CS307 Operating System Exercises Spring 2020

Exercise 3: Deadlocks

1. A computer system has 8 printers that are shared by K process. Each of the processes can take no more than 3 printers.

a. What is the minimum value of K that may cause the system deadlock? Why?

b. Is there a minimum value of K that must cause the system deadlock? Why?

Answer:

a. K_{min}= 4. When 4 process all hold 2 printers and wait for the other to release the rest one printer, this case will cause deadlock.

b. No. Because no matter what process number the system has, if the processes don′t request and hold the printers at the same time, it can′t cause the system deadlock. For example, if the processes access the printers in a sequence and we only allow at most two process to hold and use the printers, then we will never meet the deadlock.

2. Consider the following snapshot of a system:

	Allocation				Max				Available			
	R1	R2	R3	R4	R1	R2	R3	R4	R1	R2	R3	R4
P0	0	0	1	2	0	0	1	2	2	1	0	0
P1	2	0	0	0	2	7	5	0				
P2	0	0	3	4	6	6	5	6				
P3	2	3	5	4	4	3	5	6				
P4	0	3	3	2	0	6	5	2				

Answer the following questions using the banker´s algorithm:

a. What is the content of the matrix Need?

b. Is the system in a safe state? Why or why not?

c. If a request from process P2 arrives for (0,2,0,0), can the request be granted immediately?

Briefly Explain.

Answer:

a. Need = Max – Allocation. So we can get the matrix of Need as below:

	Need			
	R1	R2	R3	R4
P0	0	0	0	0
P1	0	7	5	0
P2	6	6	2	2
P3	2	0	0	2
P4	0	3	2	0

b. The system is in a safe state. Because there exist a safe sequence: P0–>P3–>P4–>P1–>P2.

c. No. Because the system’s available resources is smaller than the request resources of P2.

3. Consider a system with four processes P1, P2, P3, and P4, and two kinds of resources, R1, and R2, respectively. Each kind of resource has two instances. Furthermore:

P1 is allocated with an instance of R2, and requests an instance of R1.

P2 is allocated with an instance of R1, but doesn’t need any more resource.

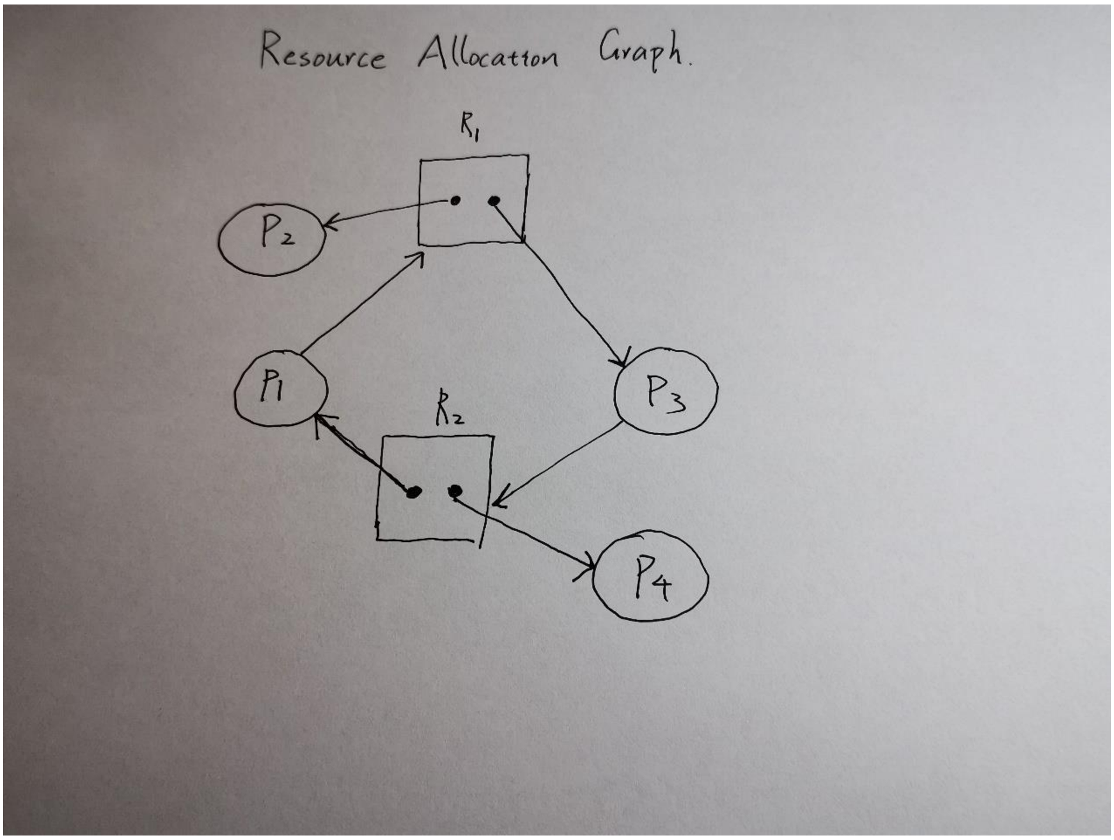
P3 is allocated with an instance of R1, and requests an instance of R2.

P4 is allocated with an instance of R2, but doesn’t need any more resource

a. Draw the resource allocation graph.
b. Is there a cycle in the graph? If yes name it.
c. Is the system in deadlock? If yes, explain why. If not, give a possible sequence of executions after which every process completes.

Answer:

a.



b. Yes. Cycle: P1 –> R1 –> P3 –> R2 –> P1.

c. No. A possible sequence: P4 –> P3 –> P2 –> P1

4. A system has four processes and five allocable resources. The current allocation and maximum needs are as follows:

	Allocated	Maximum	Available
Process A	1 0 2 1 1	1 1 2 1 3	0 0 1 X 2
Process B	2 0 1 1 0	2 2 2 1 0	
Process C	1 1 0 1 0	2 1 3 1 0	
Process D	1 1 1 1 0	1 1 2 2 1	

What is the smallest value of X for which this is a safe state?

First, we calculate the Need matrix as below:

Need

	R1	R2	R3	R4	R5
Process A	0	1	0	0	2
Process B	0	2	1	0	0
Process C	1	0	3	0	0
Process D	0	0	1	1	1.

Smallest value of X: 1.

Safe sequence: Process D –> Process A –> Process C –> Process B