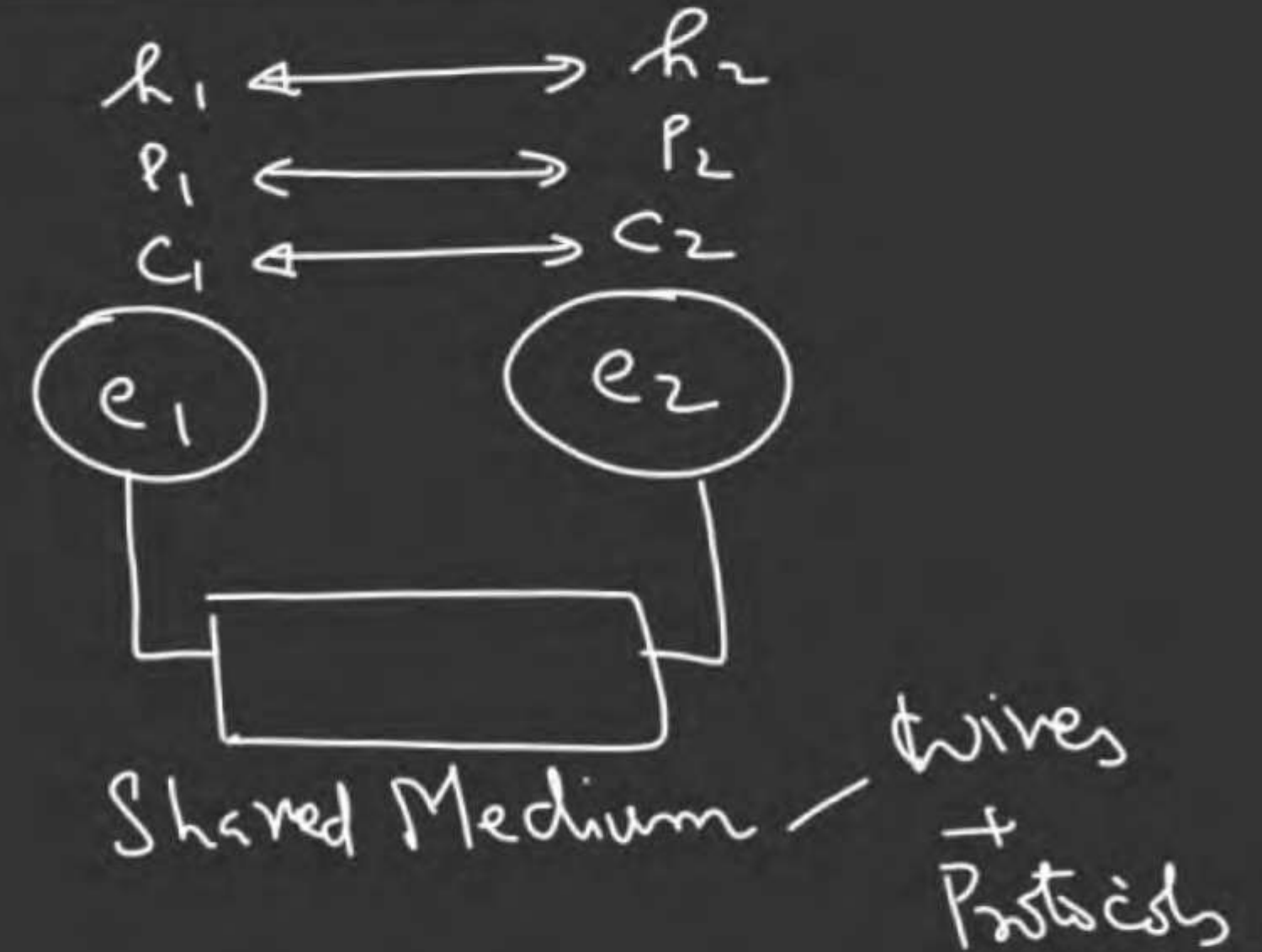
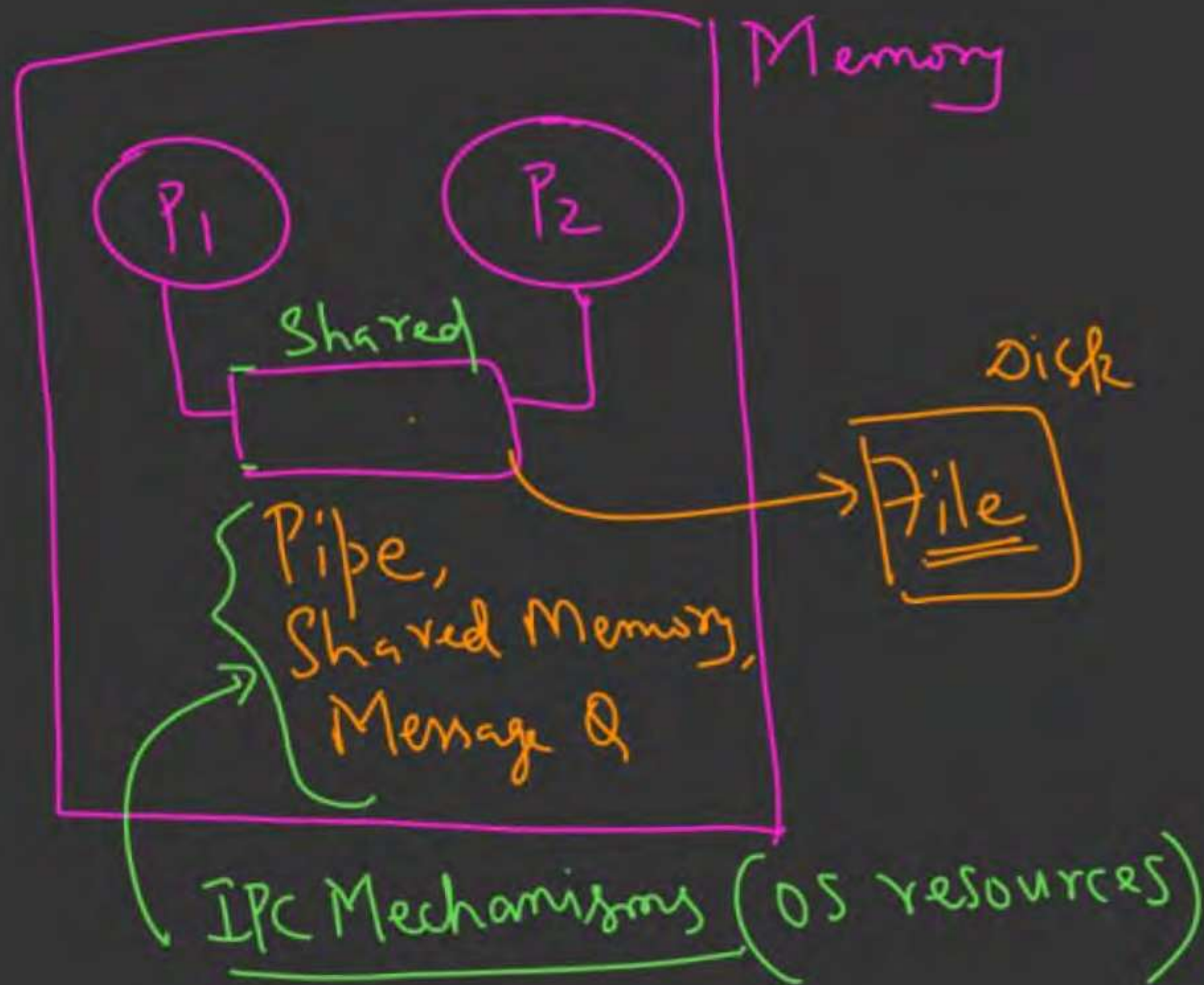


