Prob. Consider the following snapshot of a system-

Answer the following questions using the Banker's algorithm-

(i) What is the content of the matrix need?

(ii) Is the system in a safe state?

(iii) If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?

Ans.

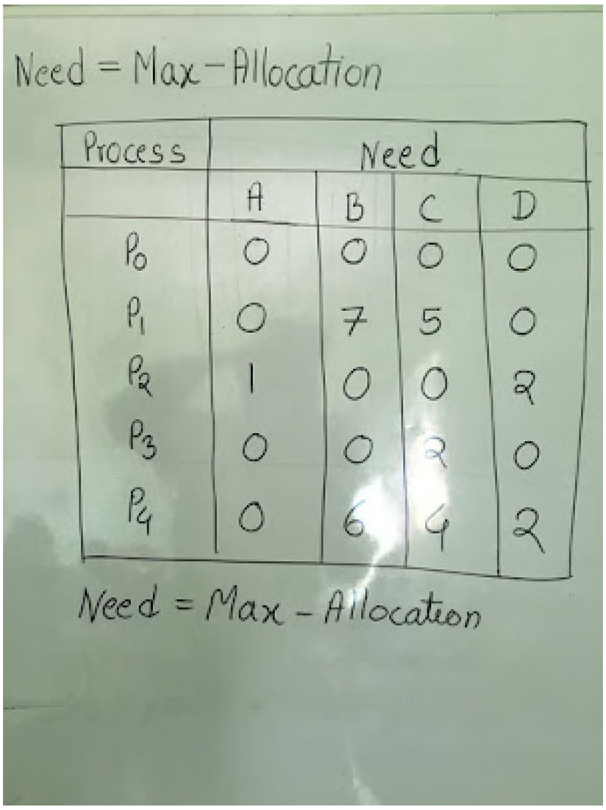
Banker's algorithm: Need Calculation

Steps to calculate need:

Step 1: in row of process P0, use formulaNeed=Max – Allocation

Step 2: Follow step 1 above for all other processes i.e. P1, P2, P3, P4, P5.

Result given below.

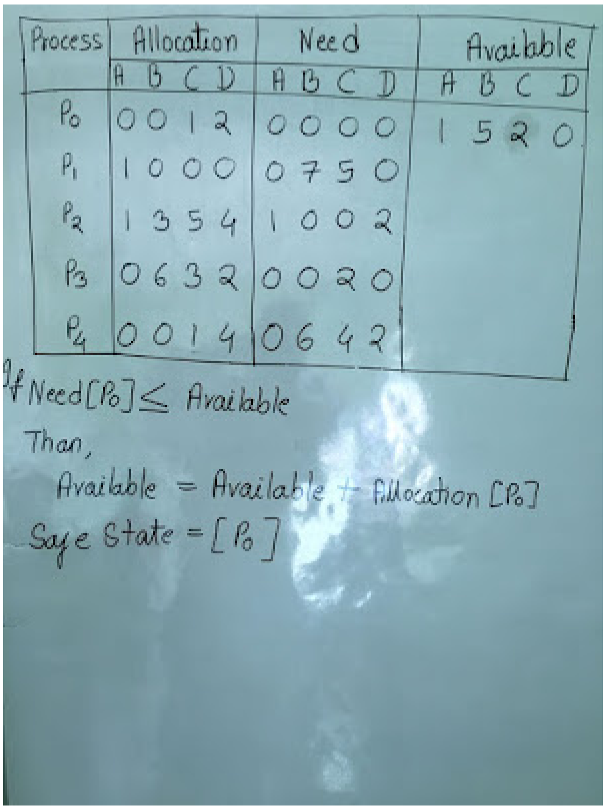


Need = Max-Allocation

Process	Need			
	A	B	C	D
P_0	0	0	0	0
P_1	0	7	5	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

Need = Max - Allocation

Bankers algorithm: Need calculationSteps to calculate Safe state:



Process	Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	0	0	1	5	2	0
P_1	1	0	0	0	0	7	5	0				
P_2	1	3	5	4	1	0	0	2				
P_3	0	6	3	2	0	0	2	0				
P_4	0	0	1	4	0	6	4	2				

If Need[P_0] $\leq$ Available

Than,

Available = Available + Allocation [P_0]

Safe State = [P_0]

Image 1

Step 1: Find the process which have Need lesser than Available.

