**操作系统第二次作业——银行家算法**

**张缤予2018010982**



由题中已知可画得如下表格

其中Need[i,j] = Max[i,j] - Allocation[i,j]。

T0时刻的Available的ABC值如下 A = 17 -  = 17-15 = 2

B = 5 -  = 5-2 = 3 C = 20 -  = 20 -17 =3

资源分配表如下

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 进程 | 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 | 2 | 3 | 3 |
| 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 |  |  |  |

（1）

T0时刻是安全状态。

T0时设P1~P4的Finish=Flase，根据银行家算法判断并计算安全序列

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程 | Work | Need | Allocation | Work+Allocation | Finish |
| 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 | True |
| P2 | 4 3 7 | 1 3 4 | 4 0 2 | 8 3 9 | True |
| P3 | 8 3 9 | 0 0 6 | 4 0 5 | 12 3 14 | True |
| P5 | 12 3 14 | 1 1 0 | 3 1 4 | 15 4 18 | True |
| P1 | 15 4 18 | 3 4 7 | 2 1 2 | 17 5 20 | True |

所以安全序列为P4->P2->P3->P5->P1，T0时刻为安全状态。

T0时刻，P2 = Request(0,3,4)

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

而Request2(0,3,4) >Available(2,3,3)，故不能分配

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

Request4(2,0,1) < Available4(2,3,3)

资源分配表如下

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

根据银行家算法进行判断并计算安全序列

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 进程 | 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 | True |
| P2 | 4 3 7 | 1 3 4 | 4 0 2 | 8 3 9 | True |
| P3 | 8 3 9 | 0 0 6 | 4 0 5 | 12 3 14 | True |
| P5 | 12 3 14 | 1 1 0 | 3 1 4 | 15 4 18 | True |
| P1 | 15 4 18 | 3 4 7 | 2 1 2 | 17 5 20 | True |

安全序列为P4->P2->P3->P5->P1，系统安全，故可以将P4所申请的资源分配

1. Request1(0,2,0) < Need1(3,4,7)

Request1(0,2,0) < Available1(0,3,2)

资源分配表如下

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

此时的可用资源Available(0,1,2)不能满足任何进程的资源需求，故系统进入不安全状态，此时不能将资源分配给P1