Lecture-17

BDu

Date 05/03

Deadlock avoidance -

Safe state - A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock.

Safe sequence - A sequence of processes $P_1, P_2, \ldots P_n$ is a safe sequence for the current allocation state if for each P_i the resources that P_i can still request, can be satisfied by the currently available resources plus the resources held by all the P_j with $j < i$.

12 instance

(Max needs − alloc^n)

order: P_0, P_1, P_2, P_0

	Max needs	Allocation	Available (12−9)	Needs
P_0	10	5	3 − 2 = 1	5 ✗
P_1	4	2+2 = 4	1+4 = 5	2 ✓ needs < avail
P_2	5	2	5+2 = 7	3
			7+5 = 12	

equal to max need was

all 12 instance back

single resource, multiple instance

We check if $need_i \le available$

P_0 does not fulfil as $5 > 3$

P_1 fulfils

Allocation + (Avail − need)

↓

(Alloc + need)

= Alloc^n + avail

Order of allocation = $\langle P_1, P_2, P_0 \rangle$

	Max needs	Allocation	Available	Needs
P_0	10	5	2	5
P_1	4	2	2+2 = 4	2
P_2	5	2+1 = 3	4+5 = 9	2

$\langle P_1, P_2, P_0 \rangle$

Available [j] = k → k no. of instance of resource j is available. Resource

Max [i, j] = k → i^th process requires k no. of j^th instance.

Allocation [i, j] = k → k no of instance of resource j is already allocated to i^th process

Need [i, j] = k → k no. of instance of resource j is needed by i^th process

Banker's algorithm → Multiple resource, multiple instance

for all i

Finish [i] = False

Need_i = Max_i − Alloc_i

if (Need_i ≤ Available)

{

Available = Available + Alloc^n ;

Finish [i] = true

}

5 processes
3 resources

Resource	Instance
A =	10
B	5
C	7

(Max − Alloc^n)

	Allocation A B C	Max A B C	Need A B C	Available A B C
P_0	0 1 0	7 5 3	7 4 3	3 3 2 → Given
P_1	2 0 0	3 2 2	1 2 2	2 0 0 = 5 3 2
P_2	3 0 2	9 0 0	6 0 0	+ 2 1 1 = 7 4 3
P_3	2 1 1	2 2 2	0 1 1	0 0 2 = 7 4 5
P_4	0 0 2	4 3 3	4 3 1	+ 0 1 0 = 7 5 5
				+ 3 0 2 = 10 5 7

Safe sequence

< P_1 P_3 P_4 P_0 P_2 >

	Allocation A B C	Max A B C	Need A B C	Available A B C
4 P_0 ✓	0 1 0	7 5 3	7 4 3	2 3 0
1 P_1 ✓	3 0 2	3 2 2	0 2 0	3 0 2 = 5 3 2
5 P_2	3 0 2	9 0 2	6 0 0	+ 2 1 1 = 7 4 3
2 P_3 ✓	2 1 1	2 2 2	0 1 1	0 0 2 = 7 4 5
3 P_4 ✓	0 0 2	4 3 3	4 3 1	0 1 0 = 7 5 5
				3 0 2 = 10 5 7 ✓

Safe sequence : < P_1 P_3 P_4 P_0 P_2 >

Scanned by CamScanner