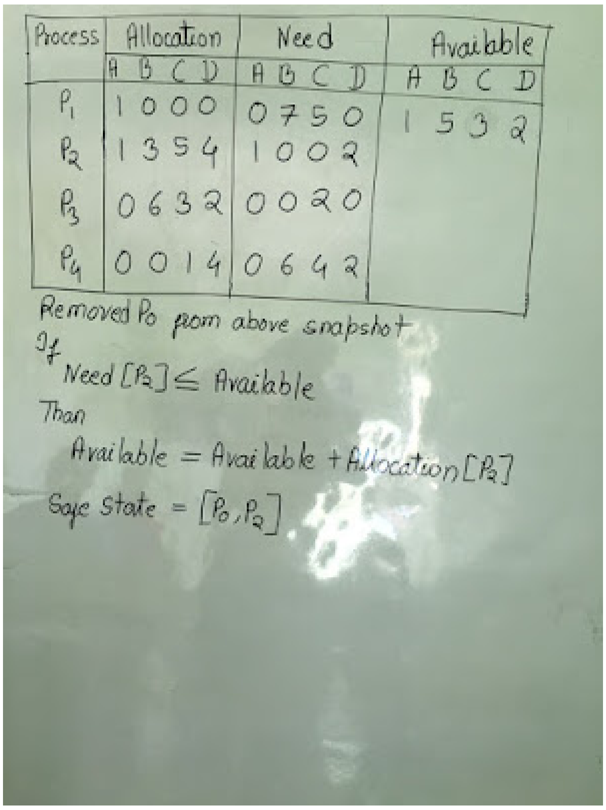
If need of process is lesser than available, add its allocation to the available and remove that process from the table.

Here as shown below,

Need [Process P0] is less than AvailableThan,New Available = Available + Allocation [Process P0] (Available value is updated in Image 2)

So ,

Safe state = [P0].



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

Removed P_0 from above snapshot

If

Need $[P_2] \leq$ Available

Than

Available = Available + Allocation $[P_2]$

Safe State = $[P_0, P_2]$

Image 2

Step 2: In STEP 1, We find the process P0 which have Need lesser than Available. We removed Process P0 from table, and updated the Available. (See Image 2).

Now we repeat Step 1 as explain above.

Here as shown below,

Need [Process P2] is less than AvailableThan,New Available = Available + Allocation [Process P2](Available value is updated in Image 3)So ,

Safe state = [P0, P2].

