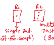
③ Deadlock Avoidance

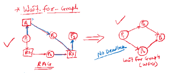
Basic fact – ① If a syst is in safe state ⇒ No Deadlock.
② If a syst is in unsafe state ⇒ possibility of Deadlock

Avoidance – Ensure that a system will never enter in a unsafe state



Single Inst. (Wait-for-Graph)
multiple Instance (Banker's Algo.)

* Wait-for-Graph



Example – Let 4 processes
Total = 12 instance

Process	max Need	Current allocation
P1	8	3
P2	9	4
P3	5	2
P4	3	1

Is system safe? If so, what is the safe seq?

⇒ Total no. of Resource Inst = 12 ✓
Total used ——— = 10 ✓
How many resources is left = 12 − 10 = 2 ✓

P1 ⇒ P1 need is 8, available = 2
∴ If 2 units given to P1, 3 + 2 = 5, i.e. 5 ≱ 8, Incomple ✗

P2 ⇒ 4 + 2 = 6, i.e. 6 ≱ 9, Incomplete ✗
P3 ⇒ 2 + 2 = 4, i.e. 4 ≱ 5, Incomplete ✗
P4 ⇒ 1 + 2 = 3, i.e. 3 ≥ 3 (True) – complete ✓
So, P4 will release allocated resources i.e 1
Total available = 2 + 1 = 3

P1 ⇒ 3 + 3 = 6, 6 ≱ 8, Incomplete ✗
P2 ⇒ 4 + 3 = 7, 7 ≱ 9, Incomplete ✗
P3 ⇒ 2 + 3 = 5, 5 ≥ 5 (True), Complete ✓
So, P3 will release 2, Total = 3 + 2 = 5
P1 ⇒ 3 + 5 = 8, i.e. 8 ≥ 8 (True), Complete ✓
So, P1 will release 3, Total = 5 + 3 = 8
P2 ⇒ 4 + 8 = 12, 12 ≥ 9 (True), complete ✓

∴ The system is safe and safe seq. is

< P4, P3, P1, P2 >

* Banker's Algorithm
(Avoidance OR Detection)

Example – Total, R1 = 10, R2 = 5, R3 = 7

Process	Allocation			max Need			Current Available			Remaining Need (max − allocation)		
	R1	R2	R3	R1	R2	R3	R1	R2	R3	R1	R2	R3
P0	0	1	0	7	5	3	~~3~~	~~3~~	~~2~~	7	4	3 → P0
P1	2	0	0	3	2	2	5	3	2	1	2	2 → P1
P2	3	0	2	9	0	2				6	0	0 → P2
P3	2	1	1	4	2	2				2	1	1 → P3
P4	0	0	2	5	3	3				5	3	1 → P4
	7	2	5									

Is syst in safe mode? If so, find safe seq?

Current Work = Total − Total Allocation
R1 = 10 − 7 = 3
R2 = 5 − 2 = 3
R3 = 7 − 5 = 2
(R1, R2, R3) = (3, 3, 2)

Banker's Algo
$\text{Need}_i \le \text{Work}$,
Work = work + Allocation

P0 ⇒ 7 4 3 ≤ 3 3 2, ✗
P1 ⇒ 1 2 2 ≤ 3 3 2, complete ✓
W = 3 3 2 + 2 0 0
= 5 3 2 ✓

P2 ⇒ 6 0 0 ≤ 5 3 2 ✗
P3 ⇒ 2 1 1 ≤ 5 3 2 (Complete) ✓
W = 5 3 2 + 2 1 1
= 7 4 3

P4 ⇒ 5 3 1 ≤ 7 4 3 – Complete ✓
W = 7 4 3 + 0 0 2
= 7 4 5

P0 ⇒ 7 4 3 ≤ 7 4 5 – Complete ✓
W = 7 4 5 + 0 1 0
= 7 5 5

P2 ⇒ 6 0 0 ≤ 7 5 5 – Complete ✓
W = 7 5 5 + 3 0 2
= 10 5 7

Thus, syst is in safe mode.
∴ Safe seq. < P1, P3, P4, P0, P2 >

④ Deadlock Detection & Recovery

– Allow syst in DL.
- * RAG ✓
- * WFG ✓
- * Banker's Algo ✓

① optimistic Approach ✓
(preemption of Resources & processes)

② pessimistic Approach
(process Termination)
– Abort all Deadlock processes ✓
– Abort one proc.

No Preem

P1 – 10% ✓
P2 – 90%
P3 – 90%
→ call Detection algo
⇓
overhead