操作系统（2024-2025）

作业 #3:
241210 死锁

截止日期：2024 年 12 月 11 日

张梓卫 (学号：10235101526)

问题 1

1　7.7

假设一个系统有 4 个相同类型的资源，并由三个进程共享。每个进程最多需要 2 个资源，证明这个系统不会死锁。

解答

四个资源，三个进程，必然有一个进程持有至少两个资源，该进程可以完成，完成后释放这两个资源，其余进程都可以完成。不会死锁。

数学证明：

- 总资源数为 R；进程数为 N；每个进程最多需要的资源为 K。

系统不会死锁的条件是：

$$R \geq (N-1) \cdot K + 1$$

$(N-1) \cdot K$ 是防止死锁的关键部分：即使 $N-1$ 个进程各占用最多的 K 个资源，系统仍需要保留至少 1 个资源，来破坏循环等待条件。

在本题中：

- $R = 4$（系统有 4 个资源）；$N = 3$（系统有 3 个进程）；$K = 2$（每个进程最多需要 2 个资源）。

$$R \geq (3-1) \cdot 2 + 1 = 4$$

$R = 4$ 满足该条件，因此系统不会死锁。

2　7.12

假设一个系统具有如下快照：

进程	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

表 1: 系统资源快照

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)

解答

Need 矩阵为：

	A	B	C	D
P_0	2	1	0	3
P_1	1	0	0	1
P_2	0	2	0	0
P_3	4	1	0	2
P_4	2	1	1	3

按银行家算法的安全性检查步骤：

1. 找满足 $\text{Need}(P_i) \leq \text{Work}$ 的未完成进程。

$$\text{Need}(P_2) = (0,2,0,0) \leq (0,3,0,1) \quad \checkmark$$
$$\text{Work} = \text{Work} + \text{Allocation}(P_2) = (0,3,0,1) + (3,1,2,1) = (3,4,2,2)$$
$$\text{Finish}[P_2] = T$$

2. 现在 $\text{Work} = (3,4,2,2)$，继续找下一个可执行的进程。

$$\text{Need}(P_1) = (1,0,0,1) \leq (3,4,2,2) \quad \checkmark$$
$$\text{Work} = (3,4,2,2) + (2,2,1,0) = (5,6,3,2)$$
$$\text{Finish}[P_1] = T$$

3. 继续，$\text{Work} = (5,6,3,2)$。

$$\text{Need}(P_3) = (4,1,0,2) \leq (5,6,3,2) \quad \checkmark$$
$$\text{Work} = (5,6,3,2) + (0,5,1,0) = (5,11,4,2)$$
$$\text{Finish}[P_3] = T$$

4. 现在 $\text{Work} = (5,11,4,2)$，剩下 P_0 和 P_4 未完成。

$$\text{Need}(P_4) = (2,1,1,3) \leq (5,11,4,2)$$

前三项满足，但最后一项 $3 \leq 2$ 不成立，无法执行 P_4。

$$\text{Finish} = [F,T,T,T,F]$$

仍有未完成进程（P_0, P_4 均无法得到所需资源），因此该状态不安全。

故：当 $\textbf{Available} = (0,3,0,1)$ 时，状态不安全。

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

$$\text{Work} = \text{Available} = (1, 0, 0, 2), \quad \text{Finish} = [F, F, F, F, F]$$

