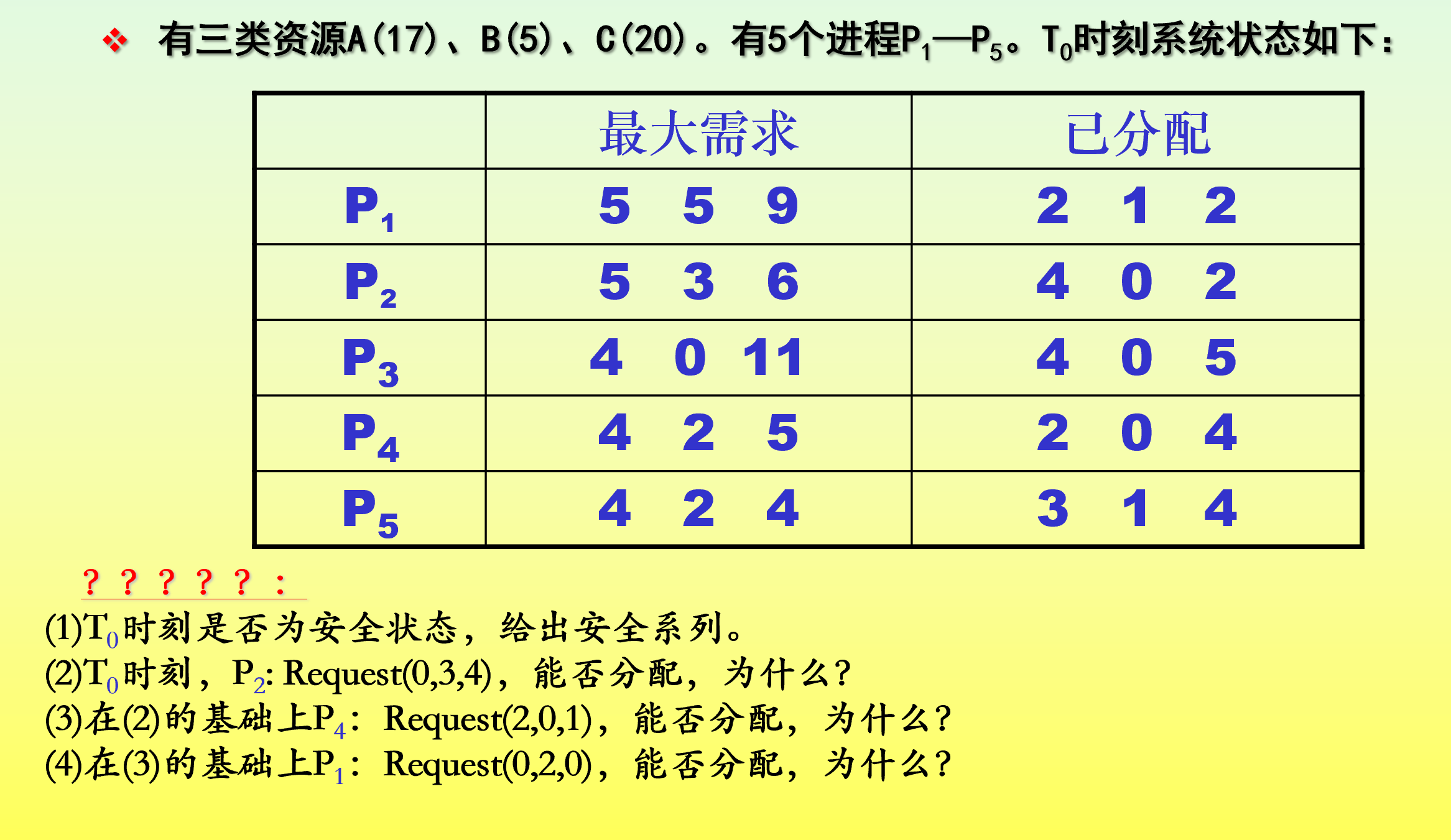
**曹博雯 2018013070**



答：

**1、**可先利用资源总数-已分配资源总数求得可分配资源：

即得到Available中A，B，C的资源数目为

A = 17-2-4-4-2-3 = 2

B = 20-2-2-5-4-4 = 3

C = 20-2-2-5-4-4 = 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Max | Allocation | Need | Available |
| A B C | A B C | A B C | A B C |
|  |  |  |  | 初始：2 3 3 |
| 5)P1 | 5 5 9 | 2 1 2 | 3 4 7 | 1)4 3 7 |
| 3)P2 | 5 3 6 | 4 0 2 | 1 3 4 | 2)7 4 11 |
| 4)P3 | 4 0 11 | 4 0 5 | 0 0 6 | 3)11 4 13 |
| 1)P4 | 4 2 5 | 2 0 4 | 2 2 1 | 4)15 4 18 |
| 2)P5 | 4 2 4 | 3 1 4 | 1 1 0 | 5)17 5 20 |

观察表格可以明显看出P4的Need4（2 2 1）小于Available（2 3 3）

1）故先给P4分配资源，分配后资源向量变为4 3 7（work+Allocation）

2）同理，可看出P5满足分配条件，为P5分配资源,分配后资源向量变为7 4 11（work+Allocation）

3）同理,可看出P2满足分配条件,为P2分配资源,分配后资源向量变为11 4 13（work+Allocation）

4）同理,可看出P3满足分配条件,为P3分配资源,分配后资源向量变为15 4 18（work+Allocation）

5）同理,可看出P1满足分配条件,为P1分配资源,分配后资源向量变为17 5 20（work+Allocation）

**可得安全序列为<P4->P5->P2->P3->P1>**

**2、**在T0时刻：进行银行家算法检查

Request2(0,3,4)<=Need2(1,3,4) 成立；

但是Request2(0,3,4)<=Available(2 ,3,3)不成立

所以进程中请求资源数大于系统分配的资源数，不可分配

**3、**在2、的基础上，先进行银行家算法检查

首先Request4(2,0,1)<=Need4(2,2,1) 成立；

并且Request4(2,0,1)<=Available(2 ,3,3)成立；

