Banker’s Algorithm

DCS4103 Operating System

Banker's Algorithm

Allocation	• The number of resources of each type currently allocated to each process.
Max	• The maximum demand of each process in the system
Available (Work)	• The number of available resources of each type
Need	• The remaining resource need for each process

Example

Considering the following example of a system, check whether the system is safe or not; using Banker's algorithm. Determine the sequence if it is safe.

Process	Allocation			Max			Available (Work)		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
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			

Solution – Step 1 (Calculate Need)

$$R_i\,(Need) = R_i\,(Max) - R_i\,(Allocation)$$

Process	Allocation			Max			Available (Work)			Need		
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3
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	6
P_3	2	1	1	2	2	2				0	1	1
P_4	0	0	2	4	3	3				4	3	1

Solution – Step 2 (Perform Banker's Algorithm)

If $Need_i \leq Work_i$ then

Update Work = work + allocation

Else

$Process_i$ wait

$\geq$

Process	Allocation			Max			Available (Work)			Need			
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	
P_0	0	1	0	7	5	3	3	3	2	7	4	3	✗ ✓
P_1	2	0	0	3	2	2	5	3	2	1	2	2	✓
P_2	3	0	2	9	0	2	7	4	3	6	0	0	✗ ✓
P_3	2	1	1	2	2	2	7	4	5	0	1	1	✓
P_4	0	0	2	4	3	3	7	5	5	4	3	1	✓
							10	5	7				

$Safe\ sequence = P_i\,, P_i\,, \ldots$

Solution – Step 2 (Perform Banker's Algorithm)

If $Need_i \leq Work_i$ then

Update Work = work + allocation

Else

$Process_i$ wait

Process	Allocation			Need			Available (Work)			$Need_i \leq Work_i$
	R_1	R_2	R_3	R_1	R_2	R_3	R_1	R_2	R_3	
P_0	0	1	0	7	4	3	3	3	2	× ✓ 4
P_1	2	0	0	1	2	2	5	3	2	✓ 1
P_2	3	0	2	6	0	0	7	4	3	× ✓ 5
P_3	2	1	1	0	1	1	7	4	5	✓ 2
P_4	0	0	2	4	3	1	7	5	5	✓ 3
							10	5	7	

$$Safe\ sequence = P_1, P_3, P_4, P_0, P_2$$