Name: Nimra Nasir
Reg No: 19-CP-35

Operating System

Lab: 13 (Examples)

Examples:

Example: 1

Consider a system with 5 processes, P_0 through P_4 and 3 resources types A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time to following snap-shot of the system has been taken:

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			

What will be the content of the Need matrix?

Answer:

Resource type A = 10 instance

Resource type B = 5 instance

Resource type C = 7 instance

Formula:

$$Need[i,j] = Max[i,j] - Allocation[i,j]$$

Process	Max			Allocation			Need = Max [i,j] − Allocation [i,j]		
	A	B	C	A	B	C	A	B	C
P_0	7	5	3	0	1	0	7	4	3
P_1	3	2	2	2	0	0	1	2	2
P_2	9	0	2	3	0	2	6	0	0
P_3	2	2	2	2	1	1	0	1	1
P_4	4	3	3	0	0	2	4	3	1

Example: 2

Is the system in safe state? if yes, then what is the safe sequence ?

Ansuer:

Available = m = 3

Processes = n = 5

work = Available

Applying the safety algorithm on the given system:

Step 1 of Safety Algorithm:

work =	3	3	2

Finish =	False	False	False	False	False
	0	1	2	3	4

Step 2:

For i = 0

$Need_0$ = 7, 4, 3

Finish [0] is false because **$Need_0$ > work**
7, 4, 3   3, 2, 2

So **P_0** must wait.

Because we need : **$Need_0 \leq$ work**

Step 3:

For $i = 1$

$Need_1 = 1, 2, 2$

Finish [1] is false and $Need_1 < work$

$1, 2, 2 < 3, 3, 2$

so P_1 must be kept in safe sequence

so, $work = work + Allocation_1$

$3, 3, 2 \quad\quad 2, 0, 0$

$work = 3+2, 3+0, 2+0$

$work = \overset{A}{5}, \overset{B}{3}, \overset{C}{2}$

Finish =

Finish	True	Finish	Finish	Finish
0	1	2	3	4

Step 2:

For $i = 2$

$Need_2 = 6, 0, 0$

Finish [2] is false and $Need_2 > work$

$6, 0, 0 \quad\quad 5, 3, 2$

so P_2 must wait

Step 2:

For $i = 3$

$Need_3 = 0, 1, 1$

Finish [3] is false and $Need_3 < work$

$0, 1, 1 \quad\quad 5, 3, 2$

So P_3 must be kept in safe sequence

Step 3:

so, $work = work + Allocation_3$

$5, 3, 2 \quad\quad 2, 1, 1$

$work = \overset{A}{7}, \overset{B}{4}, \overset{C}{3}$

Finish =

False	True	false	True	false
0	1	2	3	4

Step 2:

For i= 4

$Need_4$ = 4, 3, 1

Finish [4] = false and **$Need_4$ < work**

4,3,1 7,4,3

So **P_4** must be Kept in safe sequence

Step 3:

So, **work = work + $Allocation_4$**

7,4,3 0,0,2

	A	B	C
work =	7	4	5

Finish =

false	True	false	True	True
0	1	2	3	4

Step 2:

For i = 0

$Need_0$ = 7, 4, 3

Finish [0] is false and **$Need_0$ < work**

7,4,3 7,4,5

So **P_0** must be kept in safe sequence

Step 3:

So,

work = work + $Allocation_0$

7,4,5 0,1,0

	A	B	C
work =	7	5	5

Finish =

True	True	false	True	True

Step 2:

For $i = 2$

$Need_2 = 6,0,0$

Finish [2] is false and $Need_2 < work$

6,0,0 7,5,5

So P_2 must be kept in safe sequence

Step 3:

$work = work + Allocation_2$

7,5,5 3,0,2

	A	B	C
work =	10	5	7

	0	1	2	3	4
Finish =	True	True	True	True	True

Step 4:

Finish [i] = true for $0 \le i \le n$

Hence the system is in safe state

The safe sequence is P_1, P_3, P_4, P_0, P_2

Example: 3

What will happen if process P_1 requests one additional instances of resource type A and two instances of resource type C?

