D.E. → Python.
Website.
React | SEs

Q.]

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_1	0	0	1	2	0	0	1	2	1	5	2	0
P_2	1	0	0	0	1	7	5	0				
P_3	1	3	5	4	2	3	5	6				
P_4	0	6	3	2	0	6	5	2				
P_5	0	0	1	4	0	6	5	6				

1.) Find need matrix?
2.) Is the system in safe state?
3.) If 'YES' then find a safe sequence.
Apply Safety and Resourse-request algorithm.

33323

33323

Ans → (i) Need matrix : Need = Max - Allocation.

	Need			
	A	B	C	D
P_1	0	0	0	0
P_2	0	7	5	0
P_3	1	0	0	2
P_4	0	0	2	0
P_5	0	6	4	2

2) Checking if system is in safe state:

a) P_1 : Need <= Availability

(0,0,0,0) <= (1,5,2,0) ⇒ True.

New availability = Availability + Allocation.

= (1,5,2,0) + (0,0,1,2) = (1,5,3,2).

b) P_2 : (0,7,5,0) ~~(1,0,0,2)~~ <= (1,5,3,2) → ~~True~~ False.

Need ≤ availability ∴ P_2 is not executed.

c) P_3 : (1,0,0,2) <= (1,5,3,2) → True

Need ≤ availability → P_3 is executed.

New availability = (1,5,3,2) + (1,3,5,4)

= (2,8,8,6).

d) P_4 : (0,0,2,0) <= (2,8,8,6) → True.
Need <= Availability → P_4 will be executed.
New availability = (2,8,8,6) + (0,6,3,2) =
= (2, 14, 11, 8).

e) P_5 : (0,6,4,2) <= (2,14,11,8) → True.
Need ≤ Availability → P_5 will be executed.

After this, for P_2 : (0,7,5,0) <= (2,14,12,12) → True.
Need ≤ availability → P_2 will be executed now.

Sequence : P_1, P_3, P_4, P_5, P_2 → Hence system is in SAFE STATE.

3) Banker's Algo :
need [j][j] >= max[i][j] - allocc[i][j].
→ If request cannot be granted, its put on waiting list or denied.
→ Assumes that req. can be granted.
→ If new state is safe → grant the req.
else → deny or put on waiting list.

×