第 8 章作业

8.3

8.3 Consider the following snapshot of a system:

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
T_0	0 0 1 2	0 0 1 2	1 5 2 0
T_1	1 0 0 0	1 7 5 0	
T_2	1 3 5 4	2 3 5 6	
T_3	0 6 3 2	0 6 5 2	
T_4	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*?

b. Is the system in a safe state?

c. If a request from thread T_1 arrives for (0,4,2,0), can the request be granted immediately?

答:

a.

	Need			
	A	B	C	D
T0	0	0	0	0
T1	0	7	5	0
T2	1	0	0	2
T3	0	0	2	0
T4	0	6	4	2

b. 该系统处于安全状态，<T0, T2, T1, T3, T4>为一个安全序列。

c. 新状态为

	Allocating				Max				Available				Need			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
T0	0	0	1	2	0	0	1	2	1	1	0	0	0	0	0	0
T1	1	4	2	0	1	7	5	0					0	3	3	0
T2	1	3	5	4	2	3	5	6					1	0	0	2
T3	0	6	3	2	0	6	5	2					0	0	2	0
T4	0	0	1	4	0	6	5	6					0	6	4	2

该系统处于安全状态，<T0, T2, T1, T3, T4>为一个安全序列，所以这条请求能被立即批准。

8.9

8.9 Consider the following snapshot of a system:

	Allocation	*Max*
	A B C D	*A B C D*
T_0	3 0 1 4	5 1 1 7
T_1	2 2 1 0	3 2 1 1
T_2	3 1 2 1	3 3 2 1
T_3	0 5 1 0	4 6 1 2
T_4	4 2 1 2	6 3 2 5

Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe.

a. ***Available*** $= (0, 3, 0, 1)$

b. ***Available*** $= (1, 0, 0, 2)$

答:

	Need			
	A	B	C	D
T0	2	1	0	3
T1	1	0	0	1
T2	0	2	0	0
T3	4	1	0	2
T4	2	1	1	3

a.

该系统处于非安全状态。

T2、T1、T3 按顺序完成后，Work = (5, 11, 4, 2)，Work ≤ Need0 不成立，Work ≤ Need4 也不成立，这两个进程会死锁，无法完成。

b.

该系统处于安全状态，<T1, T2, T0, T3, T4>为一个安全序列。

8.18

8.18 Which of the six resource-allocation graphs shown in Figure 8.12 illustrate deadlock? For those situations that are deadlocked, provide the cycle of threads and resources. Where there is not a deadlock situation, illustrate the order in which the threads may complete execution.

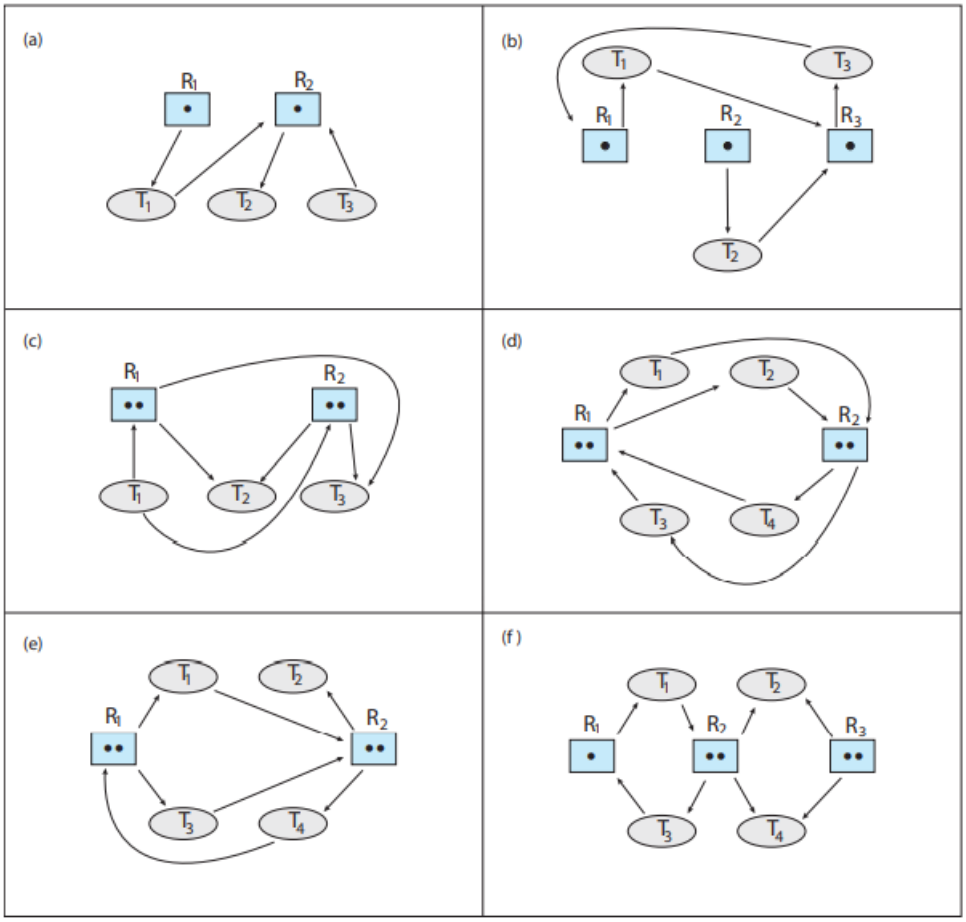

Figure 8.12 Resource-allocation graphs for Exercise 8.18.