$$\text{Need}(P_1) = (1, 0, 0, 1) \leq (1, 0, 0, 2) \quad \checkmark$$

$$\text{Work} = (1, 0, 0, 2) + (2, 2, 1, 0) = (3, 2, 1, 2)$$

$$\text{Finish}[P_1] = T$$

2. Work $= (3, 2, 1, 2)$，检查 P_2:

$$\text{Need}(P_2) = (0, 2, 0, 0) \leq (3, 2, 1, 2) \quad \checkmark$$

$$\text{Work} = (3, 2, 1, 2) + (3, 1, 2, 1) = (6, 3, 3, 3)$$

$$\text{Finish}[P_2] = T$$

3. Work $= (6, 3, 3, 3)$，检查 P_3:

$$\text{Need}(P_3) = (4, 1, 0, 2) \leq (6, 3, 3, 3) \quad \checkmark$$

$$\text{Work} = (6, 3, 3, 3) + (0, 5, 1, 0) = (6, 8, 4, 3)$$

$$\text{Finish}[P_3] = T$$

4. Work $= (6, 8, 4, 3)$，检查 P_4:

$$\text{Need}(P_4) = (2, 1, 1, 3) \leq (6, 8, 4, 3) \quad \checkmark$$

$$\text{Work} = (6, 8, 4, 3) + (4, 2, 1, 2) = (10, 10, 5, 5)$$

$$\text{Finish}[P_4] = T$$

$$\text{Need}(P_0) = (2, 1, 0, 3) \leq (10, 10, 5, 5) \quad \checkmark$$

$$\text{Work} = (10, 10, 5, 5) + (3, 0, 1, 4) = (13, 10, 6, 9)$$

$$\text{Finish}[P_0] = T$$

此时：

$$\text{Finish} = [T, T, T, T, T]$$

所有进程均能完成。
执行顺序为 $P_1 \to P_2 \to P_3 \to P_4 \to P_0$。
结论：当 Available $= (1, 0, 0, 2)$ **时，状态安全，可行的安全序列为：**$(P_1, P_2, P_3, P_4, P_0)$。

3 7.13

假设一个系统具有如下快照：

进程	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	

表 2: 系统资源快照

采用银行家算法，回答下面的问题：

a. 通过进程可以完成执行的顺序，说明系统处于安全状态。

b. 当进程 P_1 的请求为 $(1,1,0,0)$ 时，能否立即允许这一请求？

c. 当进程 P_4 的请求为 $(0,0,2,0)$ 时，能否立即允许这一请求？

解答

Need 矩阵为：

	A	B	C	D
P_0	2	2	1	1
P_1	2	1	3	1
P_2	0	2	1	3
P_3	0	1	1	2
P_4	2	2	3	3

初始：

$$\text{Work} = \text{Available} = (3,3,2,1), \quad \text{Finish} = [F,F,F,F,F]$$

检查可满足的进程（Need $\leq$ Work）：

$$\text{Need}(P_0) = (2,2,1,1) \leq (3,3,2,1) \checkmark$$

$$\text{Work} = (3,3,2,1) + \text{Allocation}(P_0) = (3,3,2,1) + (2,0,0,1) = (5,3,2,2)$$

