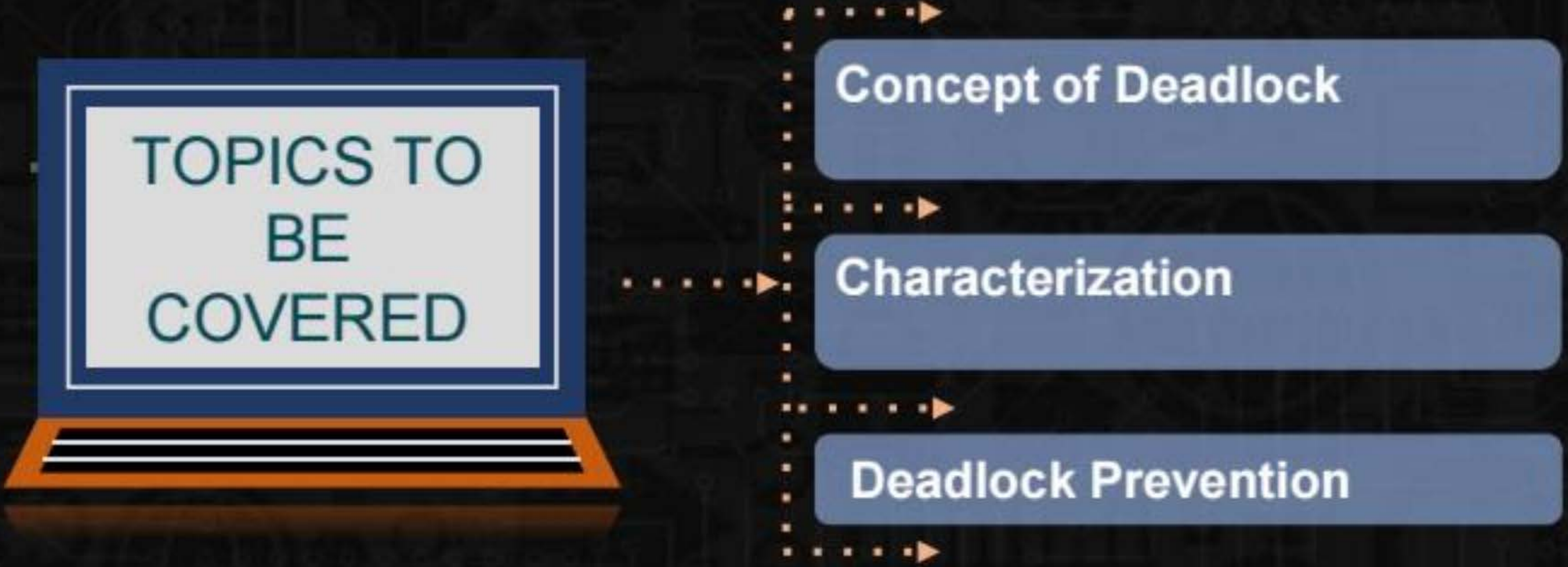
CS & IT ENGINEERING

Operating Systems

Deadlock

Lecture No. 1

By- Dr. Khaleel Khan Sir



TOPICS TO BE COVERED

- Concept of Deadlock
- Characterization
- Deadlock Prevention

Deadlock Concept:

→ Two/more Processes are said to be in deadlock iff they wait for the happening of an event which will never happen.

Consequences:

→ Thruput/Efficiency drops

→ Ineffective utilization of Resources;

→ deadlock is therefore undesirable

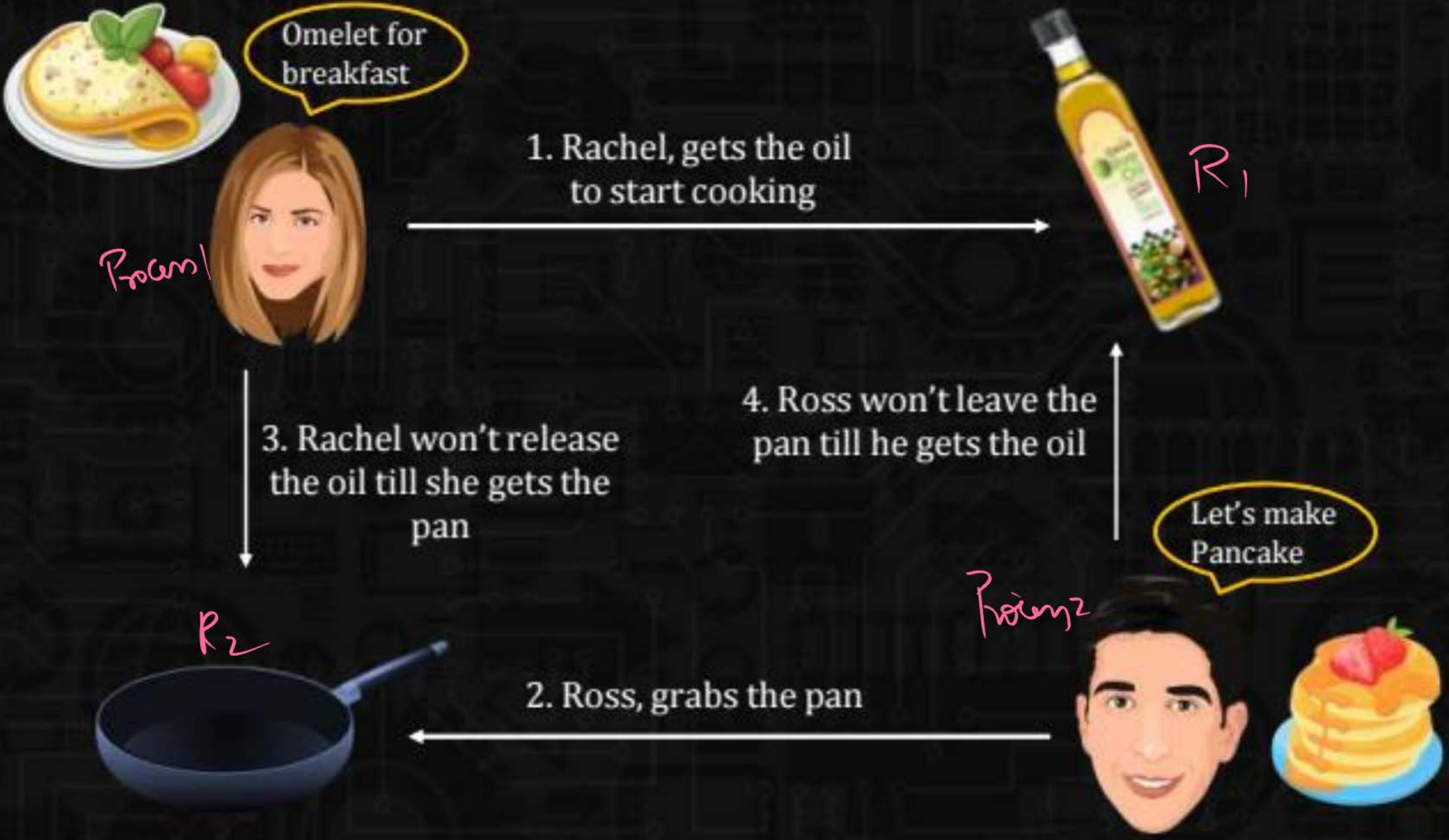
Deadlock vs Starvation

Deadlock → (Infinite Blocking of Processes)

(forever)

Starvation → (Indefinite Blocking)

Let’s see some real-life examples of Deadlock !



