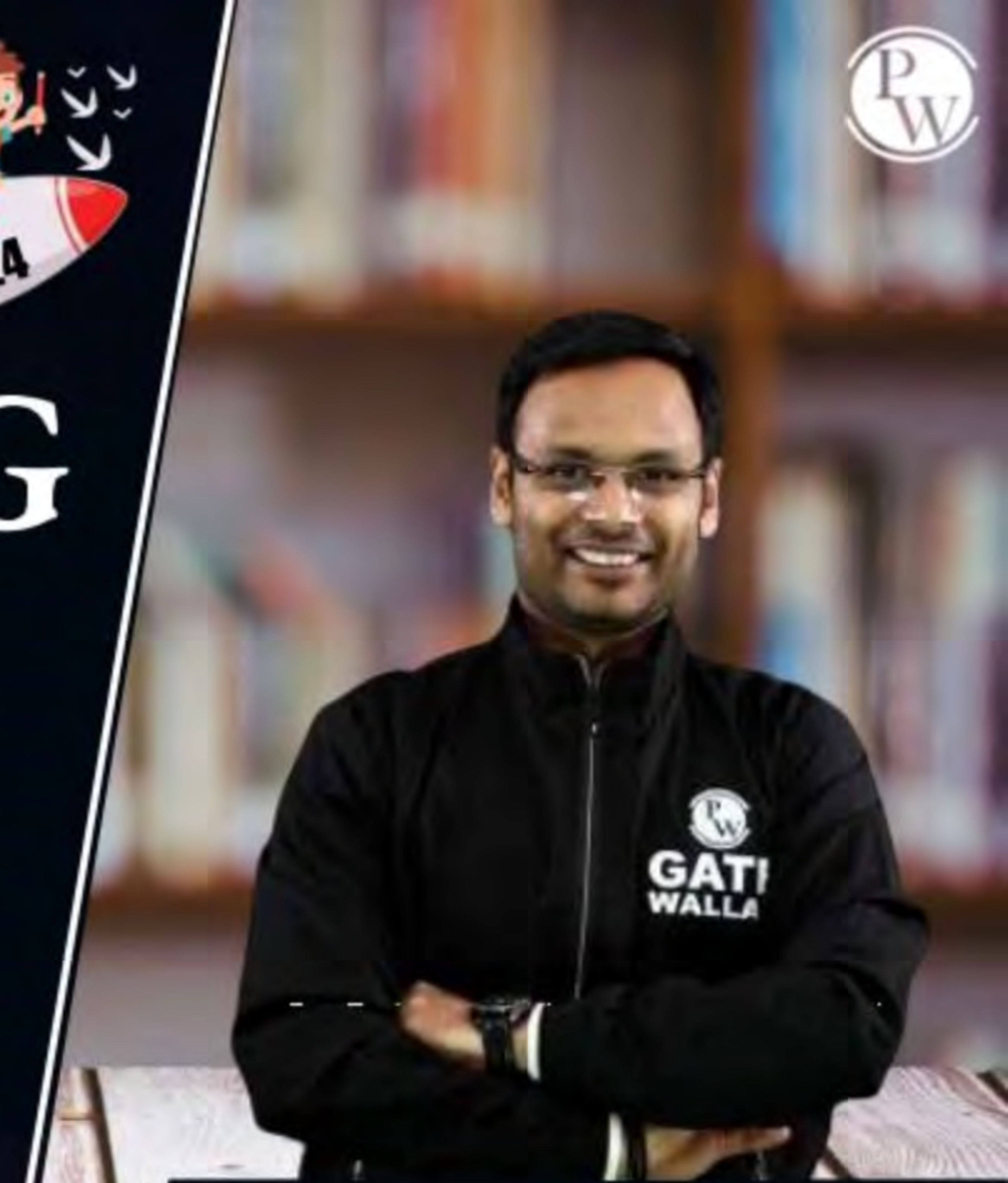
CS & IT ENGINEERING

Operating System



Deadlock

Lecture - 03

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Deadlock

Topic Deadlock Prevention

Topic Deadlock Avoidance

Topics to be Covered

Topic Deadlock Avoidance

Topic Banker's Safety Algorithm

Topic Banker's Resource Request Algorithm

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

Banker's algo :-

1. safety algo ⇒ it checks whether system is in safe state or not.

