

CS & IT ENGINEERING



Operating System

Deadlock

Lecture -1

By- Vishvadeep Gothi sir



Recap of Previous Lecture



Topic

Dining Philosopher Problem

Topic

Questions on Synchronization

Topic

Deadlock

Topics to be Covered



Topic

Deadlock

Topic

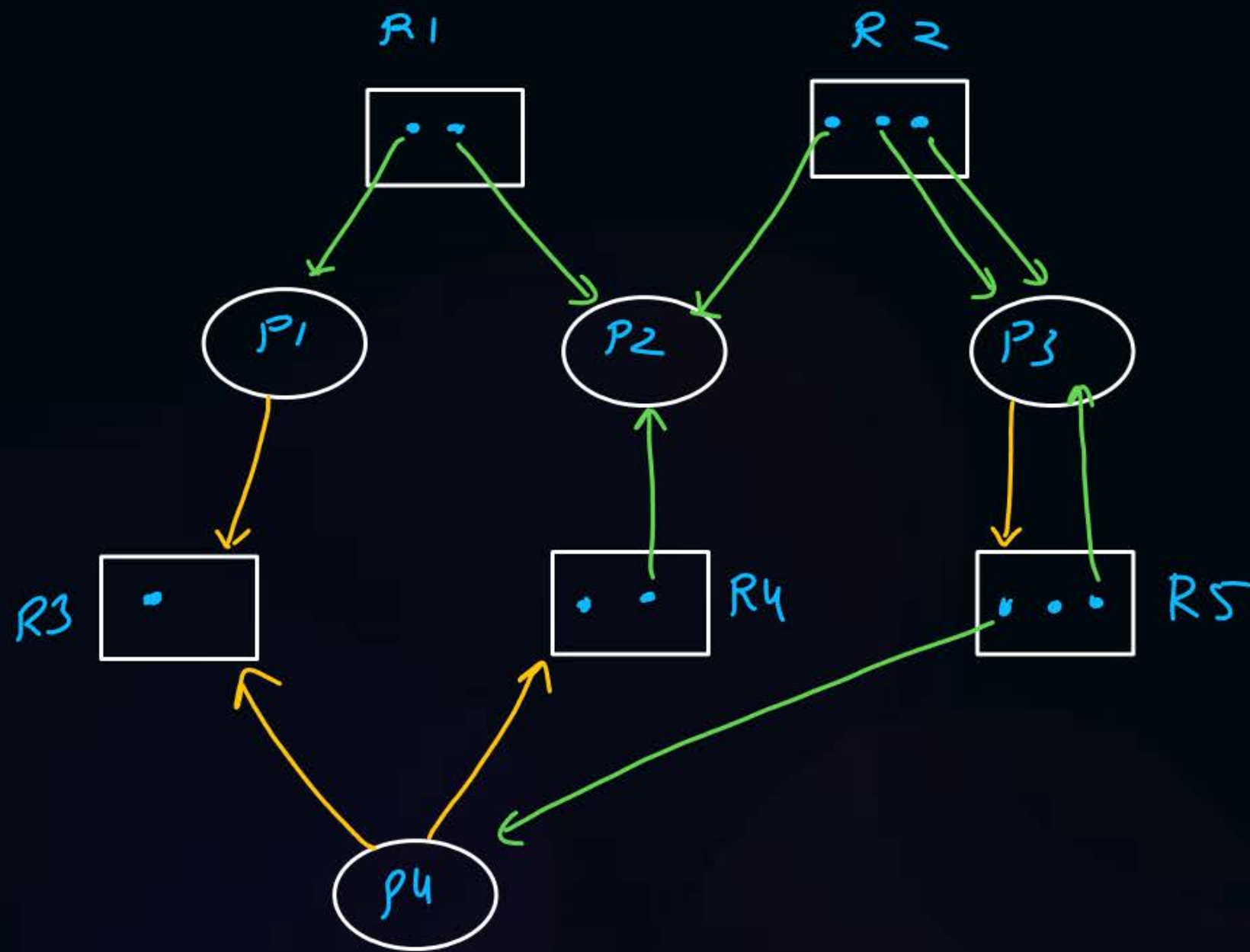
Deadlock Avoidance

Topic

Banker's Algorithm



Topic : Resource Allocation Graph



no. of requests made by P1 = 1?

Total no. of requests made 4?

How many processes do not have any pending request 1?

