Banker's Algorithm Problem Set

Solutions

Exercise 1

Assume that there are 5 processes, P_0 through P_4, and 4 types of resources. At T_0 we have the following system state:

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

1. Create the need matrix (max-allocation)

	A	B	C	D
P_0	0	1	0	0
P_1	0	4	2	1
P_2	1	0	0	1
P_3	0	0	2	0
P_4	0	6	4	2

2. Use the safety algorithm to test if the system is in a safe state.
   We will first define work and finish:

Work vector	Finish matrix	
1	P_0	False
5	P_1	False
2	P_2	False
0	P_3	False
	P_4	False

Check to see if $need_0$ (0,1,0,0) is less than or equal to work. It is, so let's set finish to true for that process and also update work by adding the allocated resources (0,1,1,0) for that process to work.

Work vector	Finish matrix	
1	P_0	True
6	P_1	False
3	P_2	False
0	P_3	False
	P_4	False

Now, let's check to see if $need_1$ (0,4,2,1) <= work. Remember that we have to check each element of the vector $need_1$ against the corresponding element in work. Because 1 is not less than 0 (the fourth element), we need to move on to P_2.

$Need_2$ (1,0,0,1) is not less than work, so must move on to P_3.

$Need_3$ (0,0,2,0) is less than work, so we can update work and finish.

Work vector	Finish matrix	
1	P_0	True
12	P_1	False
6	P_2	False
2	P_3	True
	P_4	False

Next, let's look at P_4. $Need_4$ (0,6,4,2) is less than work, so we can update work and finish as follows:

Work vector	Finish matrix	
1	P_0	True
12	P_1	False
7	P_2	False
6	P_3	True
	P_4	True

Now we can go back up to P_1. $Need_1$ (0,4,2,1) is less than work, so let's update work and finish.

Work vector	Finish matrix	
2	P_0	True
14	P_1	True
10	P_2	False
7	P_3	True
	P_4	True

Finally, let's look at P_2. $Need_2$ (1,0,0,1) is less than work, so we can then say that the system is in a safe state and the processes will be executed in the following order:

P_0, P_3, P_4, P_1, P_2

3. If the system is in a safe state, can the following requests be granted, why or why not? Please also run the safety algorithm on each request as necessary.
    a. P1 requests (2,1,1,0)

       We cannot grant this request, because we do not have enough available instances of resource A.

    b. P1 requests (0,2,1,0)

       There are enough available instances of the requested resources, so first let's pretend to accommodate the request and see what the system looks like:

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

Need matrix

	A	B	C	D
P_0	0	1	0	0
P_1	0	2	1	1
P_2	1	0	0	1
P_3	0	0	2	0
P_4	0	6	4	2

Now we need to run the safety algorithm:

Work vector	Finish matrix	
1	P_0	False
3	P_1	False
1	P_2	False
0	P_3	False
	P_4	False

Let's first look at P_0. $Need_0$ (0,1,0,0) is less than work, so we change the work vector and finish matrix as follows:

Work vector	Finish matrix	
1	P_0	True
4	P_1	False
2	P_2	False
0	P_3	False
	P_4	False

$Need_1$ (0,2,1,1) is not less than work, so we need to move on to P_2.

$Need_2$ (1,0,0,1) is not less than work, so we need to move on to P_3.

$Need_3$ (0,0,2,0) is less than or equal to work. Let's update work and finish:

Work vector	Finish matrix	
1	P_0	True
10	P_1	False
5	P_2	False
2	P_3	True
	P_4	False

Let's take a look at $Need_4$ (0,6,4,2). This is less than work, so we can update work and finish:

Work vector	Finish matrix	
1	P_0	True
10	P_1	False
6	P_2	False
6	P_3	True
	P_4	True

We can now go back to P_1. $Need_1$ (0,2,1,1) is less than work, so work and finish can be updated:

Work vector	Finish matrix	
1	P_0	True
14	P_1	True
10	P_2	False
7	P_3	True
	P_4	True

Finally, $Need_2$ (1,0,0,1) is less than work, so we can also accommodate this. Thus, the system is in a safe state when the processes are run in the following order: P_0,P_3,P_4,P_1,P_2. We therefore can grant the resource request.

Exercise 2

Assume that there are three resources, A, B, and C. There are 4 processes P_0 to P_3. At T_0 we have the following snapshot of the system:

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P_0	1	0	1	2	1	1	2	1	1
P_1	2	1	2	5	4	4			
P_2	3	0	0	3	1	1			
P_3	1	0	1	1	1	1			

1. Create the need matrix.

	Need		
	A	B	C
P_0	1	1	0
P_1	3	3	2
P_2	0	1	1
P_3	0	1	0

2. Is the system in a safe state? Why or why not?
   In order to check this, we should run the safety algorithm. Let's create the work vector and finish matrix:

Work vector	Finish matrix	
2	P_0	False
1	P_1	False
1	P_2	False
	P_3	False

$Need_0$ (1,1,0) is less than work, so let's go ahead and update work and finish:

Work vector	Finish matrix	
3	P_0	True
1	P_1	False
2	P_2	False
	P_3	False

$Need_1$ (3,3,2) is not less than work, so we have to move on to P_2.

$Need_2$ (0,1,1) is less than work, let's update work and finish:

Work vector	Finish matrix	
6	P_0	True
1	P_1	False
2	P_2	True
	P_3	False

$Need_3$ (0,1,0) is less than work, we can update work and finish:

Work vector	Finish matrix	
7	P_0	True
1	P_1	False
3	P_2	True
	P_3	True

We now need to go back to P_1. $Need_1$ (3,3,2) is not less than work, so we cannot continue. Thus, the system is not in a safe state.