What is IPC :

↳ Inter Process Communication;



The Communication must be Synchronized

Otherwise it will Lead to Problems;

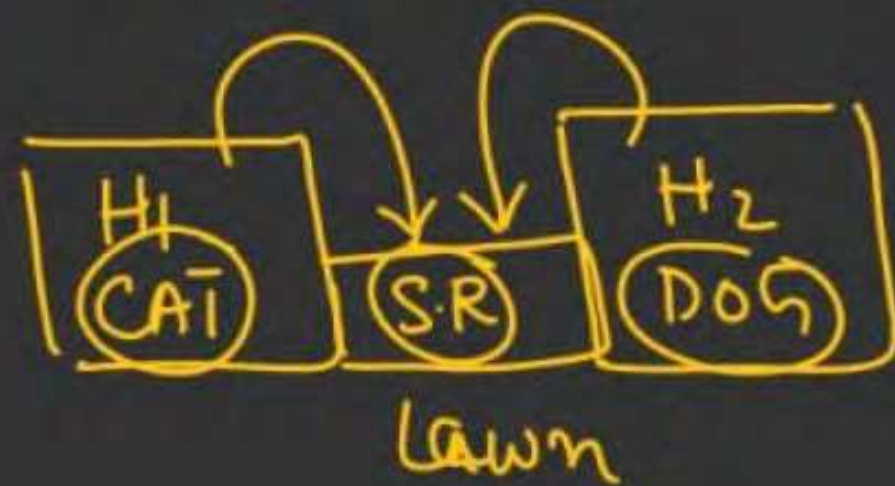
{ Need for Synchronization

(i) Inconsistency (Incorrectness)

(ii) Data Loss

(iii) Deadlock (Processes gets Blocked for ever)

(i)



(ii) Paying Guest (Milk)

