Deadlocks

The following table is a Snapshot of system consist of 5 processes P_0 through P_4 and 3 resource types: *A* (10 instances), *B* (5 instances), and *C* (7 instances) at time T_0:

	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	

Question 1 :

Can request for (3,3,0) by P_4 be granted?

Solution:

*Executing **Resource-Request** algorithm:*

1- Check that Request $\leq$ Need (that is, $(3,3,0) \leq (4,3,1) \Rightarrow$ true

2- Check that Request $\leq$ Available (that is, $(3,3,0) \leq (2,3,0) \Rightarrow$ False

We cannot grant the P_4 requesting, because the request is more than available resources. The process must be waiting until the requesting resources are available.

Question 2 :

Can request for (0,2,0) by P_0 be granted?

Solution:

Executing ***Resource-Request*** *algorithm:*

1- Check that Request $\leq$ $Need_0$ (that is, (0,2,0) $\leq$ (7,4,3) $\Rightarrow$ true

2- Check that Request $\leq$ Available (that is, (0,2,0) $\leq$ (2,3,0) $\Rightarrow$ true

3- *Available = Available – Request =>* (2,3,0) - (0,2,0) = (2,1,0)

Allocation = $Allocation_0$ + Request => (0,1,0)+(0,2,0) = (0,3,0)

Need = $Need_0$ – Request => (7,4,3) - (0,2,0) = (7,2,3)

Executing ***safety*** *algorithm:*

- Check P_0

Step 1: Work = (2,1,0)

Finish[0] = False

Step 2 : (a) *Finish [0] = false*

(b) $Need_0 \leq Work$

(7,2,3) $\leq$ (2,1,0)

so there is no i exist

Step 3 : skip it

P_0 no finish (no terminate)

- Check P_1

Step 1: Work = (2,1,0)

Finish[1] = False

Step 2 : (a) *Finish [1] = false*

(b) $Need_1 \leq Work$

(0,2,0) $\leq$ (2,1,0)

so there is no i exist

Step 3 : skip it

P_1 no finish (no terminate)

- <u>Check P_2</u>

Step 1: Work = (2,1,0)

Finish[2] = False

Step 2 : (a) *Finish* [*2*] = *false*

(b) $Need_2 \leq Work$

(6,0,0)≤ (2,1,0)

so there is no i exist

Step 3 : skip it

P_2 no finish (no terminate)

- <u>Check P_3</u>

Step 1: Work = (2,1,0)

Finish[3] = False

Step 2 : (a) *Finish* [*3*] = *false*

(b) $Need_3 \leq Work$

(0,1,1)≤ (2,1,0)

so there is no i exist

Step 3 : skip it

P_3 no finish (no terminate)

- <u>Check P_4</u>

Step 1: Work = (2,1,0)

Finish[4] = False

Step 2 : (a) *Finish* [*4*] = *false*

(b) $Need_4 \leq Work$

(4,3,1)≤ (2,1,0)

so there is no i exist

Step 3 : skip it

P_4 no finish (no terminate)

We cannot grant the P_0 requesting, because all the processes in the system cannot terminate is a safe state sequence, so the system cannot be in safe state which means the probability of deadlock .