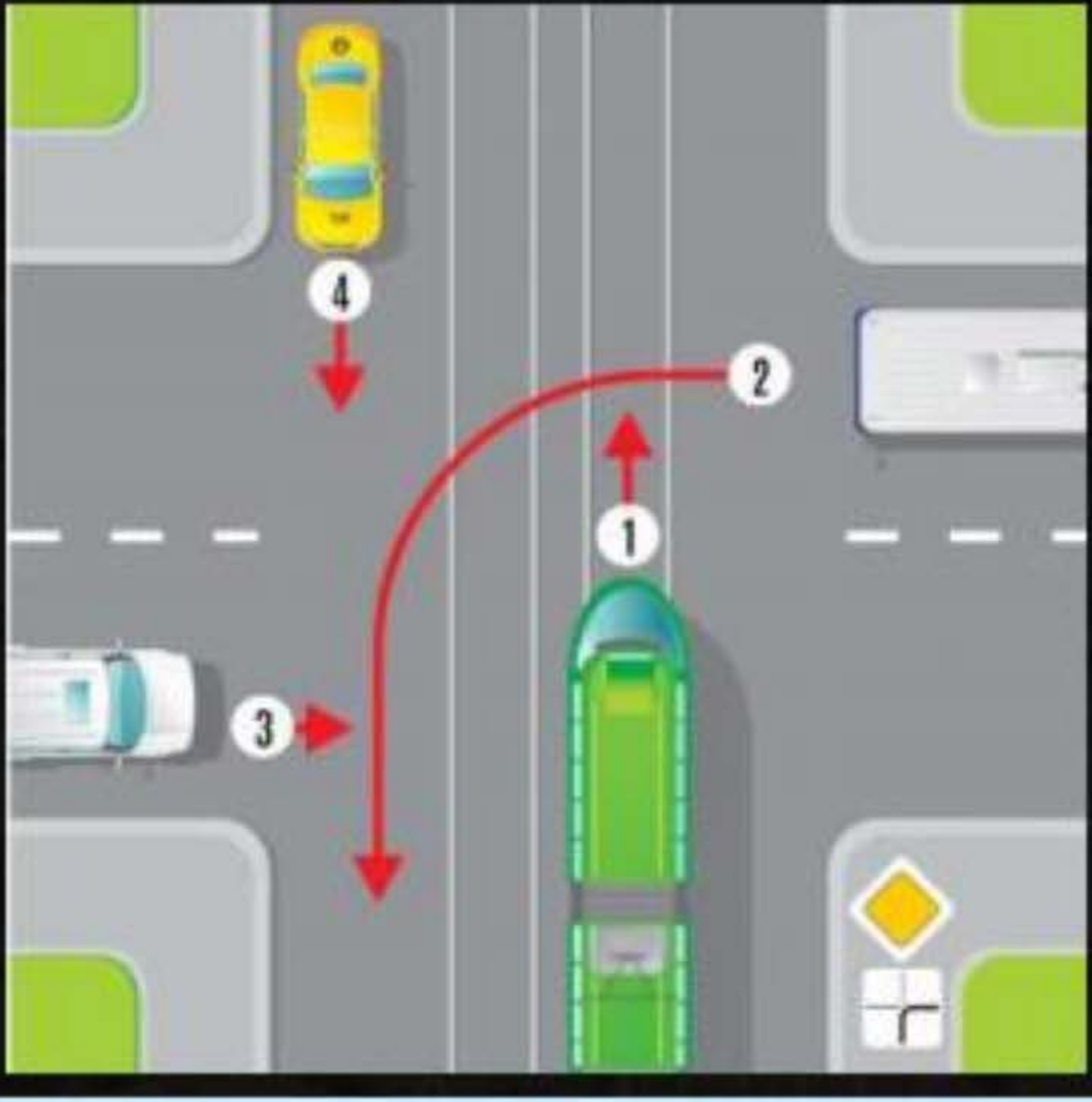
1. Rachel, gets the oil to start cooking