答:

(a) 无死锁。<T2, T1, T3>

(b) 死锁。T1 -> R3 -> T3 -> R1 -> T1

(c) 无死锁。<T2, T1, T3>

(d) 死锁。T1&T2 -> R2 -> T3&T4 -> R1 -> T1&T2

(e) 无死锁。<T2, T1, T3, T4>

(f) 该图有误，R2的两个实例不可能被 T2、T3、T4 三个进程占有。

8.27

8.27 Consider the following snapshot of a system:

	Allocation	*Max*
	A B C D	*A B C D*
T_0	1 2 0 2	4 3 1 6
T_1	0 1 1 2	2 4 2 4
T_2	1 2 4 0	3 6 5 1
T_3	1 2 0 1	2 6 2 3
T_4	1 0 0 1	3 1 1 2

Using the banker's algorithm, determine whether or not each of the following states is unsafe. If the state is safe, illustrate the order in which the threads may complete. Otherwise, illustrate why the state is unsafe.

a. $\boldsymbol{Available} = (2, 2, 2, 3)$

b. $\boldsymbol{Available} = (4, 4, 1, 1)$

c. $\boldsymbol{Available} = (3, 0, 1, 4)$

d. $\boldsymbol{Available} = (1, 5, 2, 2)$

答:

	Need			
	A	B	C	D
T0	3	1	1	4
T1	2	3	1	2
T2	2	4	1	1
T3	1	4	2	2
T4	2	1	1	1

a. 该系统处于安全状态，<T4, T0, T1, T2, T3>为一个安全序列。

b. 该系统处于安全状态，<T2, T4, T1, T0, T3>为一个安全序列。

c. 该系统处于非安全状态，空闲的 B 为 0，而每个进程都还需要B，五个进程都会死锁，无法完成。

d. 该系统处于安全状态，<T3, T1, T2, T0, T4>为一个安全序列。

8.28

8.28 Consider the following snapshot of a system:

	Allocation	*Max*	*Available*
	A B C D	*A B C D*	*A B C D*
T_0	3 1 4 1	6 4 7 3	2 2 2 4
T_1	2 1 0 2	4 2 3 2	
T_2	2 4 1 3	2 5 3 3	
T_3	4 1 1 0	6 3 3 2	
T_4	2 2 2 1	5 6 7 5	

Answer the following questions using the banker's algorithm:

a. Illustrate that the system is in a safe state by demonstrating an order in which the threads may complete.

b. If a request from thread T_4 arrives for $(2, 2, 2, 4)$, can the request be granted immediately?

c. If a request from thread T_2 arrives for $(0, 1, 1, 0)$, can the request be granted immediately?

d. If a request from thread T_3 arrives for $(2, 2, 1, 2)$, can the request be granted immediately?

答:

a.

```
    Need
    -------
    A B C D
T0  3 3 3 2
T1  2 1 3 0
T2  0 1 2 0
T3  2 2 2 2
T4  3 4 5 4
```

<T2, T0, T1, T3, T4>为一个安全序列。

b. 新状态为

```
    Allocating  Max         Available   Need
    ----------  -------     ---------   -------
    A B C D     A B C D     A B C D     A B C D
T0  3 1 4 1     6 4 7 3     0 0 0 0     3 3 3 2
T1  2 1 0 2     4 2 3 2                 2 1 3 0
T2  2 4 1 3     2 5 3 3                 0 1 2 0
T3  4 1 1 0     6 3 3 2                 2 2 2 2
T4  4 4 4 5     5 6 7 5                 1 2 3 0
```

该系统处于非安全状态，已经没有任何空闲的资源，而每个进程都还需要资源，所以这条请求不能被立即批准。

c. 新状态为

```
    Allocating  Max         Available   Need
    ----------  -------     ---------   -------
    A B C D     A B C D     A B C D     A B C D
T0  3 1 4 1     6 4 7 3     2 1 1 4     3 3 3 2
T1  2 1 0 2     4 2 3 2                 2 1 3 0
T2  2 5 2 3     2 5 3 3                 0 0 1 0
T3  4 1 1 0     6 3 3 2                 2 2 2 2
T4  2 2 2 1     5 6 7 5                 3 4 5 4
```

该系统处于 该系统处于安全状态，<T2, T0, T1, T3, T4>为一个安全序列，所以这条请求能被立即批准。

d. 新状态为

```
    Allocating  Max         Available   Need
    ----------  -------     ---------   -------
    A B C D     A B C D     A B C D     A B C D
T0  3 1 4 1     6 4 7 3     0 0 1 2     3 3 3 2
T1  2 1 0 2     4 2 3 2                 2 1 3 0
T2  2 4 1 3     2 5 3 3                 0 1 2 0
T3  6 3 2 2     6 3 3 2                 0 0 1 0
T4  2 2 2 1     5 6 7 5                 3 4 5 4
```

该系统处于 该系统处于安全状态，<T3, T0, T1, T2, T4>为一个安全序列，所以这条请求能被立即批准。