**操作系统第二次平时成绩作业**

--贾奥同 2018013344

**银行家算法问题**



解：

（1）首先列出Max、Allocation以及Need矩阵

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

已知A、B、C三类资源个数分别为17、5、20，故目前剩下的资源分别为2、3、3。

再列出Work、Need、Allocation、Work&allocation矩阵并判断是否为安全状态

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Work | | | Need | | | Allocation | | | Work&allocation | | |  |
| A | B | C | A | B | C | A | B | C | A | B | C |
| P4 | 2 | 3 | 3 | 2 | 2 | 1 | 2 | 0 | 4 | 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 | 2 | 2 | 1 | 3 | 1 | 4 | 15 | 4 | 18 | T |
| P1 | 15 | 4 | 18 | 3 | 4 | 7 | 2 | 1 | 2 | 17 | 5 | 20 | T |

故安全序列为：P4-->P2-->P3-->P5-->P1，可以安全分配

（2）根据（1）可知Need2(1,3,4)、Available(2,3,3)

P2:Request2(0,3,4)，系统用银行家算法进行检查：

Request2(0,3,4)<=Need2(1,3,4)

Request2(0,3,4)>Available(2,3,3)

故P2等待，不能完成分配

1. 根据（1）可知Need4(2,2,1)、Available(2,3,3)

在（2）的基础上，P4:Request4(2,0,1)，系统用银行家算法进行检查

Request4(2,0,1)<=Need4(2,2,1)

Request4(2,0,1)<=Available(2,3,3)

更新Max、Allocation以及Need矩阵得到

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 | 0 | 2 | 0 |
| P5 | 4 | 2 | 4 | 3 | 1 | 4 | 1 | 1 | 0 |

且Available（0，3，2）

再更新Work、Need、Allocation、Work&allocation矩阵并判断是否为安全状态

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Work | | | Need | | | Allocation | | | Work&allocation | | |  |
| 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 | 2 | 2 | 1 | 3 | 1 | 4 | 15 | 4 | 18 | T |
| P1 | 15 | 4 | 18 | 3 | 4 | 7 | 2 | 1 | 2 | 17 | 5 | 20 | T |

故安全序列为：P4-->P2-->P3-->P5-->P1，可以安全分配

由（3）可知Need1(3,4,7)、Available(0,3,2)

P1:Request1(0,2,0)，系统用银行家算法进行检查

Request1(0,2,0)<=Need1(3,4,7)

Request1(0,2,0)<=Available(0,3,2)

更新Max、Allocation以及Need矩阵得到

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Max | | | Allocation | | | Need | | |
| A | B | C | A | B | C | A | B | C |
| P1 | 5 | 5 | 9 | 2 | 1 | 2 | 3 | 2 | 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 | 0 | 2 | 0 |
| P5 | 4 | 2 | 4 | 3 | 1 | 4 | 1 | 1 | 0 |

且Available（0，1，2）

此时Available（0，1，2）不能够为任何一进程分配资源系统不安全，不能分配