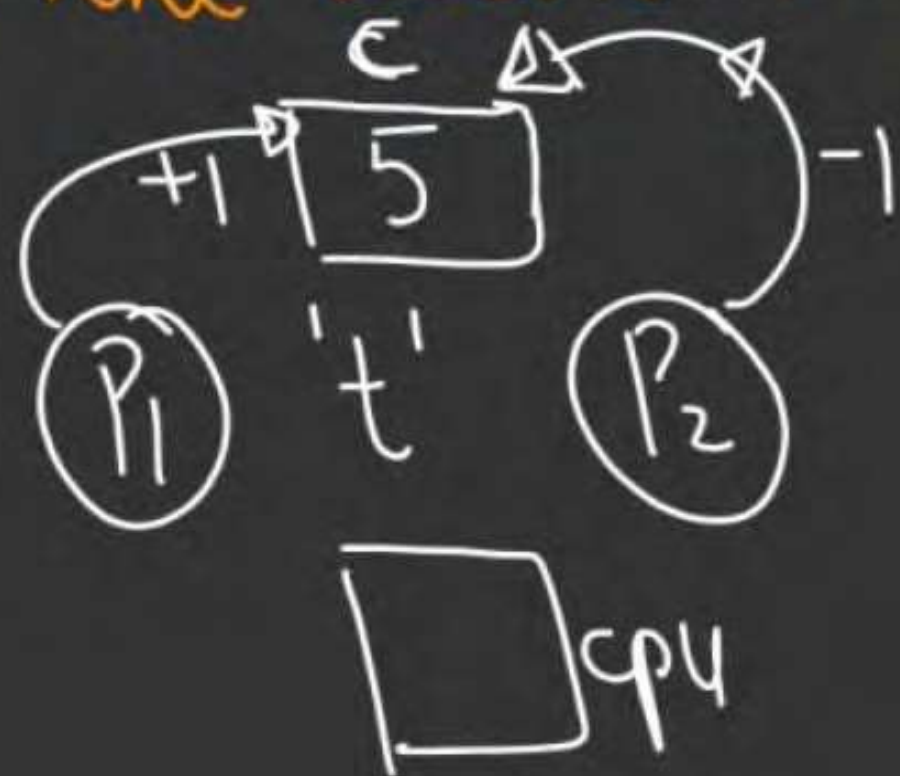


Types of Synchronization (in IPC Environment)