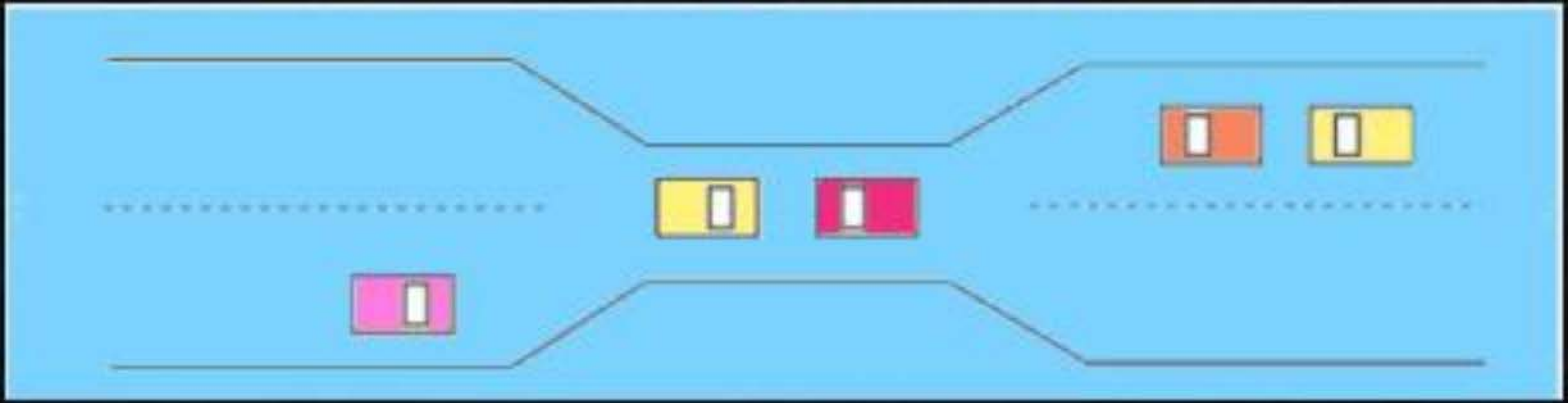
2. Ross, grabs the pan

3. Rachel won't release the oil till she gets the pan

4. Ross won't leave the pan till he gets the oil

Cross Road Traffic Jam


4
2
1
3



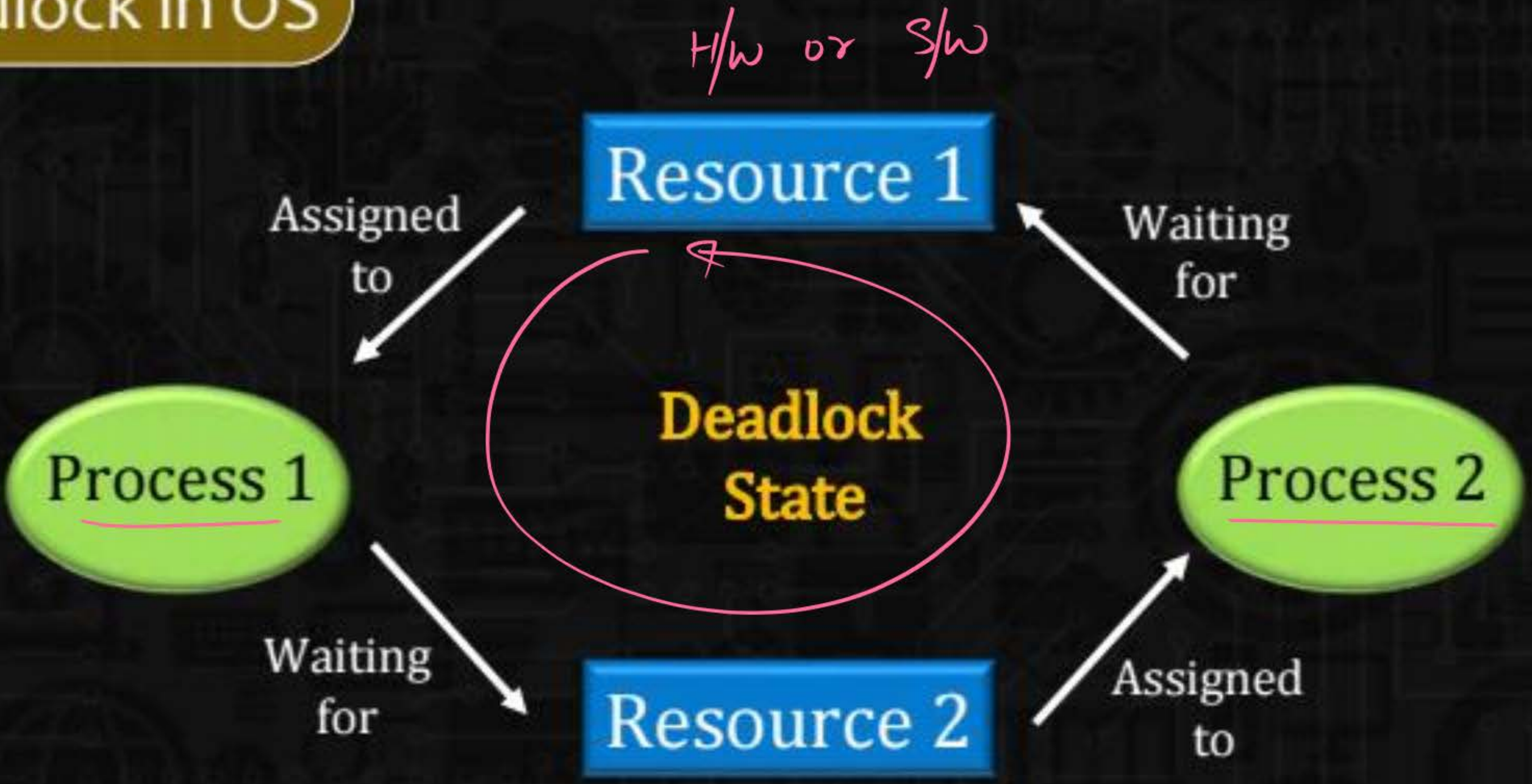
Ross! You release the lock first!
Once I have finished my task,
You can continue.

Rachel!! Why Should I?
You release the lock first
And wait until
I complete my task.

(Adamancy)

AMPUS PLAC
IMPERIAL COLLEGE OF ERING NEW
Explain Deadlock, and we'll hire you.
Hire me, and I'll explain it to you.
let's discuss the salary.

Deadlock in OS



System Model :

→ $\underline{n}$: no. of Processes

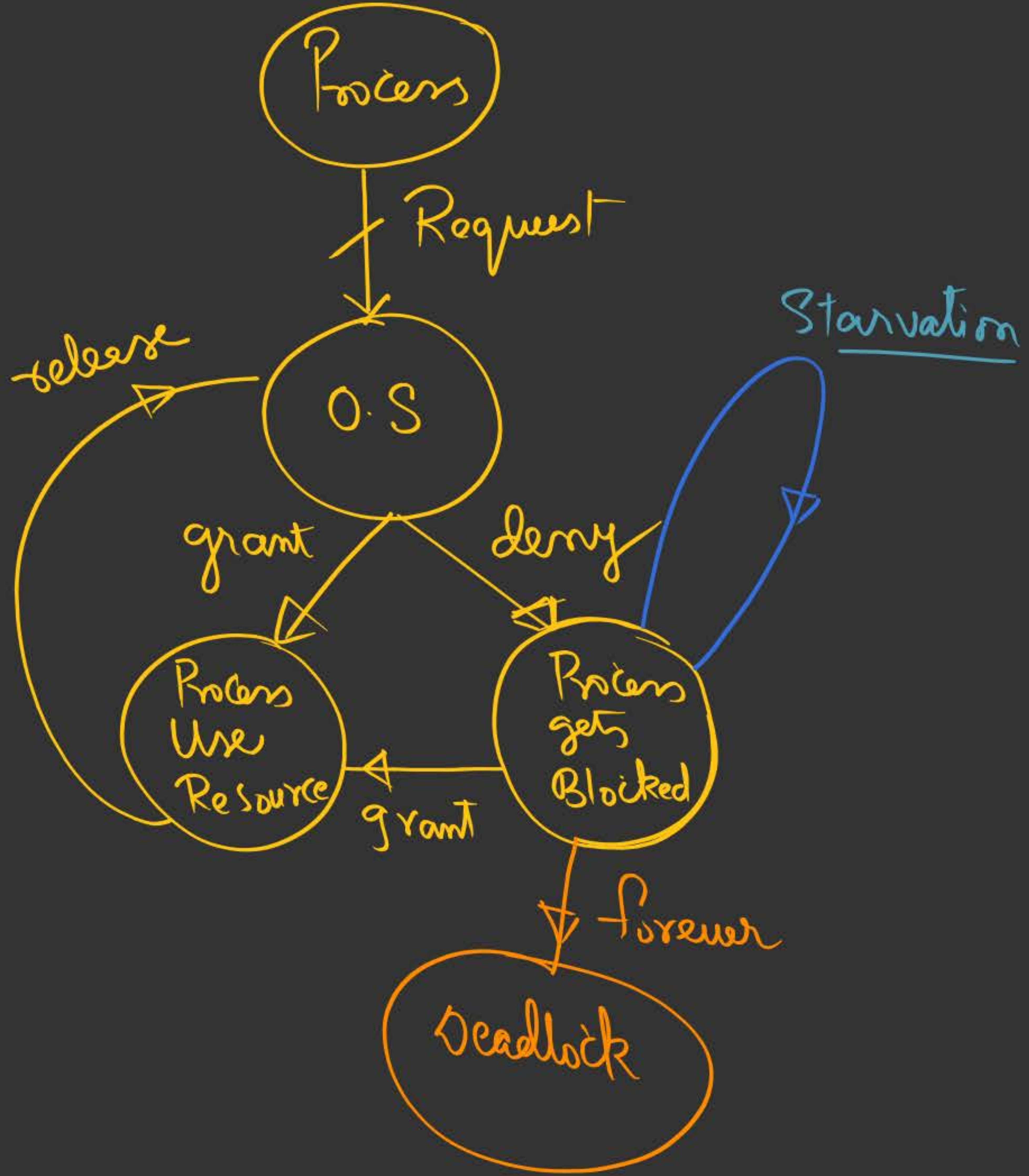
$\langle P_1 - - - - P_n \rangle$

→ 'm' : Resources

$\langle R_1 - - R_m \rangle$ < H/W, S/W

Single Instance (Copy)

Multi-Instance (Copies)