$$\text{Finish}[P_0] = T$$

2. 现在 Work $= (5,3,2,2)$，检查其他进程：

$$\text{Need}(P_3) = (0,1,1,2) \leq (5,3,2,2) \checkmark$$

$$\text{Work} = (5,3,2,2) + (1,3,1,2) = (6,6,3,4)$$

$$\text{Finish}[P_3] = T$$

3. Work $= (6,6,3,4)$，继续：

$$\text{Need}(P_2) = (0,2,1,3) \leq (6,6,3,4) \checkmark$$

$$\text{Work} = (6,6,3,4) + (2,1,0,3) = (8,7,3,7)$$

$$\text{Finish}[P_2] = T$$

4. Work $= (8,7,3,7)$，检查 P_1:

$$\text{Need}(P_1) = (2,1,3,1) \leq (8,7,3,7) \checkmark$$

$$\text{Work} = (8,7,3,7) + (3,1,2,1) = (11,8,5,8)$$

$$\text{Finish}[P_1] = T$$

5. Work $= (11,8,5,8)$，最后检查 P_4:

$$\text{Need}(P_4) = (2,2,3,3) \leq (11,8,5,8) \checkmark$$

$$\text{Work} = (11,8,5,8) + (1,4,3,2) = (12,12,8,10)$$

$$\text{Finish}[P_4] = T$$

所有进程均可完成：

$$\text{Finish} = [T,T,T,T,T]$$

安全序列为：

$$(P_0 \to P_3 \to P_2 \to P_1 \to P_4).$$

故系统处于安全状态。

(b) 当进程 P_1 的请求为 $(1,1,0,0)$ 时，能否立即允许这一请求？

请求向量为 $R_1 = (1,1,0,0)$。
检查：1. 请求是否不超过其需要：

$$\text{Need}(P_1) = (2,1,3,1), \quad R_1 = (1,1,0,0)$$

$$(1 \leq 2,\ 1 \leq 1,\ 0 \leq 3,\ 0 \leq 1) \checkmark$$

2. 请求是否不超过当前可用：

$$\text{Available} = (3,3,2,1),\ R_1 = (1,1,0,0)$$

$$(1 \leq 3,\ 1 \leq 3,\ 0 \leq 2,\ 0 \leq 1) \checkmark$$

可尝试分配：分配后更新：

$$\text{Available}_{new} = (3,3,2,1) - (1,1,0,0) = (2,2,2,1)$$

$$\text{Allocation}(P_1)_{new} = (3+1, 1+1, 2+0, 1+0) = (4,2,2,1)$$

$$\text{Need}(P_1)_{new} = (2-1, 1-1, 3-0, 1-0) = (1,0,3,1)$$

在此新状态下重复安全性检查：

$$\text{Work} = (2,2,2,1)$$

- 检查 P_0: $\text{Need}(P_0) = (2,2,1,1) \le (2,2,2,1)$？是的。执行 P_0：

$$\text{Work} = (2,2,2,1) + (2,0,0,1) = (4,2,2,2)$$

- 检查 P_3: $\text{Need}(P_3) = (0,1,1,2) \le (4,2,2,2)$？是的。执行 P_3：

$$\text{Work} = (4,2,2,2) + (1,3,1,2) = (5,5,3,4)$$

- 检查 P_2: $\text{Need}(P_2) = (0,2,1,3) \le (5,5,3,4)$？是的。执行 P_2：

$$\text{Work} = (5,5,3,4) + (2,1,0,3) = (7,6,3,7)$$

- 检查更新后 P_1: $\text{Need}(P_1) = (1,0,3,1) \le (7,6,3,7)$？是的。执行 P_1：

$$\text{Work} = (7,6,3,7) + (4,2,2,1) = (11,8,5,8)$$

- 检查 P_4: $\text{Need}(P_4) = (2,2,3,3) \le (11,8,5,8)$？是的。执行 P_4：

$$\text{Work} = (11,8,5,8) + (1,4,3,2) = (12,12,8,10)$$

所有进程均可完成，状态仍然安全。
结论：对于 P_1 的请求 $(1,1,0,0)$ 可以立即满足。

(c) 当进程 P_4 的请求为 $(0,0,2,0)$ 时能否立即满足？

检查 P_4 的请求 $R_4 = (0,0,2,0)$：
1. 请求是否不超过其需要：

$$\text{Need}(P_4) = (2,2,3,3), \quad R_4 = (0,0,2,0)$$

$$(0 \le 2,\ 0 \le 2,\ 2 \le 3,\ 0 \le 3)\ \checkmark$$

2. 请求是否不超过当前可用：

$$\text{Available} = (3,3,2,1), \quad R_4 = (0,0,2,0)$$

$$(0 \le 3,\ 0 \le 3,\ 2 \le 2,\ 0 \le 1)\ \checkmark$$

可暂时分配该请求：

更新后：

$$\text{Available}_{new} = (3,3,2,1) - (0,0,2,0) = (3,3,0,1)$$

$$\text{Allocation}(P_4)_{new} = (1,4,3+2,2) = (1,4,5,2)$$

$$\text{Need}(P_4)_{new} = (2,2,3-2,3) = (2,2,1,3)$$

再次进行安全性检查：

$$\text{Work} = (3,3,0,1)$$

依次检查进程：
- P_0: $\text{Need}(P_0) = (2,2,1,1) \leq (3,3,0,1)$?
检查 C 资源：Need C=1 $\leq$ Work C=0？不成立。P_0 无法执行。
- P_3: $\text{Need}(P_3) = (0,1,1,2) \leq (3,3,0,1)$?
Need C=1 $\leq$0？不成立。P_3 无法执行。
- P_2: $\text{Need}(P_2) = (0,2,1,3) \leq (3,3,0,1)$?
Need C=1 $\leq$0？不成立。P_2 无法执行。
- P_1: $\text{Need}(P_1) = (2,1,3,1) \leq (3,3,0,1)$?
Need C=3 $\leq$0？不成立。P_1 无法执行。
- P_4: $\text{Need}(P_4)_{new} = (2,2,1,3) \leq (3,3,0,1)$?
Need C=1 $\leq$0？不成立。P_4 无法执行。
没有任何进程可以执行，系统进入死锁状态，故不安全。
对于 P_4 的请求 $(0,0,2,0)$ 不可立即满足，否则系统不安全。