	A	B	C
$Request_1$ =	1	0	2

Answer:

To decide whether the request is granted and we use Resource Request Algorithm

Process	$Request_1$			$Need_1$			Available		
	A	B	C	A	B	C	A	B	C
P_1	1	0	2	1	2	2	3	3	2

Step1 :

$\overset{1,0,2}{Request_1} < \overset{1,2,2}{Need_1}$ ✓

Step 2 :

$\overset{1,0,2}{Request_1} < \overset{3,3,2}{Available}$ ✓

Step 3 :

Available = Available − $Request_1$

$Allocation_1$ = $Allocation_1$ + $Request_1$

$Need_1$ = $Need_1$ − $Request_1$

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

Example: 4

we must determine whether this new system state is safe. To do so, we again execute safety algorithm on the above data structures.

Answer:

Available = m = 3

Processes = n = 5

work = Available

step 1 of safety Algo:

work =	2	3	0

Finish =

False	False	false	False	false
0	1	2	3	4

Step 2:

For i = 0

$Need_0$ = 7, 4, 3

Finish [0] is false and **$Need_0$ > work**

7, 4, 3   2, 3, 0

So **P_0** must wait

Step 2:

For i = 1

$Need_1$ = 0, 2, 0

Finish [1] is false and **$Need_1$ < work**

0, 2, 0   2, 3, 0

So **P_1** must be kept in safe sequence

Step 3:

work = work + $Allocation_1$

2, 3, 0   3, 0, 2

	A	B	C
work =	5	3	2

Finish =

False	True	False	False	False
0	1	2	3	4

Step 2:

For i = 2

$Need_2$ = 6, 0, 0

Finish [2] is false and **$Need_2$ > work**

6, 0, 0   5, 3, 2

So **P_2** must wait

Step 2:

For i = 3

$Need_3$ = 0, 1, 1

0, 1, 1   5, 3, 2

Finish [3] = false and **$Need_3$ < work**

So P_3 must be kept in safe sequence

Step 3:

work = work + $Allocation_3$

5,3,2 2,1,1

	A	B	C
work =	7	4	3

	0	1	2	3	4
Finish =	False	True	False	True	False

Step 2:

For i = 4

$Need_4$ = 4, 3, 1

Finish [4] is false and **$Need_4$ < work**

4,3,1 7,4,3

So P_4 must be kept in safe sequence

Step 3:

work = work + $Allocation_4$

7,4,3 0,0,2

	A	B	C
work =	7	4	5

	0	1	2	3	4
Finish =	false	True	False	True	True

Step 2:

For i = 0

$Need_0$ = 7, 4, 3

Finish [0] is false and **$Need_0$ < work**

7,4,3 7,4,5

So P_0 must be kept in safe sequence

Step 3:

work = work + $Allocation_0$

7,4,5 0,1,0

	A	B	C
work =	7	5	5

Finish =

True	True	false	True	True
0	1	2	3	4.

Step 2:

For i = 2

$Need_2 = 6, 0, 0$

Finish [2] is false and $Need_2 < work$

6,0,0   7,5,5

So P_2 must be kept in safe sequence

step 3:

work = work + $Allocation_2$

7,5,5   3,0,2

	A	B	C
work =	10	5	7

Finish =

True	True	True	True	True

Step 4:

Finish [i] = true for $0 \le i \le n$

Hence the system is in safe sequence

The safe sequence is P_1, P_3, P_4, P_0, P_2

Example: 5

Consider the following system snapshot using data structures in the Banker's Algorithm, with resources A, B, C, D and process P_0 to P_4

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

Using Banker's Algorithm, answer following questions.

i) How many resources of type A, B, C, D are there?

A = 9 ; B = 13 ; C = 10 ; D = 11 .

ii) What are the contents of the Need matrix?

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

iii) Is the system in a safe state? Why?

The system is in a safe state as the processes can be finished in sequences **P_0, P_2, P_4, P_1 and P_3.**

iv) If a request from process P_4 arrives for additional resources of (1,2,0,0) can the Banker's algorithm grant the request immediately? Show the other criteria.

Answer:

If a request from process P_4 arrives for additional resources of (1,2,0,0) and if this system request is granted, the new system will be:

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

After P_0 completes P_3 can be allocated 1020 from released 6012 and available 2011 (Total 8023) and < P_0, P_3, P_4, P_2, P_1 > is a safe sequence