Deadlock Characterization

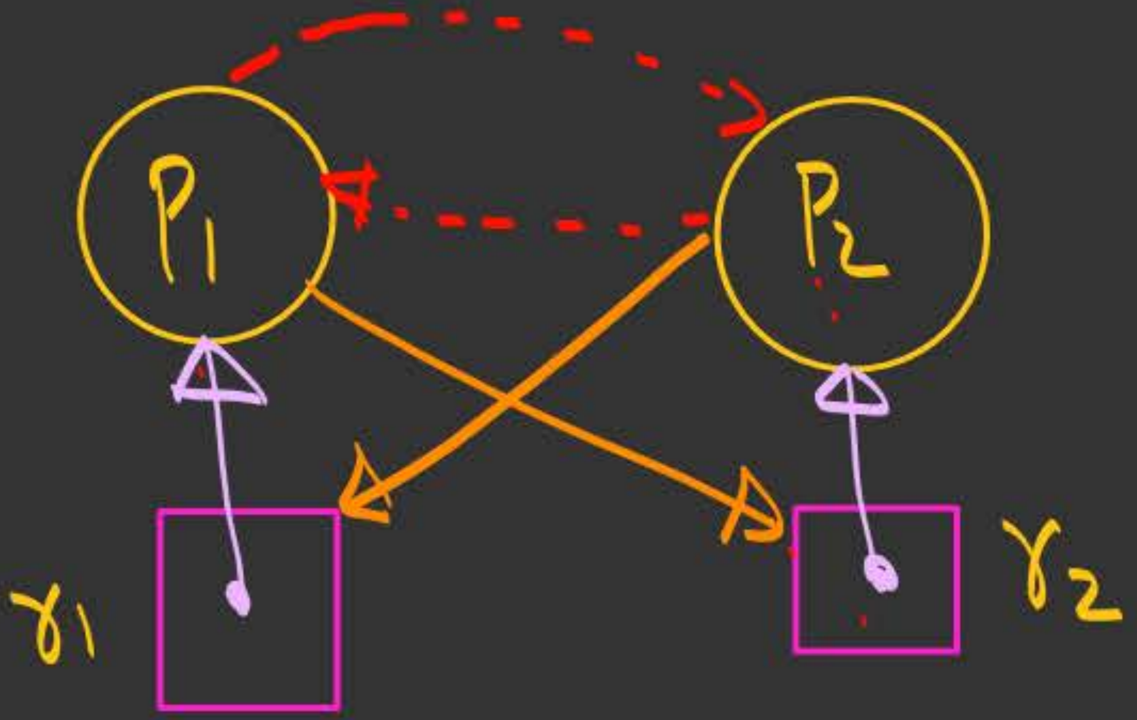
I. (Necessary Condition):

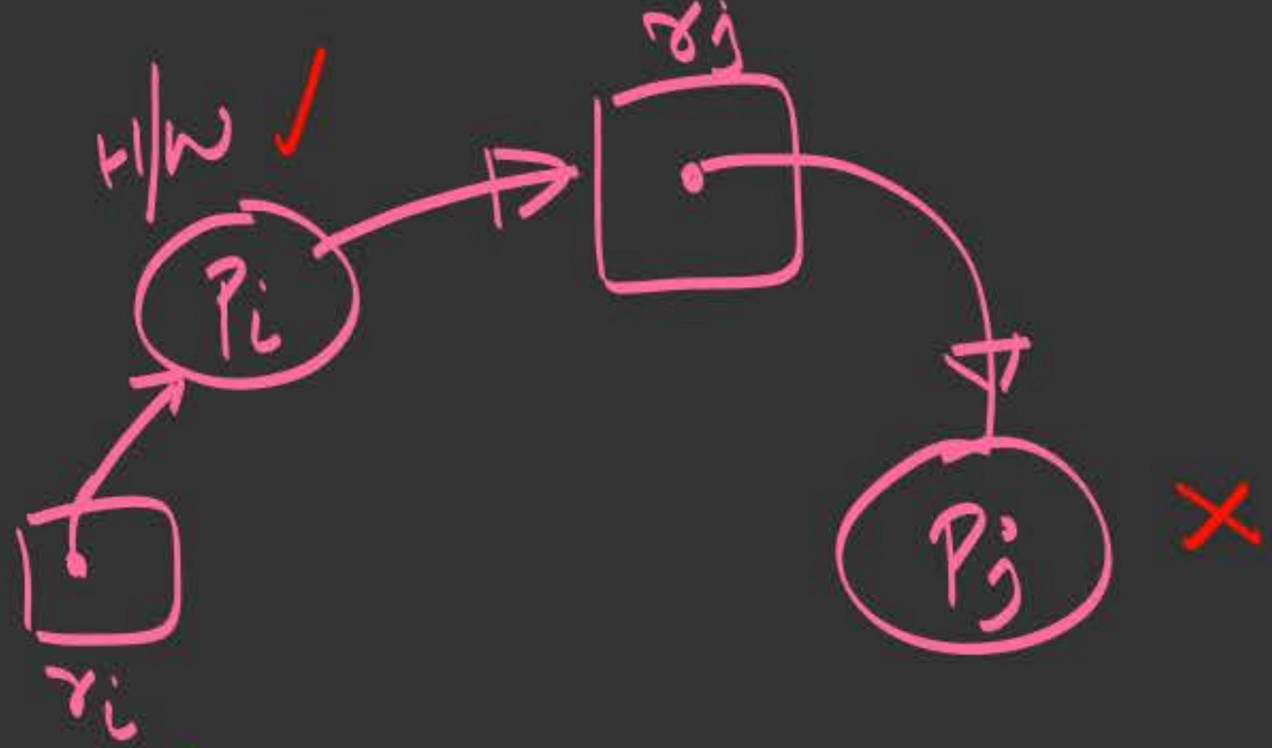
✓ a) Mutual Exclusion:

→ Shared Resources

<CS>

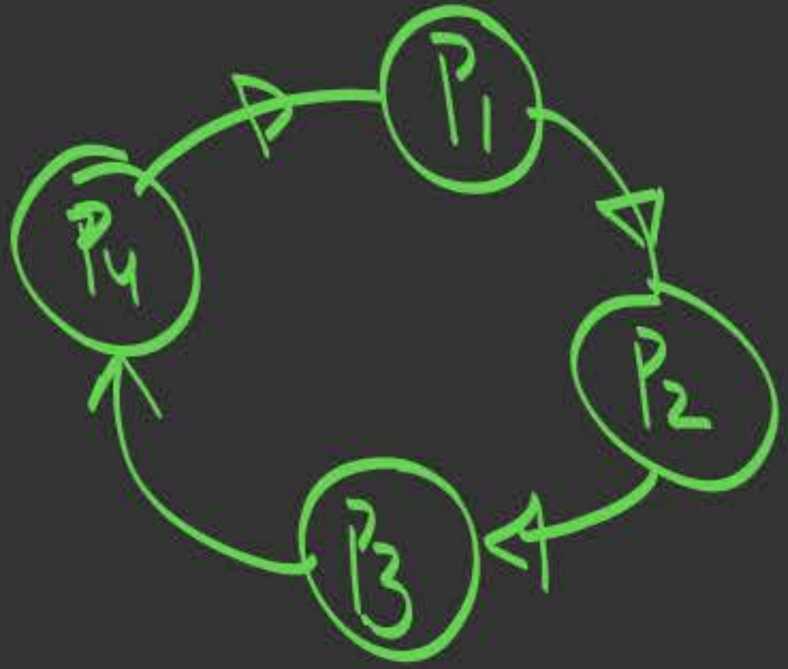
✓ b) Hold and wait:





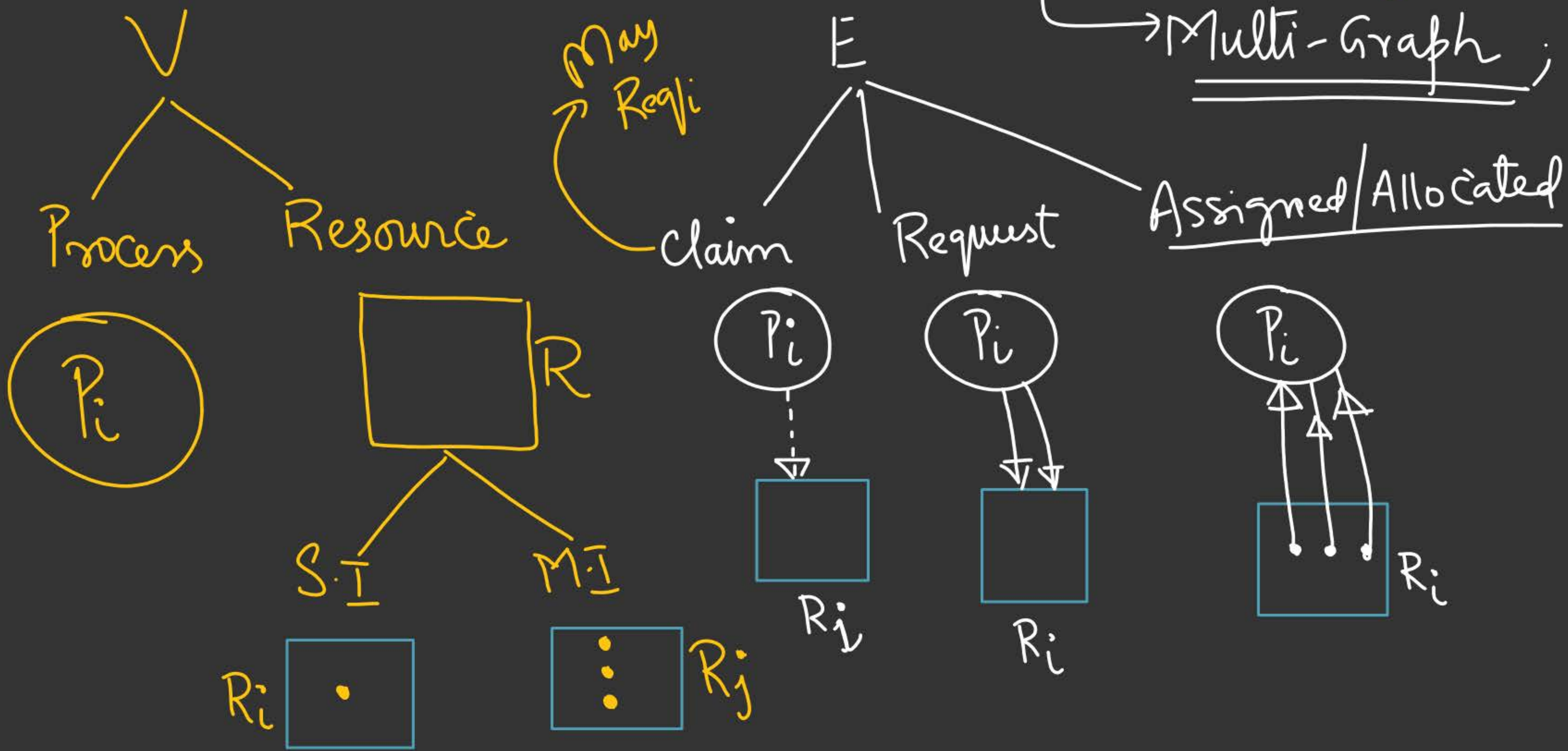
✓ c) No-Pre Emption: (of resources)

✓ d) circular wait:

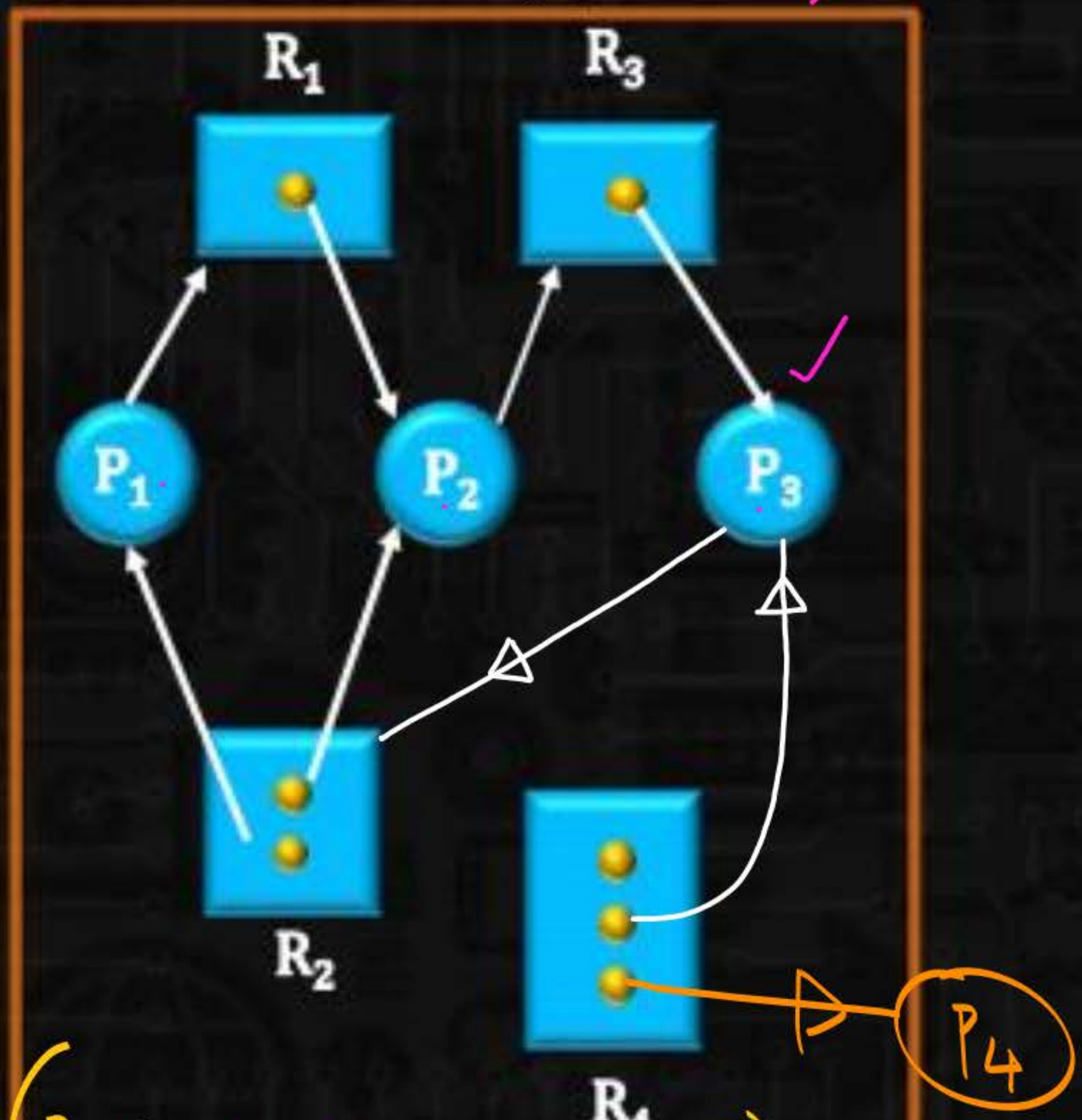


Resource Allocation Graph (R.A.G)