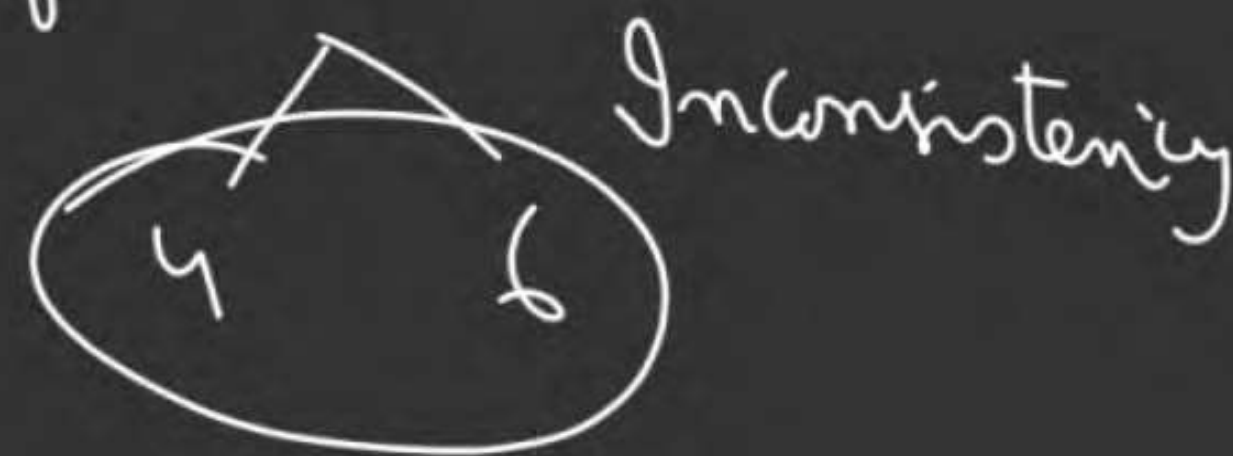
Competitive

Cooperative

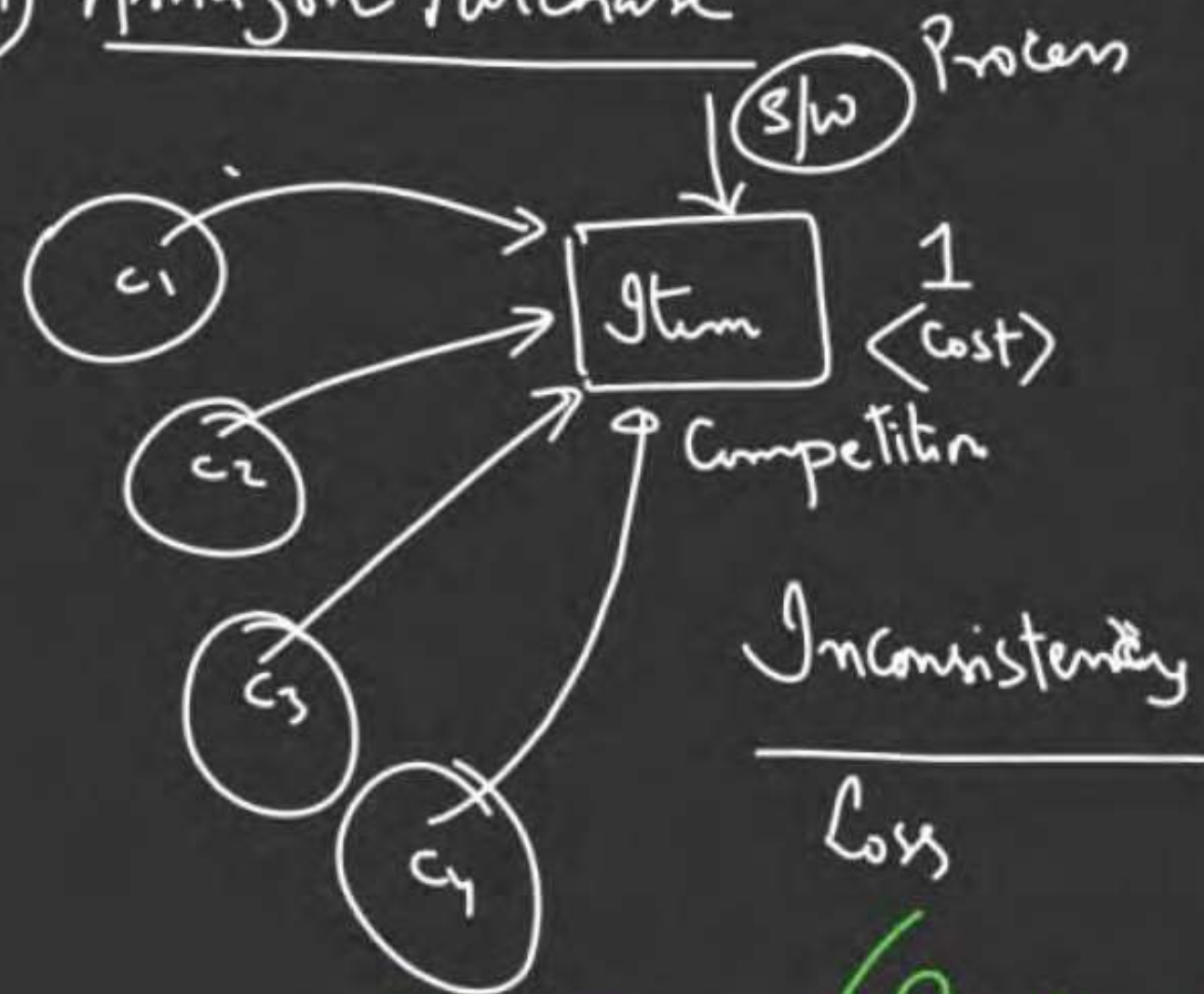
1) Competition: Two or more Processes are said to be in Competitive Synchronization, iff they compete/content for the accessibility of a shared resource;



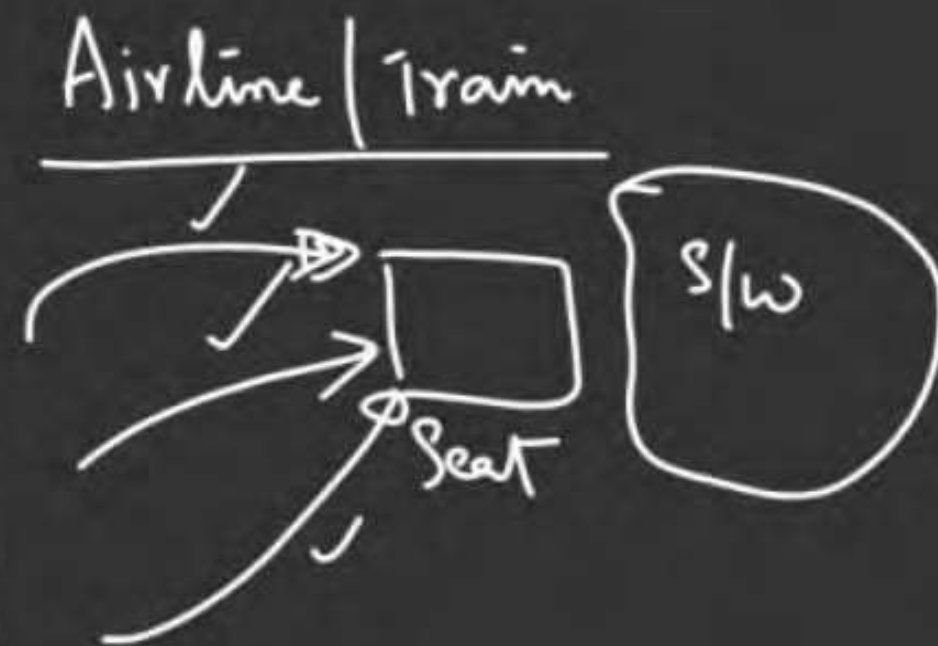
Final/Correct-value of $c = 5$



1) Amazon Purchase



2) Reservation System



→ Note: (Lack of Synchronization among competing (competition) Processes may lead to the problem of Inconsistency (or) Data loss;)

