**作业2：银行家算法**

**题目：**

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 |

(1)T0时刻是否为安全状态，给出安全系列。

(2)T0时刻，P2: Request(0,3,4)，能否分配，为什么？

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

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

**解答：**

（1）Need + Allocation = Max

根据上述公式，绘制出T0时刻进程资源分配情况表格：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 资源  进程  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 | 6 | 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 |

**执行安全检测算法**

设工作向量Work = Available = [2, 3, 3]

设完成向量Finish = [false, false, false, false false]

设i = 4，有对于任意的j(1 <= j <= 3) 有Need[4, j] <= Work[j]

当进程P4执行完毕，有Work = Work + Allocation[4] = [4, 3, 7]

Finish[4] = true

设i = 2, 有对于任意的j(1 <= j <= 3)，有Need[2, j] <= Work[j]

当进程P2执行完毕，有Work = Work + Allocation[2] = [8, 3, 9], Finish[2] = true

设i = 3, 同理，当进程P1执行完毕Work = Work + Allocation[3] = [12, 3, 14],Finish[1] = true

设i = 5, 同理，当进程P3执行完毕Work = Work + Allocation[5] = [15, 4, 18],Finish[3] = true

i = 1，同理，进程P5可顺利执行完毕。Work = Work + Allocation[1] = [17, 5, 20]

T0时刻为安全状态，存在一个安全序列P4 -> P2 -> P3 -> P5 -> P1

（2）T0时刻，P2发出请求Request(0, 3, 4) <= Need[2](1, 3, 4)

Request(0, 3, 4) > Available(2, 3, 3) 所以无法将资源分配给P2进程，P2进程等待

（3）在（2）的基础上，P4发出请求Request(2, 0, 1)

Request(2, 0, 1) <= Need[4](2, 2, 1)

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

满足上述两个条件，假设为P4分配资源（2, 0, 1）

**则此时资源分配表如下：**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 资源  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 | 6 | 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 |

**执行安全检测算法：**

设工作向量Work = Available = [0, 3, 2]

设完成向量Finish = [false, false, false, false, false]

设i = 4 , 则对于任意的j (1 <= j <= 3)有Need[4, j] < Work[j]

当进程P4执行完毕，有Work = Work + Allocation = [4, 3, 7], Finish[4] = true

此时进程执行的状态到达（1）中P4执行完毕后进程执行的状态，因此，存在一个安全序列：P4 -> P2 -> P3 -> P5 -> P1，则此时刻为安全状态，因此接受Request(2, 0, 1)为P4分配资源

**（4）**在（3）的基础上，P1发出请求Request(2, 0, 1)

Request(2, 0, 1) < Need[1](3, 4, 6)

Request(2, 0, 1) > Available(0, 3, 2)，因此无法给P1分配这个请求的资源，P1进行等待