$G = (V, E)$

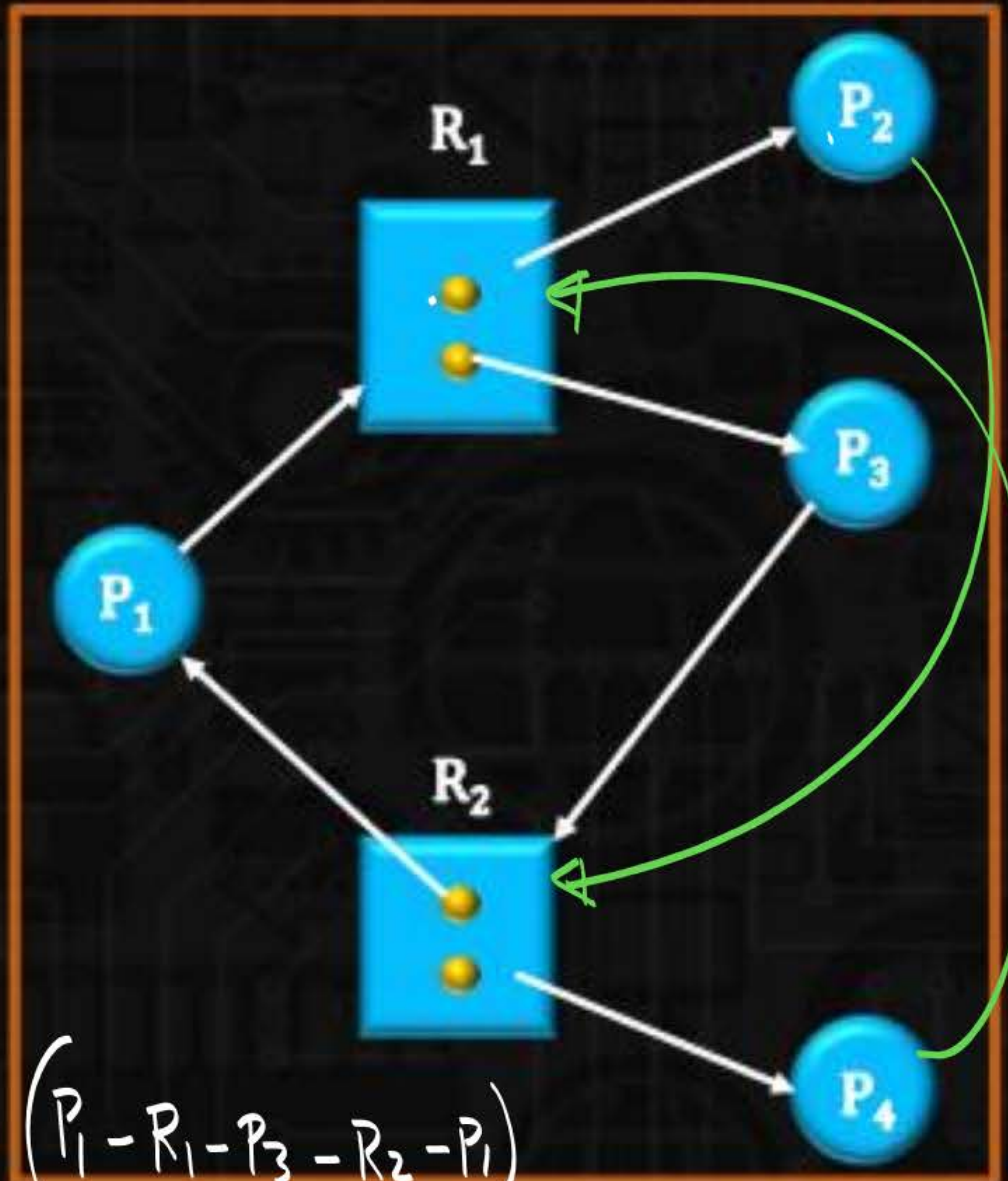


G1 (System)



(P1 - R1 - P2 - R3 - P3 - R2 - P1) cycle

G2 : No-deadlock



(P1 - R1 - P3 - R2 - P1)

't'

t'

Now deadlo ch

Deadlock Handling Strategies

1. Deadlock Prevention
2. Deadlock Avoidance → Banker's Algo [Dijkstra's Algo]

T1 : deadlock Never occurs;

3. Deadlock Detection & Recovery → * (Doctor's Algo)
4. Deadlock Ignorance (No - Strategy) → (Ostrich Algo)

T2 : deadlock definitely occurs;

I. deadlock - Ignorance (Ostrich Algo.)

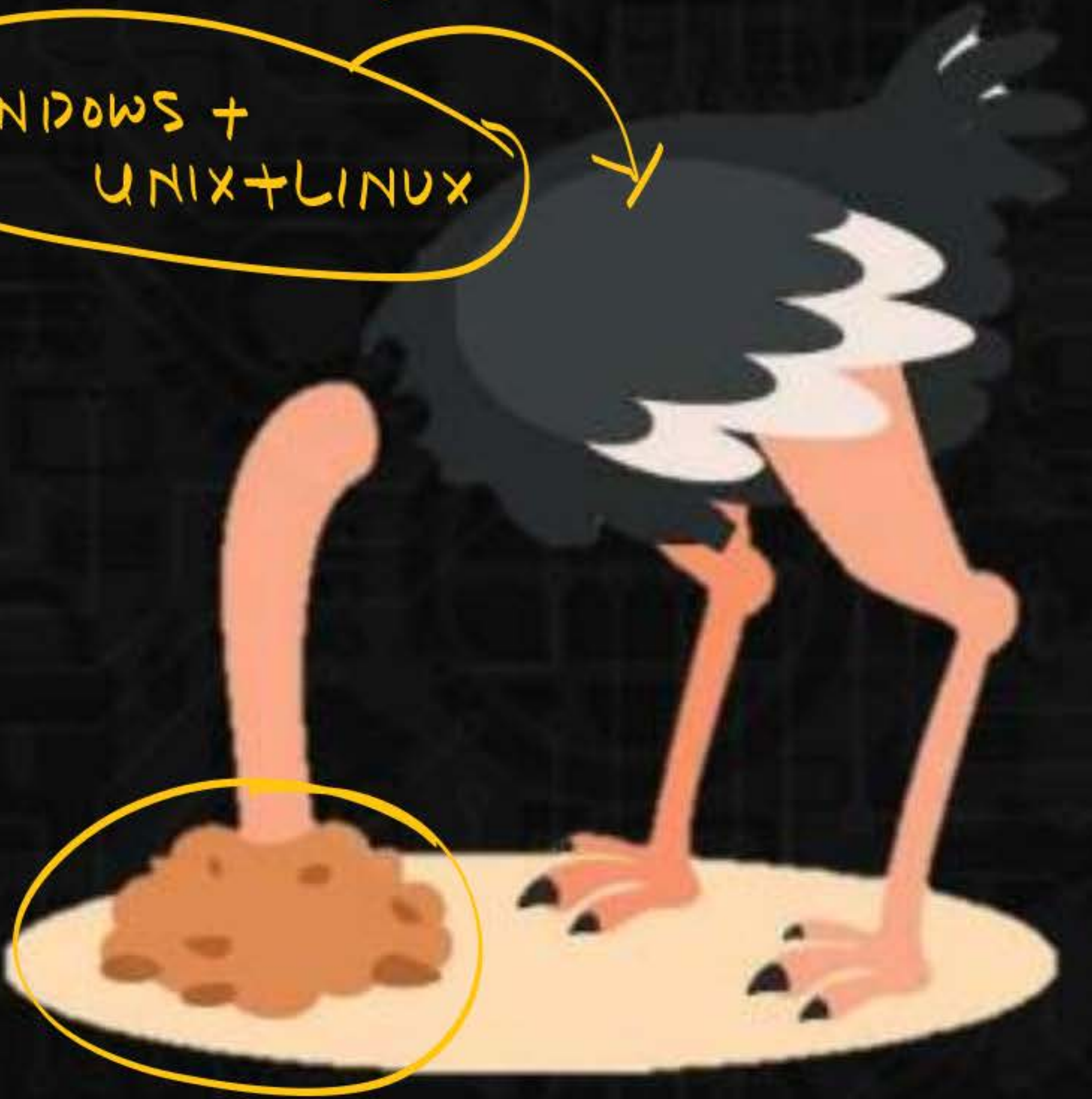
(No - Strategy)

