

操作系统原理第四章作业

姓名：马福泉 学号：23336179 截止日期：2025 年 5 月 6 日

完成日期：2025 年 5 月 5 日

Question 1:

7.1 假设有如图 7.9 所示的交通死锁情况：

- 证明这个例子中实际上包括了死锁发生的 4 个必要条件。
- 给出一个简单的规则使这个系统避免死锁。

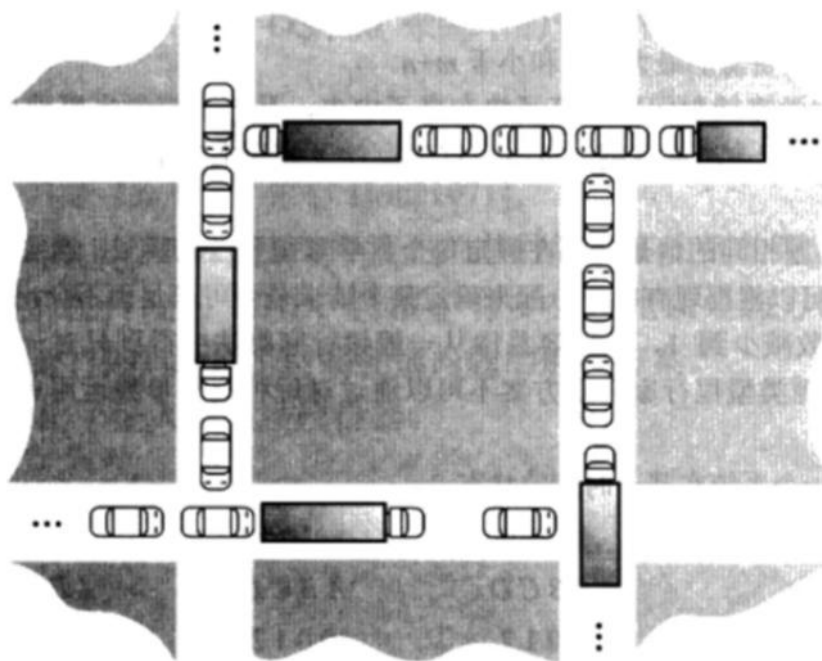


图 7.9 习题 7.1 交通死锁

Answer 1:

(1)

- ① 互斥：四个交叉路口在同一个时间点只能通过一辆车；
- ② 占有并等待：每辆车已占据部分道路（如横向或纵向车道），同时等待被其他车辆占用的道路（如交叉口）。
- ③ 非抢占：当路口有车辆时其它车辆无法经过该路口（非抢占）；
- ④ 图中存在一个循环等待链，例如：车 A 等待车 B 占用的道路，车 B 等待车 C，车 C 等待车 D，车 D 又等待车 A，形成闭环。

(2) 所有车辆在进入交叉口前，必须一次性获取全部所需道路资源，确保不会部分占用后等待。

具体实现：可以设置红绿灯信号灯，前一分钟可以让横向车通过，后一分钟可以让纵向车通过，每个一分钟进行一次轮转，车辆必须提前判断其路径上的所有路段是否可完全通行，否则需在进入前等待。

Question 2: 7.5 在一个真实的计算机系统中，可用的资源和进程对资源的要求都不会持续很久(几个月)。资源会损坏和被替换，新的进程会进入和离开系统，新的资源会被购买和加入系统。如果用银行家算法控制死锁，下面哪些变化在什么情况下是安全的(不会导致死锁)?

- a.增加可用资源(新的资源被加入系统)
- b.减少可用资源(资源被从系统中永久性地移出)
- c.增加一个进程的 **Max**(进程需要更多的资源，超过所允许的资源)
- d.减少一个进程的 **Max**(进程不再需要那么多资源)
- e.增加进程的数量
- f.减少进程的数量

Answer 2:

- (1) 增加可用资源：安全。资源增加不会破坏现有安全状态，反而可能将系统从非安全状态转为安全状态。
- (2) 减少可用资源：可能不安全。如果减少的资源数量较少，并且当前系统的资源分配仍然满足银行家算法的安全条件（即存在一个安全序列），那么系统仍然是安全的。如果减少的资源数量较多，无法找到一个安全序列，则系统可能会进入不安全状态，从而可能导致死锁。。
- (3) 增加进程的 **Max**：可能不安全。如果当前系统的可用资源足够多，即使增加进程的 **Max** 值，系统仍然能够满足进程的资源请求，并且能够找到一个安全序列，则系统仍然是安全的。如果增加进程的 **Max** 值后，系统的可用资源不足以满足进程的资源请求，或者无法找到一个安全序列，则系统可能会进入不安全状态，从而可能导致死锁。
- (4) 减少进程的 **Max**：安全。类似于释放资源，不会破坏安全状态。减少进程的 **Max** 值后，系统的资源分配压力减轻，即使其他进程请求更多的资源，系统也有更大的余地来满足这些请求，而不会导致资源不足，从而避免死锁。

