CS & IT ENGINEERING

Operating System

Deadlock

Lecture - 02

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Reader-Writer Problem

Topic Dining Philosopher Problem

Topic Deadlock

Topics to be Covered

- Topic: Deadlock
- Topic: Deadlock Prevention
- Topic: Deadlock Avoidance

Topic : Deadlock

If two or more processes are waiting for such an event which is never going to occur

Topic : Resource Allocation Graph

Denotes which process is using which resource and is waiting for which resource.

Nodes/vertices :-

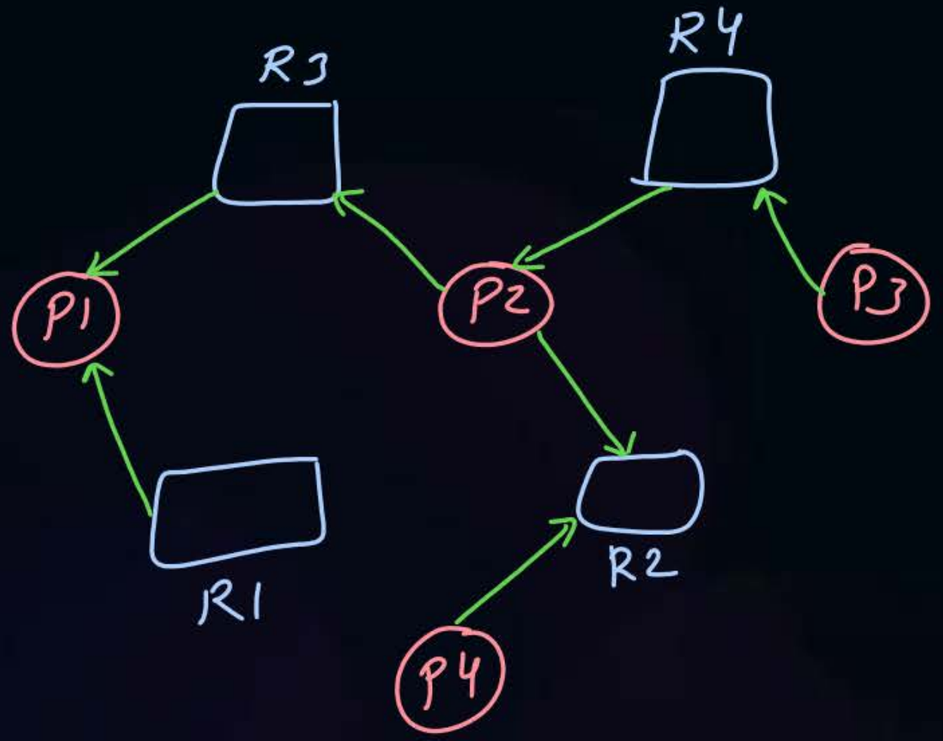
- Process — P1
- Resources — R1 ⇒  R1 (3 instances of resource R1)

Edges :-

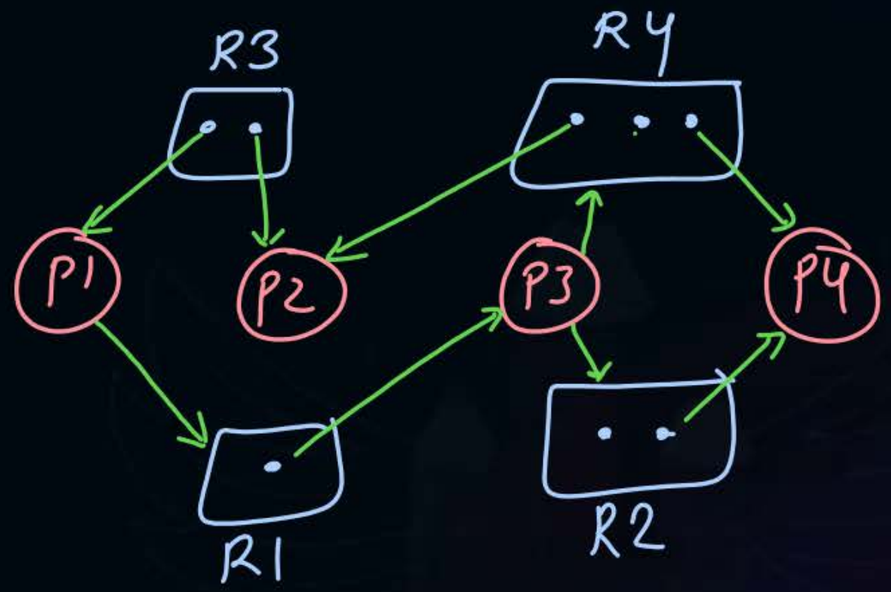
- Allocation :- from resource instance to process
- Request (wait) :- from process resource

Topic : Resource Allocation Graph

Ex:- when all resources have single instance.



when resources have multiple instances



Topic : Necessary Conditions for Deadlock

Deadlock can occur only when all following conditions are satisfied:

1. Mutual Exclusion
2. Hold & Wait
3. No-preemption
4. Circular Wait

Deadlock :- ex :-

	Hold	wait
P1	Harddisk	printer
P2	printer	keyboard
P3	keyboard	Harddisk

1. Mutual Exclusion:-
At a time one resource is used by only one pro.

2. Hold & wait:-
Each deadlocked process must hold atleast one resource and should wait for atleast one resource.

3. No preemption:-
A resource which is allocated to a process can not be preempted.

4. Circular wait:-
Each deadlocked process must wait for each other in circular manner.

Topic : Recovery From Deadlock

1. Make Sure that deadlock never occur
   - Prevent the system from deadlock or avoid deadlock
2. Allow deadlock, detect and recover
3. Pretend that there is no any deadlock $\Rightarrow$ in all modern system.