The Ostrich Algorithm

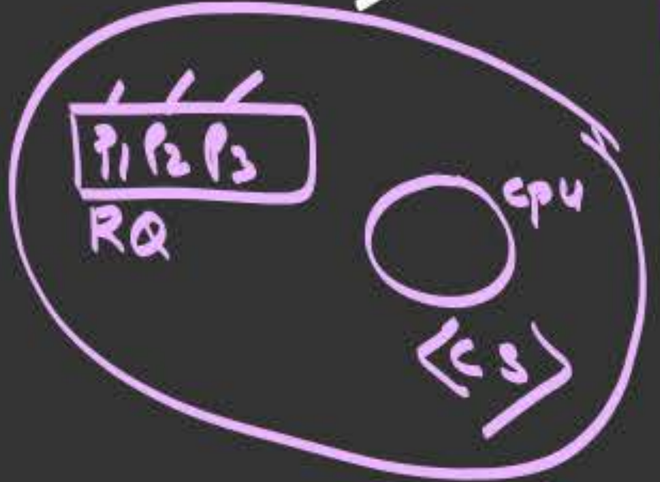
Ignorant Strategy 9

- ❑ Pretend there is no problem
- ❑ Reasonable if
  - ❖ Dead locks occur very rarely
  - ❖ Cost of prevention is high
- ❑ UNIX and Windows take this approach
- ❑ It is a trade-off between
  - ❖ Convenience ✓
  - ❖ Correctness ✓

WINDOWS + UNIX+LINUX



II. Deadlock Prevention: (By dissatisfying/Negating one/more of the Necessary conditions)



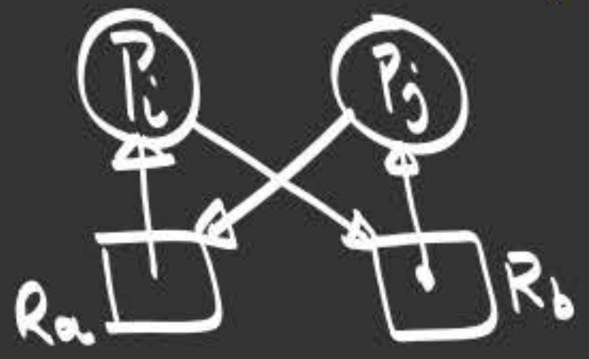
a) Mutual Exclusion:

- Since every M.Pr Env. has atleast one Shared resource, ∴ M/E is Non-Dissatisfiable;

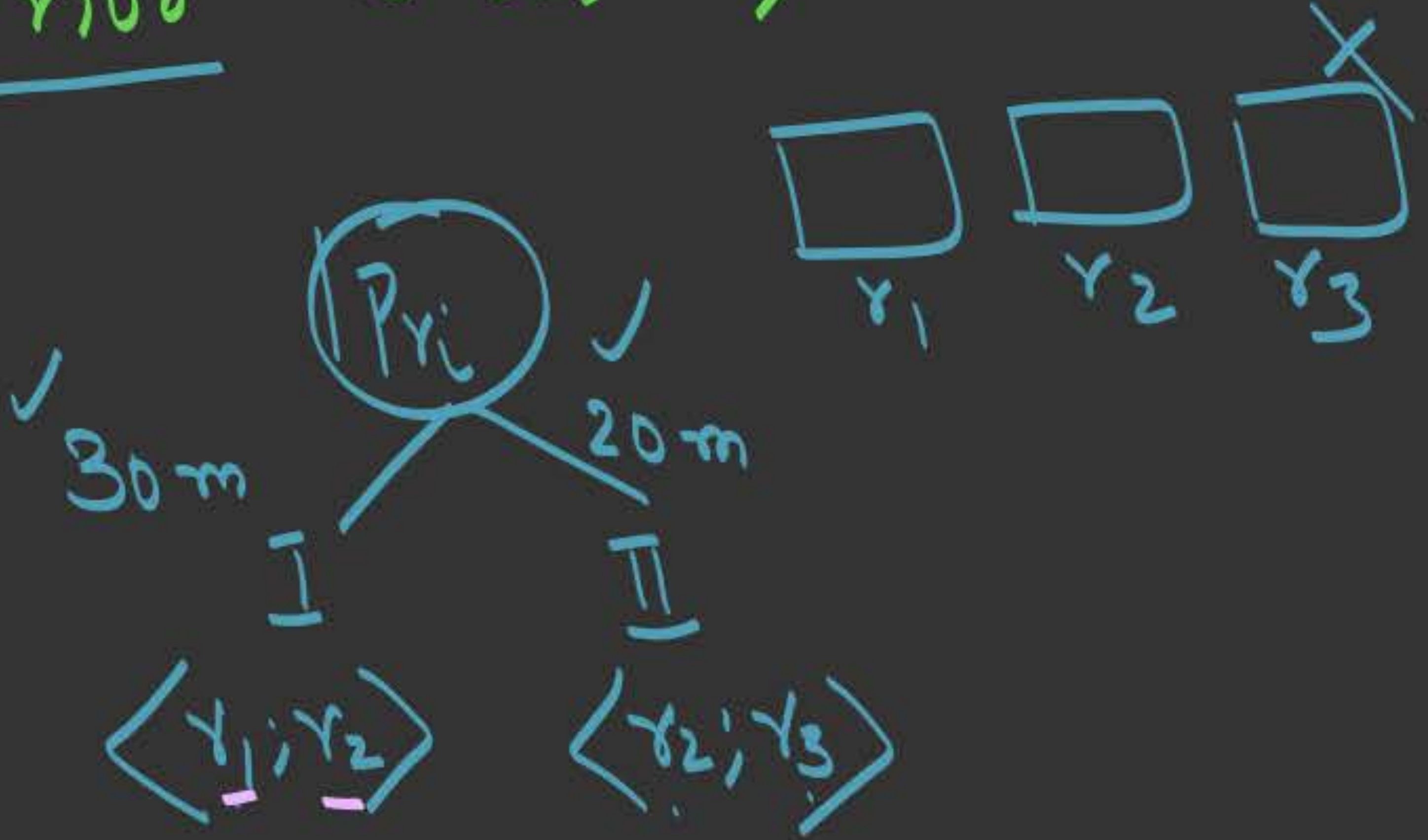
(ii) Process must release ALL resources b/f making a fresh/new request;