How many no. of instances are available 3?



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



Topic : Deadlock Prevention

Try to prevent one of 4 necessary conditions to satisfy.

1. Preventing mutual exclusion:-

1. Increase no. of resources so that resource sharing not needed
 2. make all processes independent
- not possible practically

2. Preventing hold and wait:-

make processes to either hold all resources or wait for all resources
⇒ starvation possible

3. Preventing No-preemption :-

Preempt resources from processes

- processes may go into unstable state
- selecting processes and resources for preemption
- processes may suffer from starvation

4. Preventing circular wait :-

if a process holds a resource R_i and wants a resource R_j .

Then it can request for R_j only when $i < j$.

if $i > j$ then process will release R_i . It will try to acquire R_j first then again R_i .

→ starvation possible



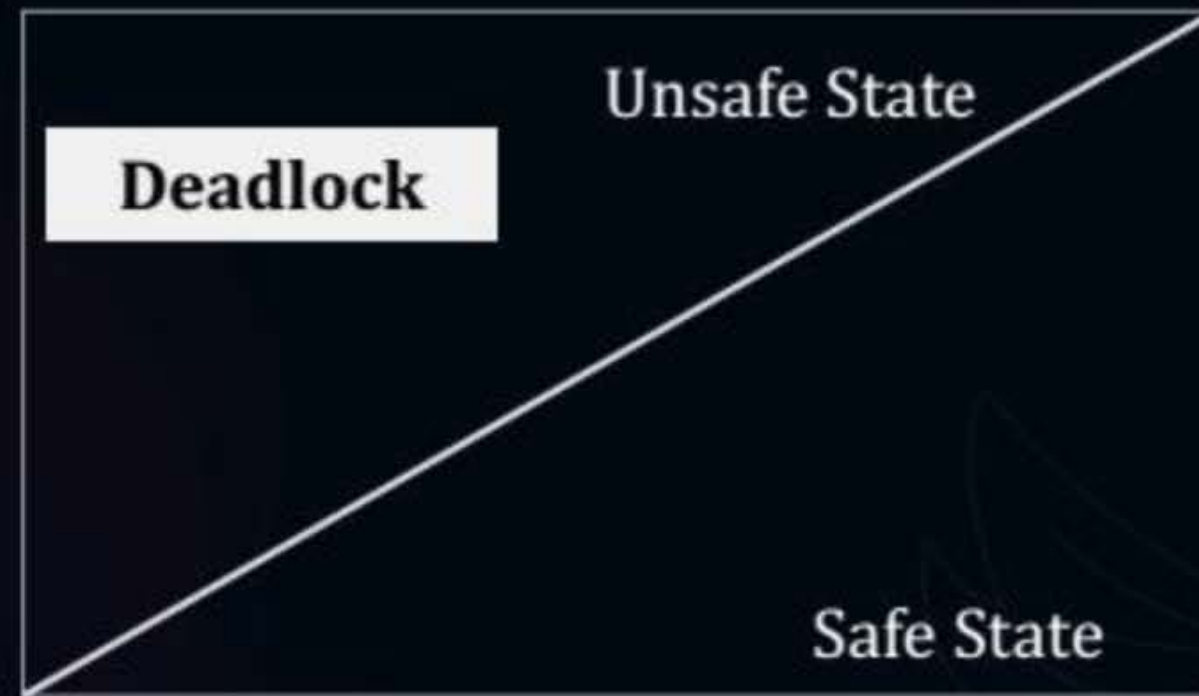
Topic : Deadlock Avoidance

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



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.





Topic : Banker's Algorithm

The banker's algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety





Topic : Banker's Algorithm

Process	Allocation	Max	Available
P1	1	3	1
P2	5	8	after P3 1+3=4
P3	3	4	after P1 5
P4	2	7	after P2 10

$$\text{Need} = \text{Max} - \text{Allocation}$$

2

3

1

5

for P3 $\text{need}_{P3} \leq \text{available}$

after P4 12

system \Rightarrow safe state

safe sequence

$\langle P3, P1, P2, P4 \rangle$

ans) In prev. question no. of safe sequences 4 ?

