## 银行家算法问题。



（1）根据Need=Max- Allocation,可得到该时刻Need的值如下：

process Need Available

P1 3 4 7 2 3 3

P2 1 3 4

P3 0 0 6

P4 2 2 1

P5 1 1 0

根据矩阵，可以找到可分配的进程p5, Available变为5 4 7；

接着找到可分配的进程p4, Available变为7 4 11；

接着找到可分配的进程p3, Available变为11 4 16；

接着找到可分配的进程p2, Available变为15 4 18；

接着找到可分配的进程p1, Available变为17 5 20；

存在一个安全序列{p5,p4,p3,p2,p1},系统是安全的。

（2）T0时刻，Request(0,3,4)<Need(1,3,4),但Request(0,3,4)> Available(2,3,3)

尚无足够资源，需等待。

（3）T0时刻，在（2）的基础上，

根据银行家算法进行检查：

Request(2,0,1)<Need(2,0,4),Request(2,0,1)< Available(2,3,3)

修改Available，Allocation4，和Need4：

Available=（0,3,2）Allocation4=(4,0,5)，Need4=(0,2,0)

process Need Available

P1 3 4 7 0 3 2

P2 1 3 4

P3 0 0 6

P4 0 2 0

P5 1 1 0

根据安全性算法进行安全性检查：

可以找到可分配的进程p4, Available变为4 3 7；

接着找到可分配的进程p5, Available变为7 4 11；

接着找到可分配的进程p3, Available变为11 4 16；

接着找到可分配的进程p2, Available变为15 4 18；

接着找到可分配的进程p1, Available变为17 5 20；

存在一个安全序列{ p4, p5,p3,p2,p1},系统是安全的，能够分配资源。

（4）在（3）的基础上，根据银行家算法进行检查：

Request（0,2,0）<Need（3,4,7），Request（0,2,0）<Available（0,3,2）

修改Available，Allocation1，和Need1：

Available=（0,1,2）Allocation1=(2,3,2)，Need1=(3,2,7)

process Need Available

P1 3 2 7 0 1 2

P2 1 3 4

P3 0 0 6

P4 0 2 0

P5 1 1 0

安全性检查：条件Need<= Available(0,1,2)都不成立，不能满足任何进程的要求，系统进入不安全状态。不能分配资源给p1。