操作系统(D)

第 8 次作业

Log Creative

2021 年 6 月 15 日

8.3 Consider the following snapshot of a system:

	Allocation	*Max*	*Available*
	ABCD	*ABCD*	*ABCD*
T_0	0012	0012	1520
T_1	1000	1750	
T_2	1354	2356	
T_3	0632	0652	
T_4	0014	0656	

Answer the following questions using the banker's algorithm:

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

解.

	Need
	ABCD
T_0	0000
T_1	0750
T_2	1002
T_3	0020
T_4	0642

b. Is the system in a safe state?

解.

Sequence	*Available*
	A, B, C, D
T_0	1,5,3,2
T_2	2,8,8,6
T_1	3,8,8,6
T_3	3,14,11,8
T_4	3,14,12,12

找到安全序列 $\langle T_0, T_2, T_1, T_3, T_4\rangle$，所以系统处于安全状态。

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

解. 如果允许：

Sequence	*Available*
	A, B, C, D
$T_1(R)$	1,1,0,0
T_0	1,1,1,2
T_2	2,4,6,6
T_1	3,8,8,6
T_3	3,14,11,8
T_4	3,14,12,12

所以可以被允许。

8.9 Consider the following snapshot of a system:

	Allocation	*Max*	***Need***
	$ABCD$	$ABCD$	$ABCD$
T_0	3014	5117	2103
T_1	2210	3211	1001
T_2	3121	3321	0200
T_3	0510	4612	4102
T_4	4212	6325	2113

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)

Sequence	*Available*
	A, B, C, D
T_2	3,4,2,2
T_1	5,6,3,2
T_3	5,11,4,2

解.

现在 T_0 和 T_4 都需要 3 个 D 资源，所以处于不安全状态。

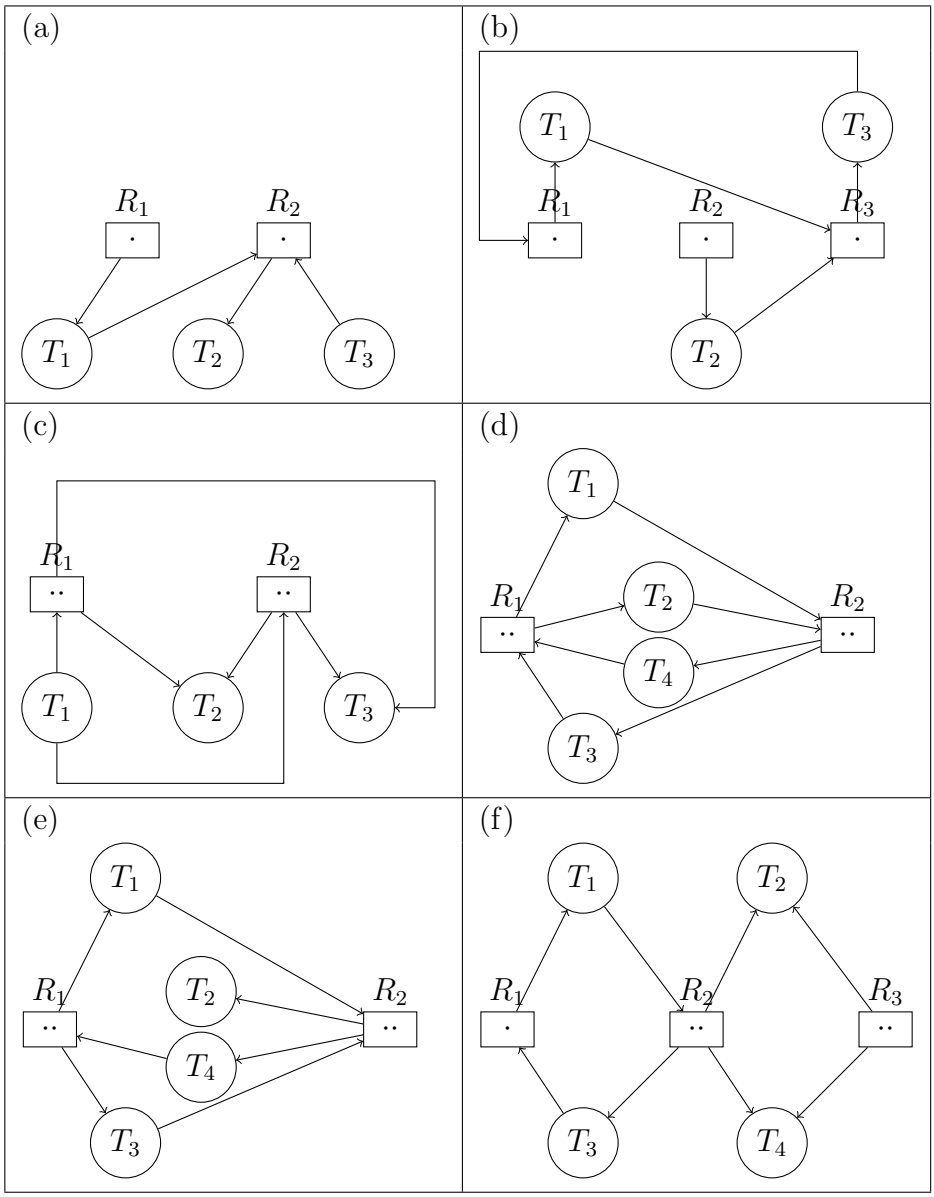
b. *Available*=(1,0,0,2)