- (5) 增加进程数量：可能不安全。如果当前系统的可用资源足够多，即使增加新的进程，系统仍然能够满足所有进程的资源请求，并且能够找到一个安全序列，则系统仍然是安全的。• 如果增加新的进程后，系统的可用资源不足以满足所有进程的资源请求，或者无法找到一个安全序列，则系统可能会进入不安全状态，从而可能导致死锁
- (6) 减少进程数量：安全。减少进程的数量后，系统的资源分配压力减轻，即使某些进程请求更多的资源，系统也有更大的余地来满足这些请求，而不会导致资源不足，从而避免死锁。

Question 3:

7.11 考虑下面的一个系统在某一时刻的状态。

	<u>Allocation</u>	<u>Max</u>	<u>Available</u>
	<i>A B C D</i>	<i>A B C D</i>	<i>A B C D</i>
P_0	0 0 1 2	0 0 1 2	1 5 2 0
P_1	1 0 0 0	1 7 5 0	
P_2	1 3 5 4	2 3 5 6	
P_3	0 6 3 2	0 6 5 2	
P_4	0 0 1 4	0 6 5 6	

使用银行家算法回答下面问题：

- $Need$ 矩阵的内容是怎样的？
- 系统是否处于安全状态？
- 如果从进程 P_1 发来一个请求 $(0,4,2,0)$ ，这个请求能否立刻被满足？

Answer 3:

(a)

$Need = Max - Allocation$

P_0	0	0	0	0
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) 令 $Work = Available = (1, 5, 2, 0)$, $Finish[i] = false$;
 先分配给 P_0 , then $Work = (1, 5, 3, 2)$, $Finish[0] = true$;
 然后分配给 P_2 , then $Work = (2, 8, 8, 6)$, $Finish[2] = true$;
 然后分配给 P_1 , then $Work = (3, 8, 8, 6)$, $Finish[1] = true$;
 然后分配给 P_3 , then $Work = (3, 14, 11, 8)$, $Finish[3] = true$;
 然后分配给 P_4 , then $Work = (3, 14, 12, 12)$, $Finish[4] = true$;
 所以 $\{P_0, P_2, P_1, P_3, P_4\}$ 是一个安全序列, 系统处于安全状态。

(c) 假设可以立即分配,
 令 $Work = Available = (1, 1, 0, 0)$, $Finish[i] = false$;
 先分配给 P_0 , then $Work = (1, 1, 1, 2)$, $Finish[0] = true$;
 然后分配给 P_2 , then $Work = (2, 4, 6, 6)$, $Finish[2] = true$;
 然后分配给 P_1 , then $Work = (3, 8, 8, 6)$, $Finish[1] = true$;
 然后分配给 P_3 , then $Work = (3, 14, 11, 8)$, $Finish[3] = true$;
 最后分配给 P_4 , then $Work = (3, 14, 12, 12)$, $Finish[4] = true$;
 所以 $\{P_0, P_2, P_1, P_3, P_4\}$ 是一个安全序列, 系统仍然处于安全状态。

Question 4:

假设具有 5 个进程的进程集合 $P = \{P_0, P_1, P_2, P_3, P_4\}$, 系统中有三类资源 A, B, C, 假设在某时刻有如下状态:

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

当前系统是否处于安全状态? 若系统中的可利用资源 Available 为 (0, 6, 2), 系统是否安全? 若系统处在安全状态, 请给出安全序列; 若系统处在非安全状态, 简要说明原因。

Answer 4:

(1) 当前系统安全

令 $Work = Available = (1, 4, 0)$, $Finish[i] = [False]$

- ① $P_2: Need = (1, 0, 0) \leq (1, 4, 0) \rightarrow$ 可执行。 $Work += (1, 3, 5) = (2, 7, 5)$
 $Finish[2] = True$
- ② $P_0: Need = (0, 0, 1) \leq (2, 7, 5) \rightarrow$ 可执行。 $Work += (0, 0, 3) = (2, 7, 8)$
 $Finish[0] = True$
- ③ $P_1: Need = (0, 7, 5) \leq (2, 7, 8) \rightarrow$ 可执行。 $Work += (1, 0, 0) = (3, 7, 8)$
 $Finish[1] = True$

④ P3: Need = (0,6,2) \leq (3,7,8) \rightarrow 可执行。Work += (0,0,2) = (3,7,10)

Finish[3] = True

⑤ P4: Need = (0,6,4) \leq (3,7,10) \rightarrow 可执行。Work += (0,0,1) = (3,7,11)

Finish[4] = True

安全序列: P2 ,P0 ,P1 ,P3 ,P4。所以当前系统安全

(2) 不安全, 分析如下

令 Work = (0, 6, 2)

Fin[i] = [False]

寻找可执行进程 :

P0: Need = (0,0,1) \leq (0,6,2) \rightarrow 可执行。

Work += (0,0,3) = (0,6,5)

Finish[0] = True

P2: Need = (1,0,0) \leq (0,6,5)不成立。

P3: Need = (0,6,2) \leq (0,6,5) \rightarrow 可执行。

Work += (0,0,2) = (0,6,7)

Finish[3] = True

P4: Need = (0,6,4) \leq (0,6,7) \rightarrow 可执行。

Work += (0,0,1) = (0,6,8)

Finish[4] = True

P1: Need = (0,7,5) \leq (0,6,8)不成立。

P2: Need = (1,0,0) \leq (0,6,8)不成立。

剩余进程 P1 和 P2 无法完成(资源不足)。

系统不安全, 原因是无法找到完整的安全序列。