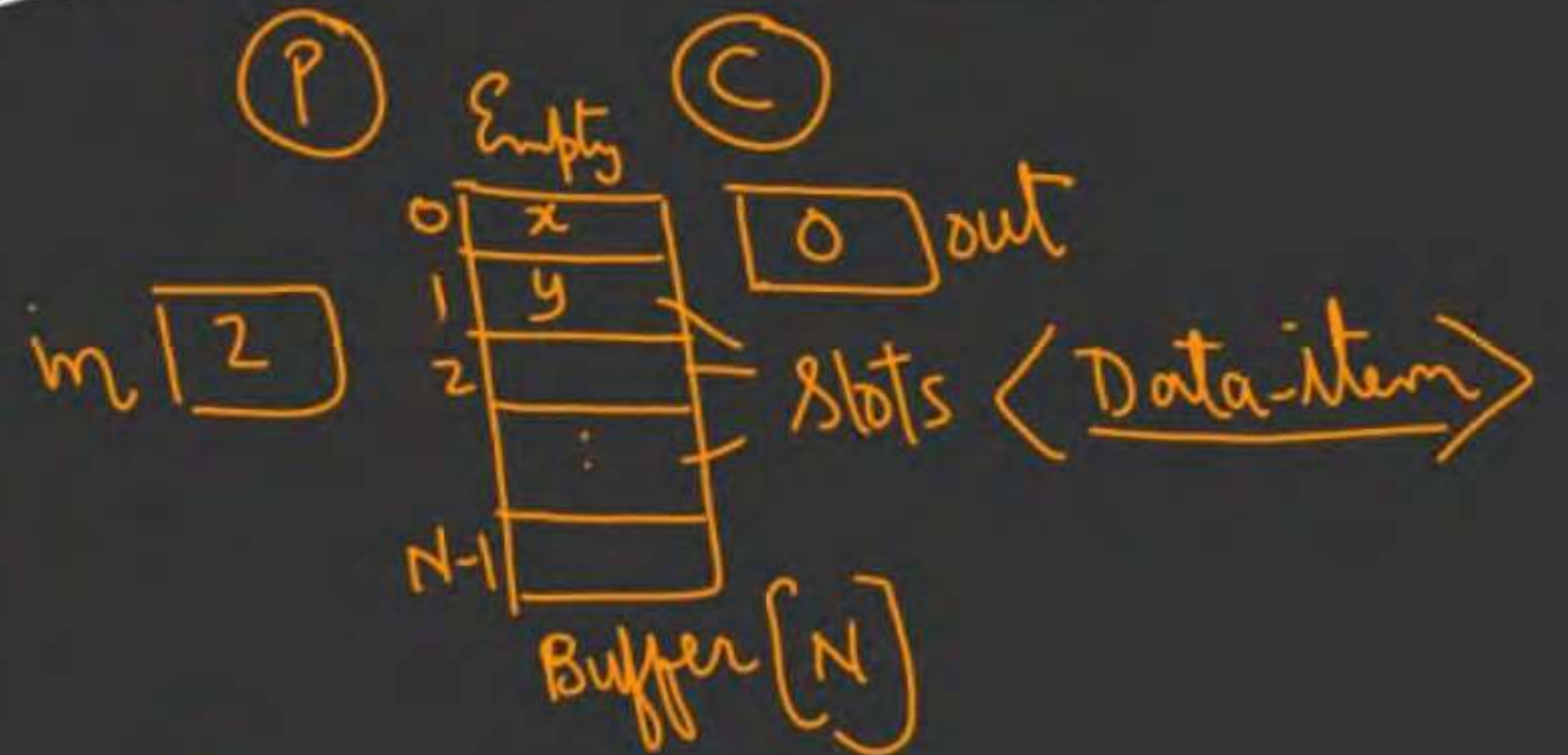
2) Cooperative Synchronization:

Two/more processes are said to be in Cooperative Synchronization, iff they get affected by each other;

"Execution of one Process affects the other"