Deadlock prevention :-

Try to prevent atleast one of 4 necessary Conditions for deadlock, to occur.

1. Preventing Mutual Exclusion :-

- Make all processes independent
- Avail resources for each processes individually

→ not practical

2. Preventing Hold & wait :-

each process must hold all required resources; or else must wait for all required resources.

→ If some resources are acquired and other resources are not available; then process must release acquired resources and must wait for all resources.

Problems ⇒

A process may suffer from starvation if all required resources are not available.

3. Preventing No pre-emption :-

OS allows preemption of resources from processes.

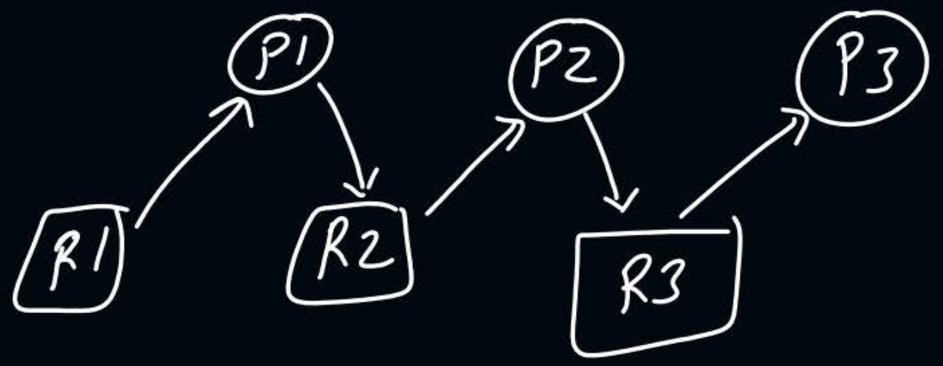
Problems :-

1. selectⁿ of victim processes and resources
2. The victim process may suffer from unstability or starvation
3. How many resources to be preempted.

4. Preventing circular wait :-

Each resource is given a unique number and each process is allowed to request resources in ascending order.

A process P can request for a resource R_j while holding a resource R_i only when $i < j$.



P3 can not request for R1 while holding R3.

P3 must release R3 and should try to acquire R1 first.

Topic : Deadlock Avoidance

In deadlock avoidance, the OS tries to keep system in safe state

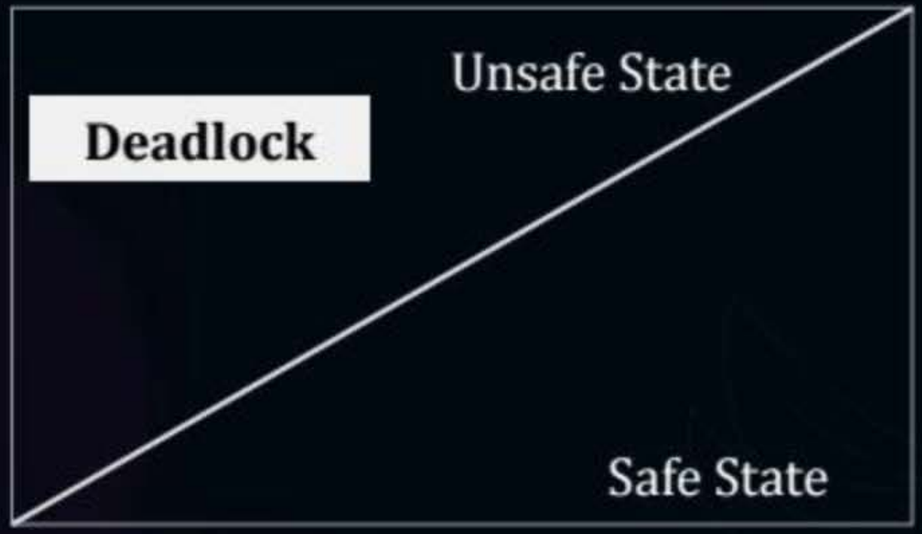
↓

deadlock will never occur

unsafe state :- possibility of deadlock

Topic : Deadlock Avoidance

In deadlock avoidance, the OS tries to keep system in safe state



Topic : Deadlock Avoidance

In deadlock avoidance, the request for any resource will be granted if the resulting state of the system doesn't cause deadlock in the system.

To check if system is in safe state or not

& if a request of a process must be granted or not

Banker's Algorithm

2 mins Summary

Topic Deadlock

Topic Deadlock Prevention

Topic Deadlock Avoidance

Happy Learning

THANK - YOU