#### 银行家算法问题

###### 谭琳2018012119

设系统中有3种类型的资源（A，B，C）和5个进程P1、P2、P3、P4、P5，A资源的数量为17，B资源的数量为5，C资源的数量为20。在T0时刻系统状态见下表（T0时刻系统状态表）所示。系统采用银行家算法实施死锁避免策略。

T0时刻系统状态表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 最大资源需求 | 已分配资源数量 | 系剩余资源数量 |
| A B C | A B C | A B C |
| P1 | 5 5 9 | 2 1 2 | 2 3 3 |
| 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 |  |

(1)   T0时刻是否为安全状态？若是，请给出安全序列。

(2)   在T0时刻若进程P2请求资源（0，3，4），是否能实施资源分配？为什么？

(3)   在（2）的基础上，若进程P4请求资源（2，0，1），是否能实施资源分配？为什么？

(4)   在（3）的基础上，若进程P1请求资源（0，2，0），是否能实施资源分配？为什么？

解：

(1)安全性算法:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Work | Need | Allocation | Work\_Allocation | Finish |
| A B C | A B C | A B C | A B C |
| P5 | 2 3 3 | 1 1 0 | 3 1 4 | 5 4 7 | T |
| P4 | 5 4 7 | 2 2 1 | 2 0 4 | 7 4 11 | T |
| P3 | 7 4 11 | 0 0 6 | 4 0 5 | 11 4 16 | T |
| P2 | 11 4 16 | 1 3 4 | 4 0 2 | 15 4 18 | T |
| P1 | 15 4 18 | 3 4 7 | 2 1 2 | 17 5 20 | T |

存在着一个安全序列{P5,P4,P3,P2,P1}，故系统是安全的。

(2)P2请求资源：P2发出请求向量Request(0,3,4)，系统按银行家算法进行检查：

①Request2(0,3,4)<=Need2(1,3,4);

②Request2(0,3,4)>Available(2,3,3)，申请超过预定需求，所以不给予分配资源。

(3)P4请求资源：P4发出请求向量Request(2,0,1)，系统按银行家算法进行检查：

①Request4(2,0,1)<=Need4(2,2,1);

②Request4(2,0,1)<=Available(2,3,3);

对P4的申请（2，0，1）进行预分配，预分配后系统状态：

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

安全性算法：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Work | Need | Allocation | Work\_Allocation | Finish |
| A B C | A B C | A B C | A B C |
| P4 | 0 3 2 | 0 2 0 | 4 0 5 | 4 3 7 | T |
| P2 | 4 3 7 | 1 3 4 | 4 0 2 | 8 3 9 | T |
| P3 | 8 3 9 | 0 0 6 | 4 0 5 | 12 3 14 | T |
| P5 | 12 3 14 | 1 1 0 | 3 1 4 | 15 4 18 | T |
| P1 | 15 4 18 | 3 4 7 | 2 1 2 | 17 5 20 | T |

存在着一个安全序列{P4,P2,P3,P5,P1}，故系统是安全的。

(4)P1请求资源：P1发出请求向量Request(0,2,0)，系统按银行家算法进行检查：

①Request1(0,2,0)<=Need1(3,4,7);

②Request1(0,2,0)<=Available(0,3,2);

对P1的申请（0，2，0）进行预分配，预分配后系统状态：

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

有效资源不能满足任何进程的Need值,所以对P1的申请进行分配后，系统进入不安全状态，因此不能分配资源。