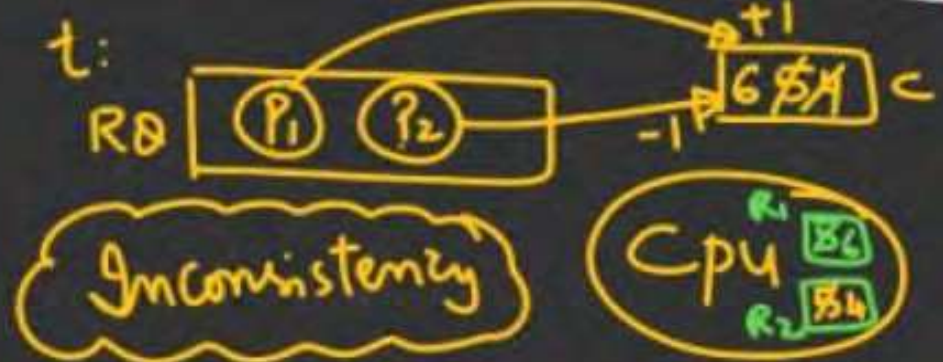
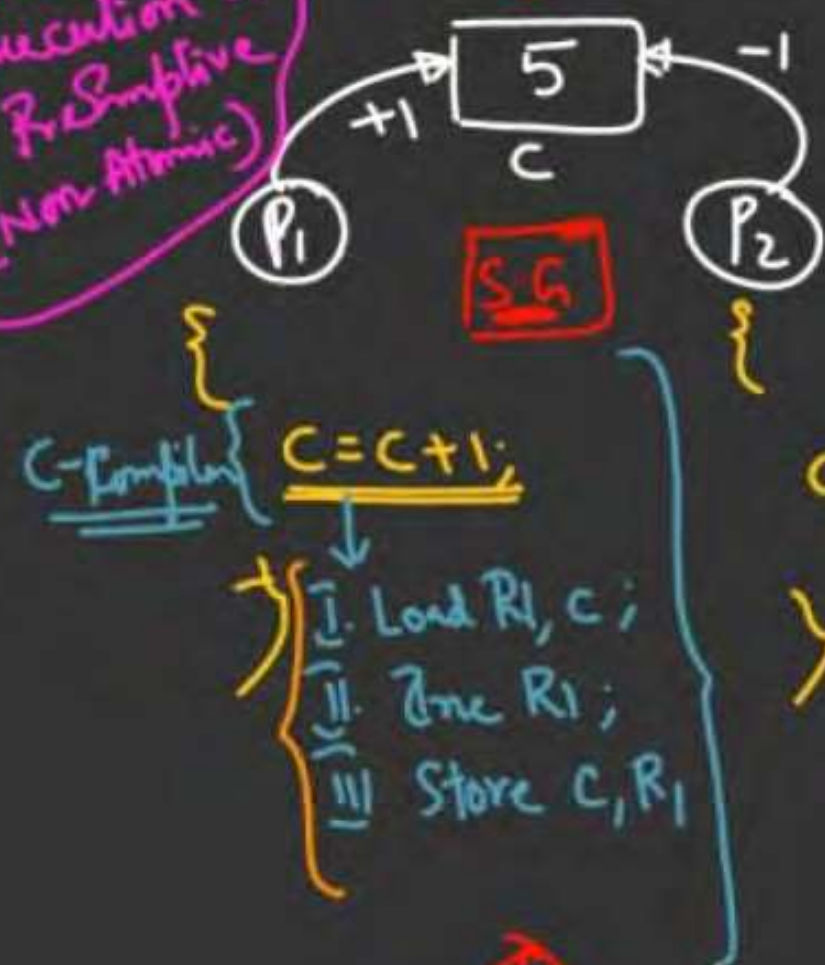
Ex: Producer-Consumer:

"Lack of Synch. b/w Cooperating Processes May lead to Deadlock"



Ex: to demonstrate Inconsistency in Competition Mode

User mode execution is Preemptive (Non Atomic)



Inconsistency

CPU

$t_1: P_1: I; II; IV$
 $t_2: P_2: I; II; III$
 $t_3: P_1: III$

$\frac{6}{4} \times 5$

User Process can get Preempted after completion of any instruction

need a Synch. Tool

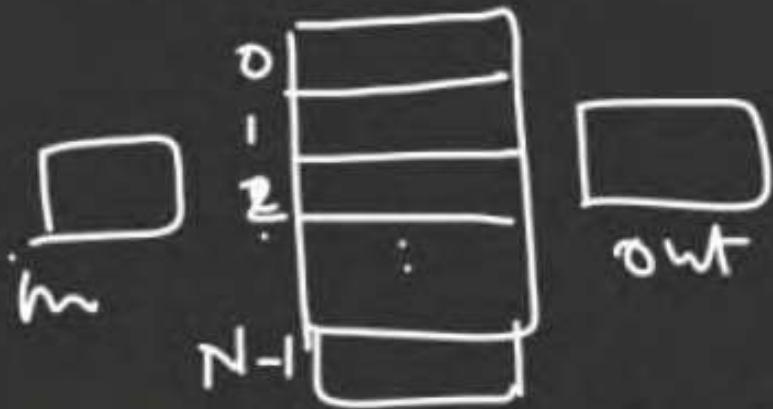
→ Inconsistency is Possible with Preemption during the execution of process;

→ In case of no Preemption during the execution, we always get correct result

Implementation of Producer-Consumer



(P) (C)



Circular Buffer(N)

(P) (C)



S.T.M

```
#define N 100
int Buffer[N];
int Count = 0;
```

```
void Producer(void)
{
    int item, in = 0;
    while (1)
    {
        1. item = Produce_item();
        2. while (Count == N);
        3. Buffer[in] = item;
        4. in = (in + 1) % N;
        5. Count = Count + 1;
    }
}
```

Inconsistency

Q: Is this Impl of (P) & (C) Correct/Incorrect?

```
void Consumer(void)
{
    int itemc, out = 0;
```

```
while (1)
{
    1. while (Count == 0);
    2. itemc = Buffer[out];
    3. out = (out + 1) % N;
    4. Count = Count - 1;
    5. Process_item(itemc);
}
```