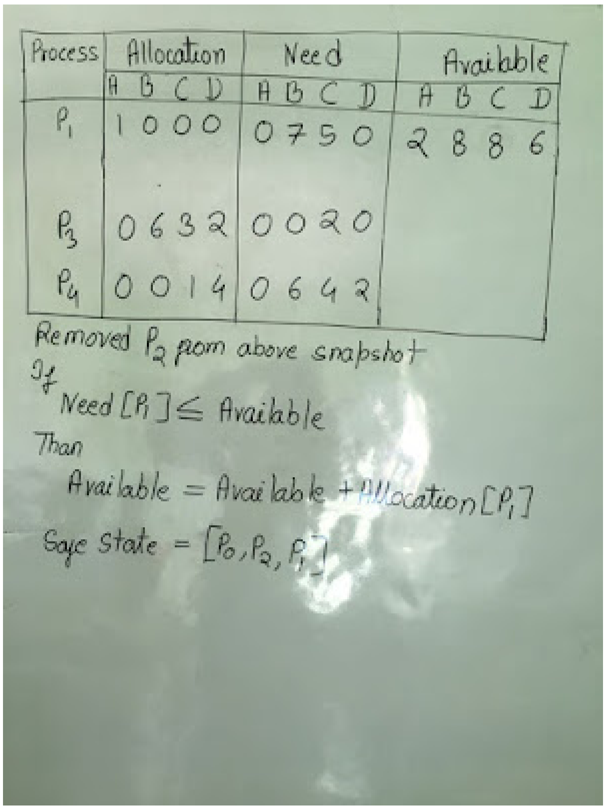


Image 3

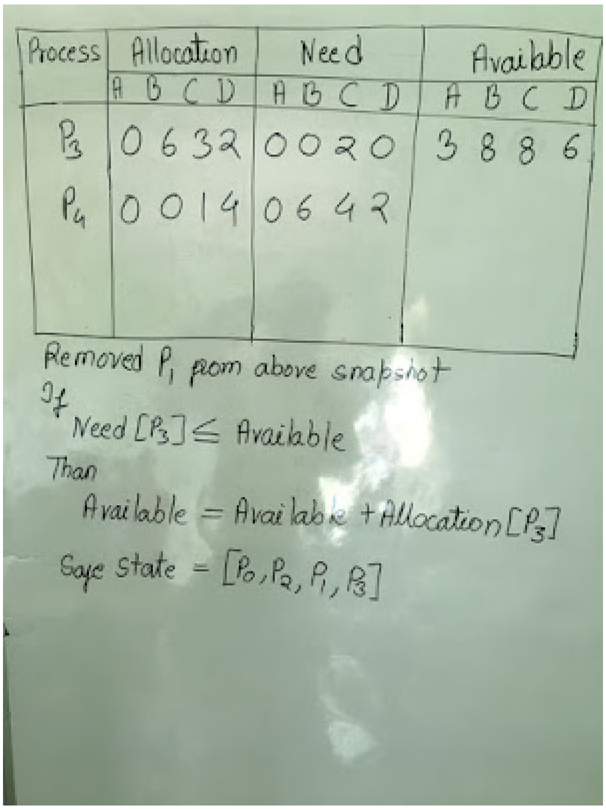
Step 3: In STEP 2, we find the process P2 which have Need lesser than Available. We removed Process P2 from table, and updated the Available.(See Image 3).

Now we repeat Step 1 as explain above.

Here as shown below,

Need [Process P1] is less than AvailableThan,New Available = Available + Allocation [Process P1](Available value is updated in Image 4)So ,

Safe state = [P0, P2, P1].



Process	Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_3	0	6	3	2	0	0	2	0	3	8	8	6
P_4	0	0	1	4	0	6	4	2				

Removed P_1 from above snapshot

If

Need $[P_3] \leq$ Available

Than

Available = Available + Allocation $[P_3]$

Safe State = $[P_0, P_2, P_1, P_3]$

Image 4

Step 4: In STEP 3, we find the process P1 which have Need lesser than Available. We removed Process P1 from table, and updated the Available.(See Image 4).

Now we repeat Step 1 as explain above.

Here as shown below,

Need [Process P3] is less than AvailableThan,New Available = Available + Allocation [Process P3](Available value is updated in Image 5)So ,

Safe state = [P0, P2, P1, P3].

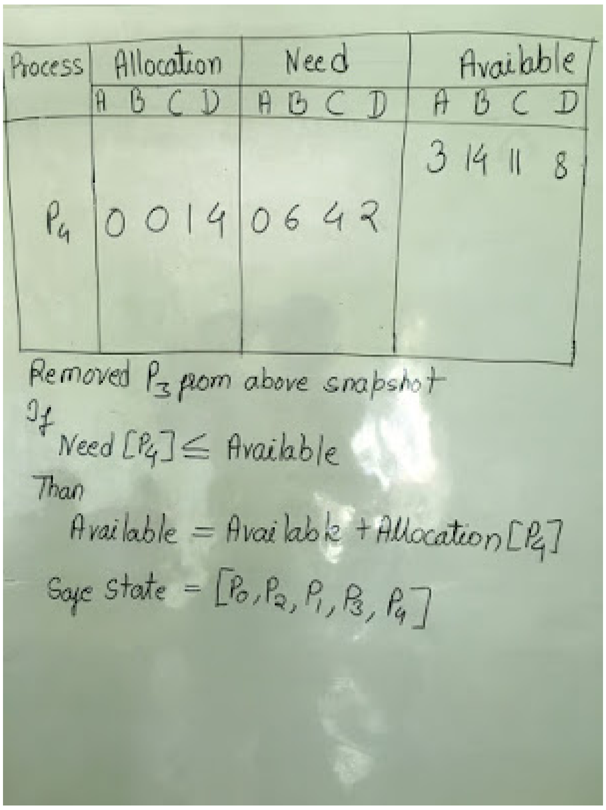


Image 5

Step 5: In STEP 4, we find the process P3 which have Need lesser than Available. We removed Process P3 from table, and updated the Available.(See Image 5).

Now we repeat Step 1 as explain above.

Here as shown below,
Need [Process P4] is less than AvailableSo ,
Safe state = [P0, P2, P1, P3, P4].
As all the processes comes under safe state, so system is in a safe state.