

银行家算法问题

吕鹏 2016011902

有三类资源 A(17)、B(5)、C(20)。有 5 个进程 P1-P5。T0 时刻系统状态如下：

	最大需求	已分配
P1	5 5 9	2 1 2
P2	5 3 6	4 0 2
P3	4 0 11	4 0 5
P4	4 2 5	2 0 4
P5	4 2 4	3 1 4

T0 时刻是否是安全状态，给出安全序列。

T0 时刻，P2: Request(0, 3, 4)，能否分配，为什么？

在(2)的基础上 P4: Request(2, 0, 1)，能否分配，为什么？

在(3)的基础上 P1: Request(0, 2, 0)，能否分配，为什么？

解：

(1)：

	Max	Allocation	Need
	A B C	A B C	A B C
P1	5 5 9	2 1 2	3 4 7
P2	5 3 6	4 0 2	1 3 4
P3	4 0 11	4 0 5	0 0 6
P4	4 2 5	2 0 4	2 2 1
P5	4 2 4	3 1 4	1 1 0

剩余资源如下：

Available		
A	B	C
2	3	3

开始进行安全性检查，Work 向量初始化为 Available：

Work		
A	B	C
2	3	3

依次检查未完成的进程，检测到 P4 可分配，于是将 Work 向量更新如下：

Work		
A	B	C
4	3	7

依次检查未完成的进程，检测到 P2 可分配，于是将 Work 向量更新如下

Work		
A	B	C
8	3	9

依次检查未完成的进程，检测到 P3 可分配，于是将 Work 向量更新如下：

Work		
A	B	C
12	3	14

依次检查未完成的进程，检测到 P5 可分配，于是将 Work 向量更新如下：

Work		
A	B	C
15	4	18

依次检查未完成的进程，检测到 P1 可分配，于是将 Work 向量更新如下：

Work		
A	B	C
17	5	20

存在安全序列 (P4, P2, P3, P5, P1)，因此 T0 时刻是安全状态。

(2) :

在 T0 时刻，进程 P2

满足 $\text{Request}(0, 3, 4) \leq \text{Need}(1, 3, 4)$

但是不满足 $\text{Request}(0, 3, 4) \leq \text{Available}(2, 3, 3)$

因此当前没有足够的资源，该请求需要等待，不予分配资源。

(3) :

当前，进程 P4

满足 $\text{Request}(2, 0, 1) \leq \text{Need}(2, 2, 1)$

满足 $\text{Request}(2, 0, 1) \leq \text{Available}(2, 3, 3)$

因此我们尝试将资源分配给 P4，并判断分配完该需求后是否处于安全状态：

同(1)，我们得到如下 Work 更新表和安全序列：

	Work		
	A	B	C
P4	4	3	7
P2	8	3	9
P3	12	3	14
P5	15	4	18
P1	17	5	20

存在安全序列 (P4, P2, P3, P5, P1)，因此该请求分配后仍为安全状态，因此该请求可以满足，可以为其分配所请求的资源。

(4) :

当前，进程 P1

满足 $\text{Request}(0, 2, 0) \leq \text{Need}(3, 4, 7)$

满足 $\text{Request}(0, 2, 0) \leq \text{Available}(0, 3, 2)$

因此我们尝试将资源分配给 P1，并判断分配完该需求后是否处于安全状态：

流程同第(1)小问，我们发现，分配完 $\text{Request}(0, 2, 0)$ 后， $\text{Available}(0, 3, 2)$ 不能满足任何进程的需求，即无法找到合法的安全序列，分配请求后进入不安全状态。因此，该请求需要等待，不予分配资源。