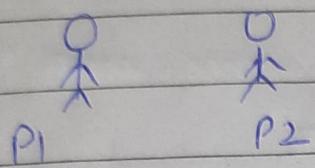
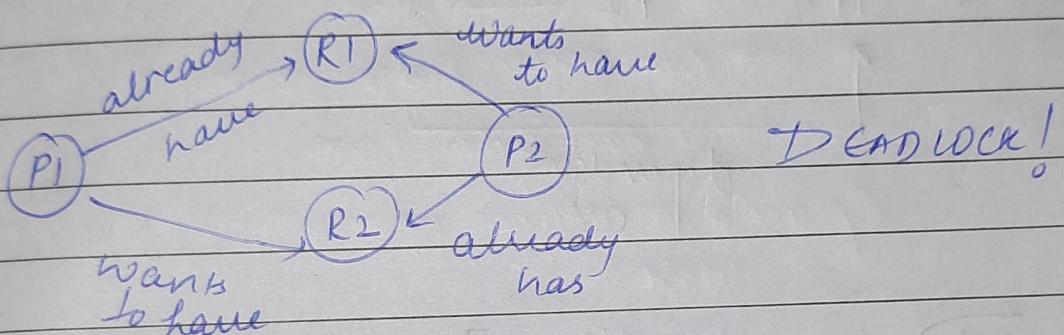


# Deadlock Introduction



P1 says P2 leave resources  
after you!

P2 says P1 leave resources  
after you!

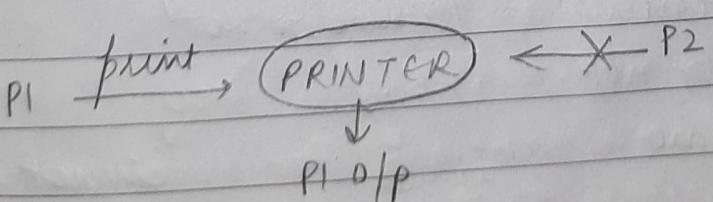


## Conditions of a Deadlock

- \* Mutual Exclusion
- \* No preemption
- \* Hold & Wait
- \* Circular wait

## Mutual Exclusion

- access of Resources must be mutually exclusive
- Resources are in non-shareable mode



## No preemption

- A resource can't be snatched from a process until released by the process itself.

## Held and Wait

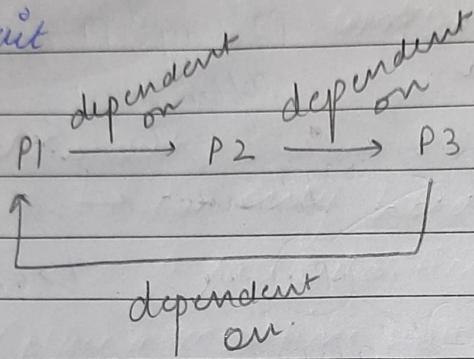
- If P1 needs 10 and has held 3 resources

DATE: / /  
PAGE:

→ P1 should hold 3 Resources

while waiting for + other resources to come.

## Circular wait



## Deadlock + Handling Methods

(#)

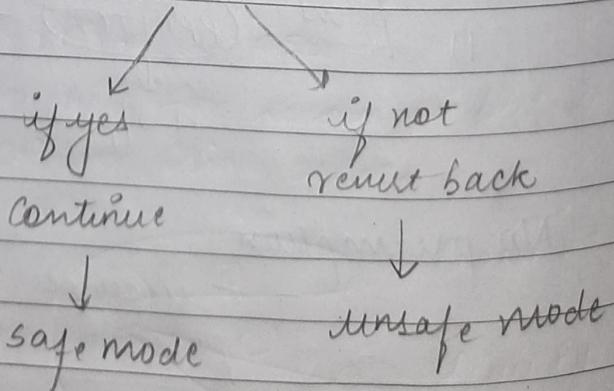
### Prevention of deadlock

- design system such that deadlock never occurs.
- Try to violate any of the four conditions

(#)

### Avoidance of deadlock

- whenever a process needs a resource system takes decision of enter. based on some data. → check if giving the resource possible or not



## # Detection and Recovery of deadlock

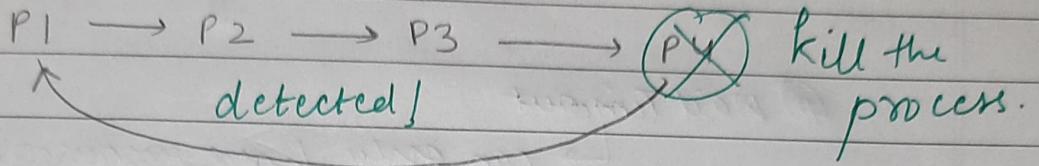
- decision after deadlock occurs

DATE: / /

PAGE:

detect deadlock

Recover from deadlock



Recovery ↪ kill process (inefficient)  
Recovery ↪ Resource pre-emption

## # Ignorance of deadlock

↳ Ignoring deadlock

→ Isal mei ek bar  
ayega

Recovery: ← System restart ←

## Deadlock Prevention

Condition ① Mutually exclusive

For deadlock, resources must be non-shareable and to prevent deadlock we can make resources shareable.