Comp

Necessary conditions for Synch. Problems to occur in IPC Environment

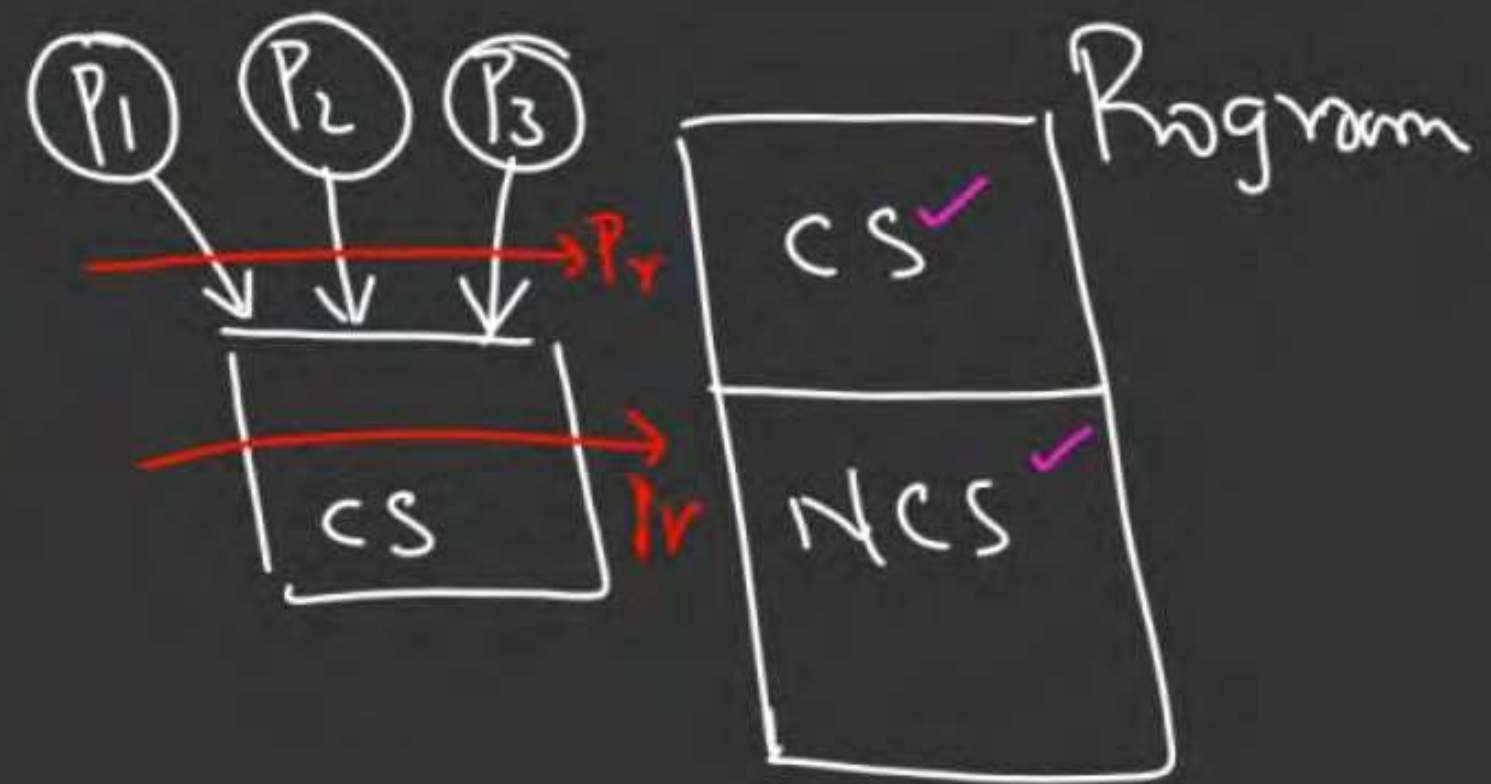


1. ✓ Critical Section (CS): is that part of the Program where Shared resources are accessed < Terminology >

→ Non-CS / other Section: is that part of the program which does not access shared resource;

2. Race Condition:

Situation wherein multiple Processes tries to access shared resource (CS) & the final outcome depends on the order in which they finish;



