

7.7 假设一个系统有 4 个相同类型的资源，并由 3 个进程共享。每个进程最多需要 2 个资源。证明这个系统不会死锁。

7.7

如果三个进程都请求资源，每个进程最多请求 2 个资源，至少有一个进程只能请求资源，而等待其他资源；一共三个进程，因此系统总是能分配至少一个资源来避免死锁。

7.12 假设一个系统具有如下快照：

	Allocation	Max
	A B C D	A B C D
P_0	3 0 1 4	5 1 1 7
P_1	2 2 1 0	3 2 1 1
P_2	3 1 2 1	3 3 2 1
P_3	0 5 1 0	4 6 1 2
P_4	4 2 1 2	6 3 2 5

337

230 第二部分 进程管理

采用银行家算法，确定如下每个状态是否安全的。如果状态是安全的，那么说明进程可以完成的顺序。否则，说明为什么状态是不安全的。

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

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

7.12

	Allocation A B C D	Max A B C D	Need A B C D
P_0	3 0 1 4	5 1 1 7	2 1 0 3
P_1	2 2 1 0	3 2 1 1	1 0 0 1
P_2	3 1 2 1	3 3 2 1	0 2 0 0
P_3	0 5 1 0	4 6 1 2	4 1 0 2
P_4	4 2 1 2	6 3 2 5	2 1 1 3

a. ~~序列~~ $\langle P_2, P_1 \rangle$ 无安全序列，因为无论按何种序列执行，可供分配的资源D始终达不到3个，也就无法运行进程 P_0, P_4

b. 执行 P_1, P_2 后 Available = (6, 3, 3, 3)，可任意执行
 因此安全序列为 $\langle P_1, P_2, P_0, P_3, P_4 \rangle$ ， P_0, P_3, P_4 执行顺序任意

7.13 假设一个系统具有如下的快照：

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

采用银行家算法，回答下面的问题：

- 通过进程可以完成执行的顺序，说明系统处于安全状态。
- 当进程 P_1 的请求为 $(1, 1, 0, 0)$ 时，能否立即允许这一请求？
- 当进程 P_4 的请求为 $(0, 0, 2, 0)$ 时，能否立即允许这一请求？

7.13	Allocation	Max	Need	Available
	A B C D	A B C D	A B C D	A B C D
P_0	2 0 0 1	4 2 1 2	2 2 1 1	3 3 2 1
P_1	3 1 2 1	5 2 5 2	2 1 3 1	
P_2	2 1 0 3	2 3 1 6	0 2 1 3	
P_3	1 3 1 2	1 4 2 4	0 1 1 2	
P_4	1 4 3 2	3 6 6 5	2 2 3 3	
<p>a. 若先执行 P_0, $Available = (5, 3, 2, 2)$, 后可执行 P_3, $Available = (6, 6, 3, 4)$ 已满足所有进程需求, \therefore 安全序列为 $\langle P_0, P_3, P_1, P_2, P_4 \rangle$, P_1, P_2, P_4 可任意顺序执行</p>				
<p>b. P_1 $Available > Need$ \therefore 可以</p>				
<p>c. $Available > Need$ \therefore 可以</p>				
<p>不可以</p>				
<p>若立即允许, 则 ① 没有 C 资源, 则不存在安全序列</p>				