

## Exercise for Deadlocks

3180103570 卢佳盈

**7.1 Consider the traffic deadlock depicted in Figure 7.9 , p269 of the textbook.**

**a. Show that the four necessary conditions for deadlock indeed hold in this example.**

1. mutual exclusion: only one car can occupy a space in the roadway
2. hold-and-wait: a car need to hold onto its place in the roadway while it waits to advance in the roadway
3. no preemption: a car cannot be removed from its position in the roadway
4. circular wait: each car is waiting for a subsequent car to advance

**b. State a simple rule for avoiding deadlocks in this system.**

a car may not advance

**7.6 Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock-free.**

When we allocate four resources to three processes, at least one process can be allocated to two resources, then the process can proceed smoothly. After the process runs, it will release the original two resources. At this time, four resources are allocated to two processes in the system, and there will be no deadlock.

**7.11 Consider the following snapshot of a system:**

	<i>Allocation</i>	<i>Max</i>	<i>Available</i>
	<i>A B C D</i>	<i>A B C D</i>	<i>A B C D</i>
<i>P<sub>0</sub></i>	0 0 1 2	0 0 1 2	1 5 2 0
<i>P<sub>1</sub></i>	1 0 0 0	1 7 5 0	
<i>P<sub>2</sub></i>	1 3 5 4	2 3 5 6	
<i>P<sub>3</sub></i>	0 6 3 2	0 6 5 2	
<i>P<sub>4</sub></i>	0 0 1 4	0 6 5 6	

**Answer the following questions using the banker's algorithm:**

**a. What is the content of the matrix Need?**

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

		<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
	$P_0$	0	0	0	0
$Need =$	$P_1$	0	7	5	0
	$P_2$	1	0	0	2
	$P_3$	0	0	2	0
	$P_4$	0	6	4	2

**b. Is the system in a safe state?**

Yes, it is in a safe state.

P0: Its matrix Need is [0 0 0 0], so the resource can be released. Then the Available Matrix is [1 5 3 2]

P1: Its matrix Need is [0 7 5 0] > matrix Available, so ignore it.

P2: Its matrix Need is [1 0 0 2] < matrix Available, so the resource can be released. Then the Available Matrix is [2 8 7 6].

P3: Its matrix Need is [0 0 2 0] < matrix Available, so the resource can be released. Then the Available Matrix is [2 14 11 8].

P4: Its matrix Need is [0 6 4 2] < matrix Available, so the resource can be released. Then the Available Matrix is [2 14 12 12].

P1: Now its Need < Available, it can be allocated the resource.

Finally, all processors are finished, so it is in a safe state.

**c. If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?**

Yes.

If  $request[i] < need[i]$  and  $request[i] < available$ , we can renew the matrix as following

$$\begin{aligned} available - &= request[i] \\ need[i] - &= request[i] \\ allocation[i] + &= request[i] \end{aligned}$$

Then do the judge of safety again like 7.11.b. If it is still in safe state, the request can be granted immediately.

Therefore, the request from process P1 arrives for (0,4,2,0) can be granted immediately.