→ Starvation
→ No inefficiency

b) (Hold and wait): Hold (or) wait

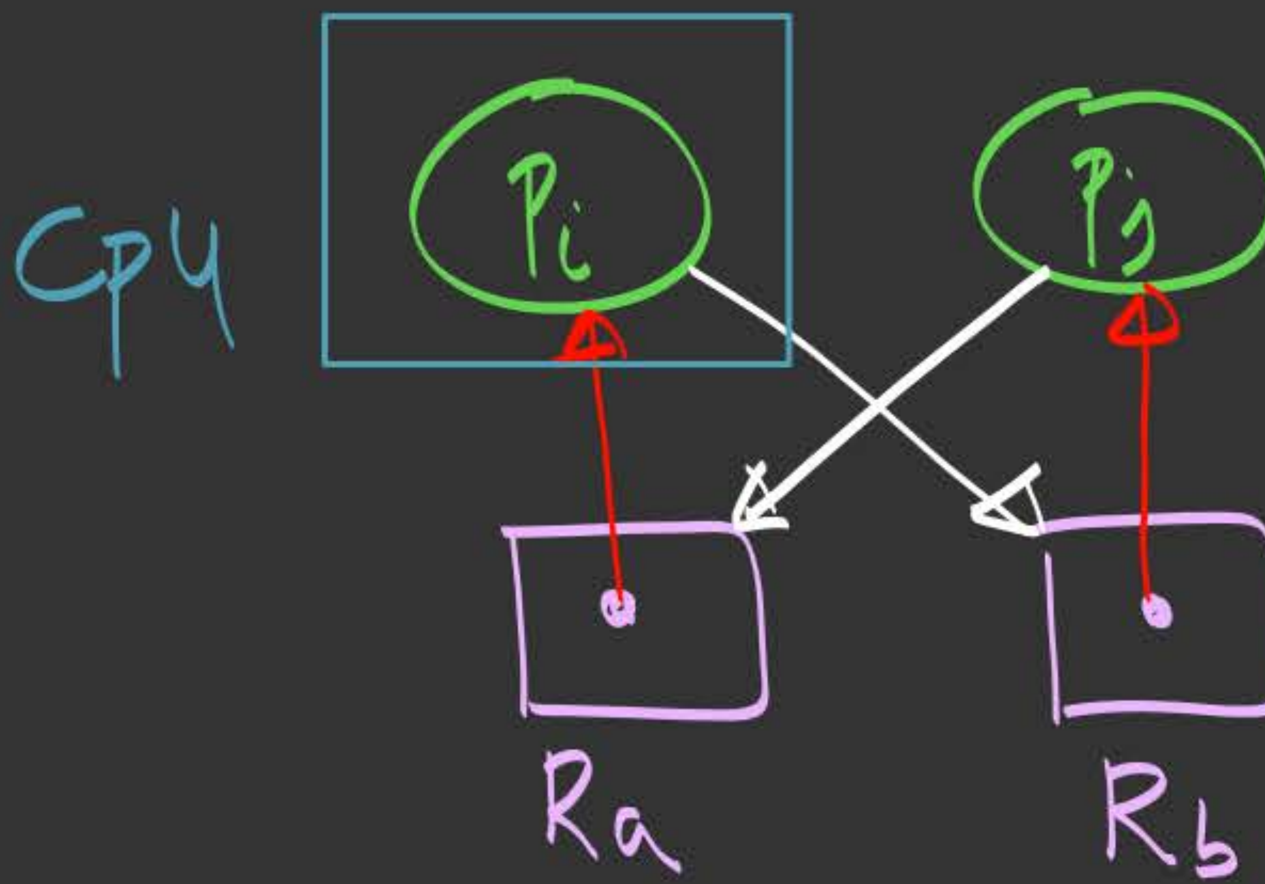


1) Process must request & be allocated ALL resources Prior to its start;



((i) Starvation
(ii) Inefficiency)

c)! (No-PreEmption) = PreEmption of Resources from Processes;



Forceful

(Running Process Should Not Block)

Self (Selfless)

(Other Processes Complete first)

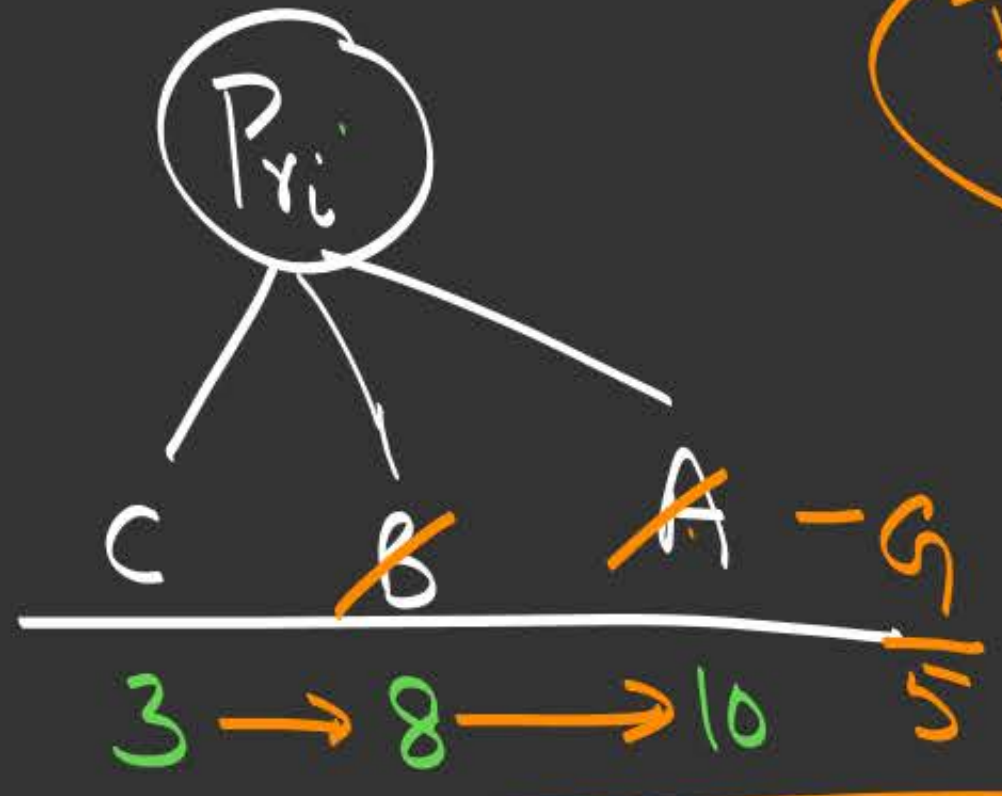
< Starvation >

d) Circular-wait: is Prevented by a total order (Imc) Relation among all processes & resources;

I) Assign unique No's to each Resource;

II) Never allow a Process to request a Lower numbered Resource than the last one allocated;

Resources	Res-Id
A	10
B	8
C	3
D	4
E	12
F	20
G	5



3 → 5 → 8 → 10

Resource PreEmption

Starvation

H/w

Max → 10

Min → 2

THANK YOU
GW
SOLDIERS !