2. Resource Request Algo

Banker's Algo :-

Requirement :- Every process must announcement the max. no. of instances of each resource before the process starts execution.

It is practically not possible.

Hence banker's algo is not implemented practically.

Topic : Banker's Algorithm

already allocated to process

max no. of instances of resource needed to process for execution

Process	Allocation	Max	Available
P1	1	3	1
P2	5	8	
P3	3	4	
P4	2	7	

no. of available instances of resource in system

$$\text{available} = \text{Total no. of instances} - \sum_{i=}^{n} \text{Allocation}_i$$

Topic : Banker's Algorithm

no. of needed instances to completely execute the process

↓

Process	Allocation	Max	Available	Need = Max − Allocation
P1	1	3	~~1~~	2
P2	5	8	After P3 ⇒ 4	3
P3	3	4	After P1 ⇒ 5 After P2 ⇒ 10	1
P4	2	7	After P4 ⇒ 12	5

find one process P_i for which $Need_i \leq Available$

1. P_3 is such process.

After P_3 ⇒ $Available = Available + Allocation_3$

$= 1 + 3$

$= 4$

2. P1 has $need_1 \le available$

after P1 $\Rightarrow$ Available $= 4+1=5$

3. P2 has $need_2 \le available$

after P2 $\Rightarrow$ available $= 5+5 = 10$

4. P4 has $need_4 \le available$

after P4 $\Rightarrow$ available $= 10+2 = 12$

All process can finish hence

$\Downarrow$

system is in safe state

---

safe sequence

<P3, P1, P2, P4>

or

<P3, P2, P1, P4>

---

multiple safe sequences are possible.

Topic : Banker's Algorithm

Safe

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 P1 5 3 2	1 2 2 ✓
P_2	3 0 2	9 0 2	After P3 7 4 3	6 0 0
P_3	2 1 1	2 2 2	After P0 7 5 3	0 1 1 ✓
P4	0 0 2	4 3 3	After P2 10 5 5	4 3 1
			After P4 10 5 7	

available

After P1, P3 ⇒ 7 4 3

⇓

which is greater than or equal to need of all remaining processes P0, P2, P4.

Hence we can conclude here itself that system is in safe state.

---

safe sequne ⇒ P1, P3, (P0, P2, P4)!

⇒ P3, P1, (P0, P2, P4)!

Topic : Banker's Algorithm

1. Allocation:
2. Max:
3. Need:
4. Available:

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….n
2. Find an i such that both (a) Finish[i] = false
   (b) Needi <= 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

#Q. safe or not

Process	Allocation A B C D	Max A B C D	Available A B C D	Need A B C D
P1	0 0 1 2	0 0 1 2	~~1 5 2 0~~	0 0 0 0
P2	1 0 0 0	1 7 5 0	P1 1 5 3 2	0 7 5 0
P3	1 3 5 4	2 3 5 6	P3 2 8 8 6	1 0 0 2
P4	0 6 3 2	0 6 5 4		0 0 2 2
P5	0 0 1 4	0 6 5 6		0 6 4 2

safe state <P1, P3, P2, P4, P5>

Topic : Banker's Algorithm

Resource Request Algo

Process	Allocation	Max	Available	Need
	A B C	A B C	A B C	A B C
P_0	~~0 1 0~~ 1 1 2	7 5 3	~~3 3 2~~ 2 3 0	~~7 4 3~~ 6 4 1
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

Unsafe state because no any process has $Need_i \le available$

$\Downarrow$

hence request Rejected

[NAT]

(Rejected)

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

$Request_0 \langle 1, 0, 2 \rangle$

1. if $Request_i \leq Need_i$ then goto step 2
else invalid request

2. if $Request_i \leq Available$ then goto step 3
process will wait

3. $Available = Available - Request_i$
$Allocation_i = Allocation_i + Request_i$
$Need_i = Need_i - Request_i$

4. Run safety algo.
If safe then grant request.
Else request rejected

Topic : Banker's Algorithm

H.W.

Process	Allocation	Max	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

[NAT]

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

$$request_3 = \langle 0, 1, 0 \rangle$$

2 mins Summary

Topic Deadlock Avoidance

Topic Banker's Safety Algorithm

Topic Banker's Resource Request Algorithm

Happy Learning

THANK - YOU