Sequence	*Available*
	A, B, C, D
T_1	3,2,1,2
T_2	6,3,3,3
T_4	10,5,4,5
T_0	13,5,5,9
T_3	13,10,6,9

存在安全序列 $\langle T_1, T_2, T_4, T_0, T_3 \rangle$，所以是安全的。

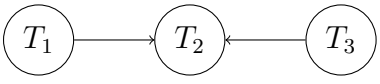
8.18 Which of the six resource-allocation graphs shown in Figure 8.12 illustrate deadlock? For those situations that are deadlocked, provide the cycle of threads and resources. Where

there is not a deadlock situation,illustrate the order in which the threads may complete execution.



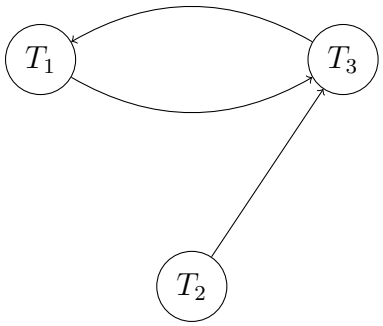
解.

(a) 等待图为



图中没有环，所以不会死锁。

(b) 等待图为



图中有环，所以会死锁。

(c) 虽然等待图中存在环$T_1 \leftrightarrow T_2$，但是，如果将T_2释放，图中就不会存在死锁，所以该图不会死锁。

(d) 该图存在死锁，T_1, T_2 请求 R_2 而 T_3, T_4 请求 R_1，资源已经都被占用。

(e) 该图不存在死锁，释放T_2，再释放T_1（或T_3）就不会导致死锁。

(f) R_2不足，左侧存在死锁，所以该图存在死锁。

8.27 Consider the following snapshot of a system:

	Allocation	*Max*	***Need***
	$ABCD$	$ABCD$	$ABCD$
T_0	1202	4316	3114
T_1	0112	2424	2312
T_2	1240	3651	2411
T_3	1201	2623	1422
T_4	1001	3112	2111

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*=(2,2,2,3)

解.

Sequence	*Available*
	A, B, C, D
T_4	3,2,2,4
T_0	4,4,2,6
T_3	5,6,2,7
T_1	5,7,3,9
T_2	6,9,7,9

存在安全序列 $\langle T_4, T_0, T_3, T_1, T_2 \rangle$，所以系统是安全的。

b. *Available*=(4,4,1,1)

	Sequence	*Available*
		A, B, C, D
	T_4	5,4,1,2
解.	T_2	6,6,5,2
	T_3	7,8,5,3
	T_1	7,9,6,5
	T_0	8,11,6,7

存在安全序列 $\langle T_4, T_2, T_3, T_1, T_0\rangle$，所以系统是安全的。

c. *Available*=(3,0,1,4)

解. 都至少需要一个 B 资源，所以处于不安全状态。

d. *Available*=(1,5,2,2)

	Sequence	*Available*
		A, B, C, D
	T_3	2,7,2,3
解.	T_2	3,9,6,3
	T_1	3,10,7,5
	T_0	4,12,7,7
	T_4	5,12,7,8

存在安全序列 $\langle T_3, T_2, T_1, T_0, T_4\rangle$，所以系统处于安全状态。

8.28 Consider the following snapshot of a system:

	Allocation	*Max*	***Need***	*Available*
	$ABCD$	$ABCD$	$ABCD$	$ABCD$
T_0	3141	6473	3332	2224
T_1	2102	4232	2130	
T_2	2413	2533	0120	
T_3	4110	6332	2222	
T_4	2221	5675	3454	

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.

	Sequence	*Available*
		A, B, C, D
	T_2	4,6,3,7
解.	T_3	8,7,4,7
	T_1	10,8,4,8
	T_0	13,9,8,9
	T_4	16,13,13,13

安全序列为 $\langle T_2, T_3, T_1, T_0, T_4\rangle$。

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

答: 不可以，会导致资源耗尽，但没有释放，所有的进程都不能进行完毕。

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

答: 可以，只要继续完成 T_2，就可以按照相同的安全序列完成所有任务。

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

答: 可以，只要继续完成 T_3，就可以按照安全序列 $\langle T_3, T_2, T_1, T_0, T_4\rangle$完成任务。