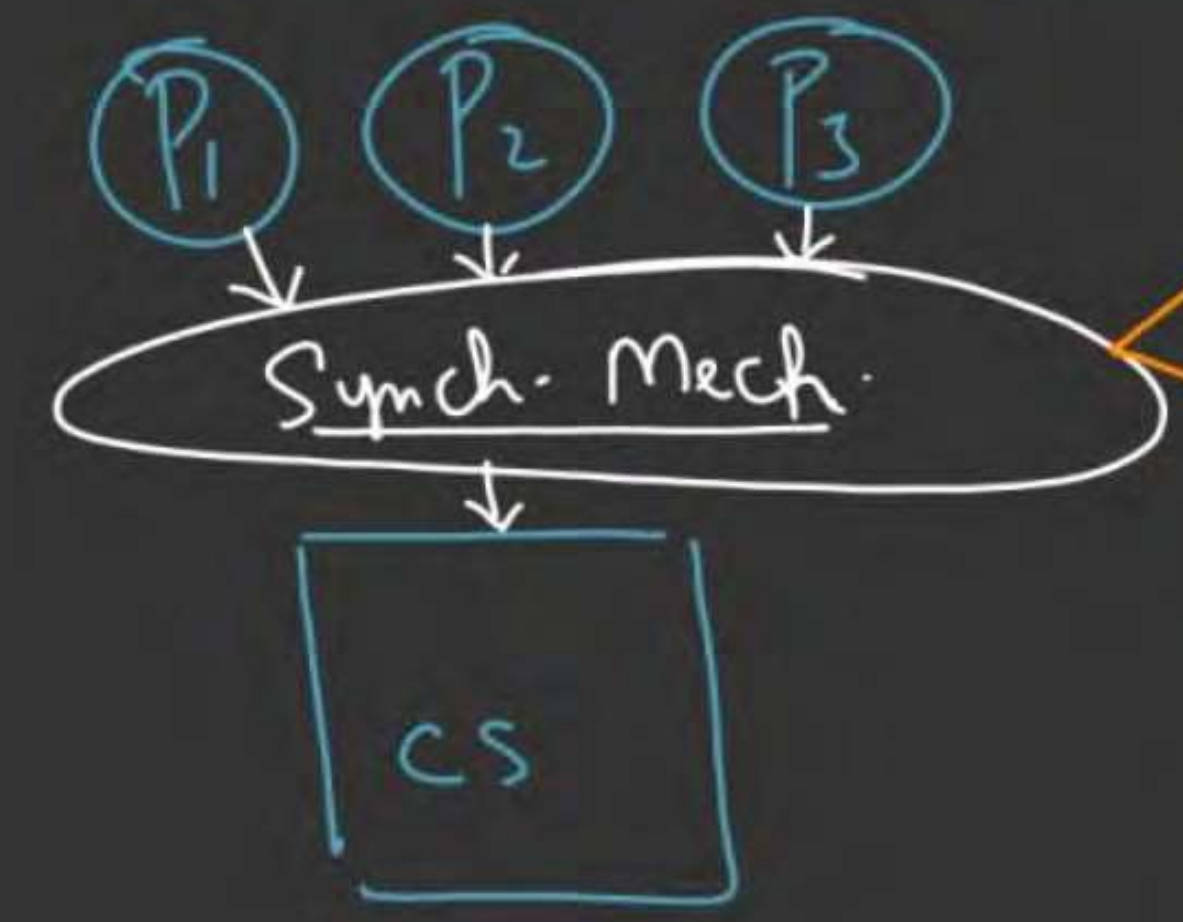
3. Pre Emption



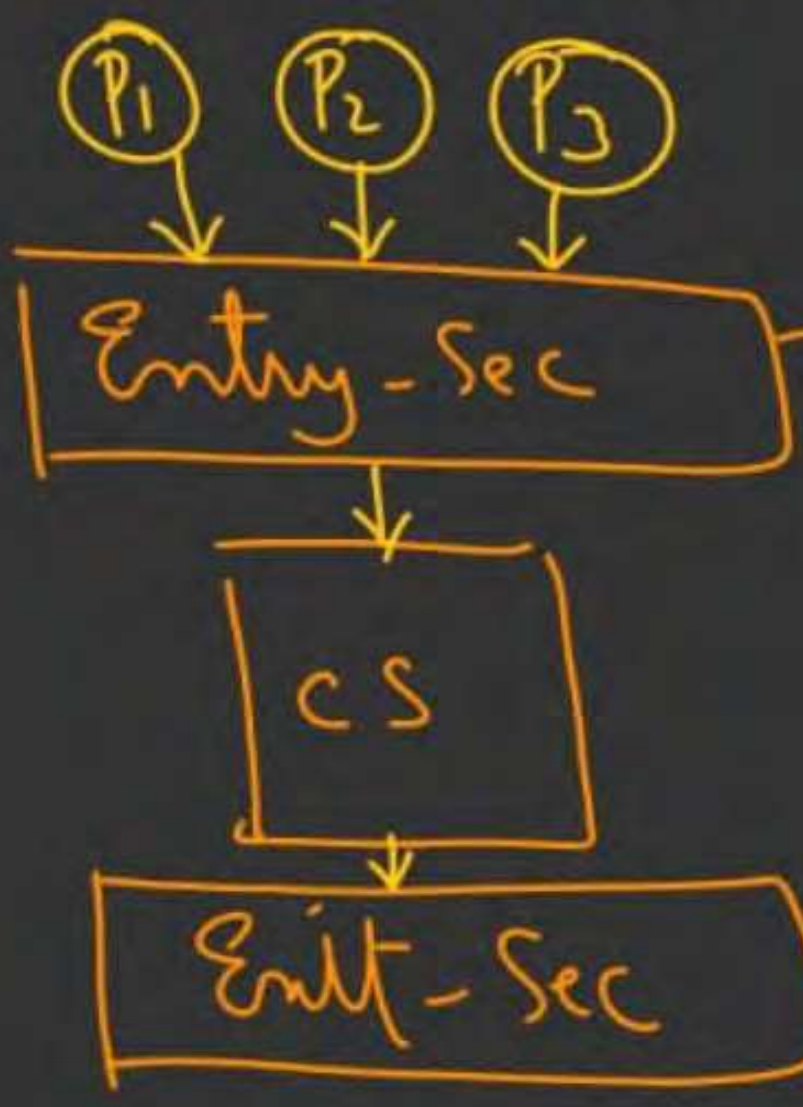
Name of this problem: Critical Section Problem

CS - Problem

11:45 am



Entry-Section
Exit-Section



Require ments
Solu: Synch. Mech.
Algorithm Tool

Architecture
Model

of Synch Tool / CS Problem