$\langle P_3, P_1, P_2, P_4 \rangle$

$\langle P_3, P_1, P_4, P_2 \rangle$

$\langle P_3, P_2, P_1, P_4 \rangle$

$\langle P_3, P_2, P_4, P_1 \rangle$

Ques)

	Allocation	Max	Available	Need
P1	2	8	1	6
P2	4	10	2	6
P3	1	2		1
P4	3	6		3

Unsafe

Total no. of resources system has, is = 11.

available = Total resources - sum of allocation

$$= 11 - 10$$

$$= 1$$



Topic : Banker's Algorithm

Process	Allocation	Max	Available	Need
	A B C	A B C	A B C	A B C
P ₀	0 1 0	7 5 3	3 3 2	7 4 3
P ₁	2 0 0	3 2 2	After P ₁ 5 3 2	1 2 2
P ₂	3 0 2	9 0 2	After P ₃ 7 4 3	6 0 0
P ₃	2 1 1	2 2 2		0 1 1
P ₄	0 0 2	4 3 3		4 3 1

safe state

At any point of time if $\text{Available} \geq \text{Need}$ of all remaining processes
↓
system is safe



Topic : Banker's Algorithm

1. Allocation: $\Rightarrow n \times m$ array
2. Max: $\Rightarrow n \times m$ array
3. Need: $\Rightarrow n \times m$ array
4. Available: $\Rightarrow 1-D$ array of size m

assume $\Rightarrow n$ processes
 m resources

(worst case)
Runtime Complexity of safety algo:-

1. need matrix calculation $\Rightarrow n \times m$

2. check which process has

$$\text{need}_{p_i} \leq \text{Available} \Rightarrow n \times m$$

if yes then add $\text{allocat}_{p_i}^n$
into available $\Rightarrow m$

repeat n times

$$\boxed{n^2 \times m} \in \text{R.T.C.}$$



Topic : Banker's Algorithm

1. Let Work and Finish be vectors of length 'm' and 'n' respectively.
Initialize: $Work = Available$
 $Finish[i] = false$; for $i=1, 2, 3, 4, \dots, n$
2. Find an i such that both (a) $Finish[i] = false$
(b) $Need_i \leq Work$
if no such i exists goto step (4)
3. $Work = Work + Allocation[i]$
 $Finish[i] = true$ goto step (2)
4. If $Finish[i] = true$ for all i
then the system is in a safe state

H.W.

#Q.

Process	Allocation A B C D	Max A B C D	Available A B C D
P1	0 0 1 2	0 0 1 2	1 5 2 0
P2	1 0 0 0	1 7 5 0	
P3	1 3 5 4	2 3 5 6	
P4	0 6 3 2	0 6 5 4	
P5	0 0 1 4	0 6 5 6	



Topic : Banker's Algorithm

Process	Allocation	Max	Available	Need
	A B C	A B C	A B C	A B C
P ₀	^{1 1 2} 0 1 0	7 5 3	3 3 2 2 3 0	^{6 4 1} 7 4 3
P ₁	2 0 0	3 2 2		1 2 2
P ₂	3 0 2	9 0 2		6 0 0
P ₃	2 1 1	2 2 2		0 1 1
P ₄	0 0 2	4 3 3		4 3 1

No any process
Can run.

⇓
unsafe
⇓

request of P₀ will be
rejected.

#Q. What will happen if process P0 requests one additional instance of resource type A and two instances of resource type C?

$$\text{Request}_{P_0} = \langle 1, 0, 2 \rangle$$

1. if $\text{Request}_i \leq \text{Need}_i$ then goto step 2
else invalid request and exit
2. if $\text{Request}_i \leq \text{Available}$ then goto step 3
else enough resources are not available and exit
3. $\text{Allocation}_i = \text{Allocation}_i + \text{request}_i$
 $\text{Need}_i = \text{Need}_i - \text{Request}_i$
 $\text{Available} = \text{Available} - \text{Request}_i$
4. run safety algo, if safe then request granted else rejected



Topic : Banker's Algorithm

Process	Allocation	Max	Available	Need
	A B C	A B C	A B C	A B C
P ₀	0 1 0	7 5 3	3 3 2	7 4 3
P ₁	2 0 0	3 2 2		1 2 2
P ₂	3 0 2	9 0 2		6 0 0
P ₃	2 1 1	2 2 2		0 1 1
P ₄	0 0 2	4 3 3		4 3 1

[NAT]

H.W.



#Q. What will happen if process P3 requests one additional instance of resource type B ?



2 mins Summary

Topic

Deadlock

Topic

Deadlock Avoidance

Topic

Banker's Algorithm



Happy Learning

THANK - YOU