BUT we can't change it (property of resource)

Can't be violated

## Condition ⑩ Hold and wait

When a process enters → ~~provide all resources~~  
then only ←  
start

DATE: / /  
PAGE:

BUT inefficient ⇒ many processes will keep waiting

## ⑪ Wait Timeout

→ can only hold resources for a particular time

(place a max-time upto which a process can wait)

→ After which process must release all the resources.

## Condition ⑫ Non pre-emption

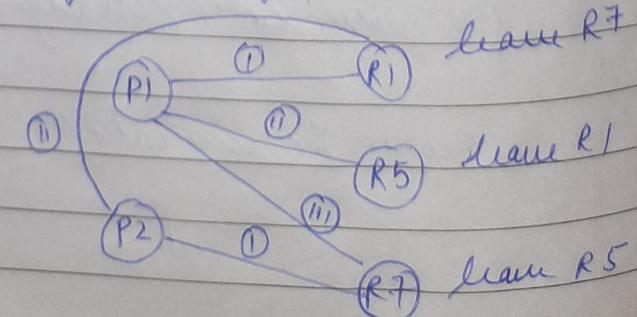
Based on priority [ we allow a process to forcefully preempt the resource held by other process  
→ preferably wait state process. ]

## Condition ⑬ Circular wait

P1 → P2  
P2 → P3  
P3 → P4  
P4 → P1  
- Resources are numbered R1, R2, R3, ..., R50

- Process can ask for resources but in either Pending or Using order.

Process	Need
P1	R1 R5 R7
P2	R7 R1



## Numerical on Banker's Algorithm

Process	Allocated			Request		
P0	X	Y	Z	X	Y	Z
P0	1	2	1	1	0	3
P1	2	0	1	0	1	2
P2	2	2	1	1	2	0

All resources X, Y and Z are available 5 units each.  
Which process is executed last?

First let's compute how many units of X, Y & Z are available after allocation.

$$\text{Total units of } X \text{ allocated} = 1 + 2 + 2 = 5$$

$$\text{Total units of } Y \text{ allocated} = 2 + 0 + 2 = 4$$

$$\text{Total units of } Z \text{ allocated} = 1 + 1 + 1 = 3$$

$$\text{Units of } X \text{ available} = 0$$

$$Y = 1$$

$$Z = 2$$

Now P0 requires 1 unit of X  
0 units of Y  
and 3 units of Z

Not available

to us



P0 needs to wait.

On the other hand P1 req. are

satisfied!  $\Rightarrow$  P1 will run

$\Rightarrow$  Release resources.

Units of X available now

$\Rightarrow$   $(X \text{ units allocated} + X \text{ units requested})$

$$\Rightarrow 2+0 \Rightarrow \underline{\underline{2}} \text{ units}$$

## Similarly

Units of  $\gamma$  available  $\Rightarrow 0+1=1$

Units of  $z$  available  $\Rightarrow 1+2 = 3$

$$\text{for P2} \quad \begin{array}{l} \text{units of } X \text{ needed} = 1 \\ \hline Y = 2 \\ Z = 0 \end{array} \quad \Rightarrow \quad \begin{array}{l} \text{Can't run} \\ \text{as } Y \text{ units} \\ \text{are not} \\ \text{complete.} \end{array}$$

$$\text{for } P_0 \quad \begin{array}{c} \text{units of } X \text{ needed} = 1 \\ \hline y - = 0 \\ \hline z - = 3 \end{array} \quad ] \text{ satisfied!}$$

Po will sun ✓

Release resources ✓

$$\begin{array}{rcl}
 \text{Units of } X \text{ available} & \Rightarrow & 1 + 1 + 1 = 3 \\
 \hline
 Y & \Rightarrow & 3 \\
 \hline
 Z & \Rightarrow & 4
 \end{array}$$

Check for P2  $\Rightarrow$  Reg. Satisfied!

$\Rightarrow P_2$  will run ✓

P2 releases resources ✓

Units of X available = 5

$$\underline{4} = 5$$

$$x = 5$$

LAST PROCESS EXECUTED  $\Rightarrow$  P2 Ans