所以修改P4的Available4与Need4以及Available，如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Max | Allocation | Need | Available |
| A B C | A B C | A B C | A B C |
|  |  |  |  | 初始：2 3 3 |
|  | Request4(2,0,1) | | | 修改后：0 3 2 |
| 4)P1 | 5 5 9 | 2 1 2 | 3 4 7 | 1)4 3 7 |
| 2)P2 | 5 3 6 | 4 0 2 | 1 3 4 | 2)8 3 9 |
| 3)P3 | 4 0 11 | 4 0 5 | 0 0 6 | 3)12 3 14 |
| 1)P4 | 4 2 5 | 2 0 4 | 2 2 1 | 4)14 4 16 |
|  |  | （4 0 5） | （0 2 0） |  |
| 5)P5 | 4 2 4 | 3 1 4 | 1 1 0 | 5)17 5 20 |

为P4分配资源后，资源向量修改为0 3 2

观察表格可以明显看出P4的Need（0 2 0）小于Available（0 3 2）

1）故先给P4分配资源，分配后资源向量变为4 3 7（work+Allocation）

2）同理，可看出P2满足分配条件，为P2分配资源,分配后资源向量变为8 3 9（work+Allocation）

3）同理,可看出P3满足分配条件,为P3分配资源,分配后资源向量变为12 3 14（work+Allocation）

4）同理,可看出P1满足分配条件,为P1分配资源,分配后资源向量变为14 4 16（work+Allocation）

5）同理,可看出P5满足分配条件,为P5分配资源,分配后资源向量变为17 5 20（work+Allocation）

**可得安全序列为<P4->P2->P3->P1->P5>**

**4、**在3、的基础上，先进行银行家算法检查

首先Request1(0,2,0)<=Need1(3,4,7) 成立；

并且Request1(0,2,0)<=Available(0,3,2)成立；

所以修改P1的Available4与Need4以及Available，如下

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Max | Allocation | Need | Available |
| A B C | A B C | A B C | A B C |
|  |  |  |  | 初始：2 3 3 |
|  | Request4(2,0,1) | | | 修改后：0 3 2 |
|  | Request1(0,2,0) | | | 修改后：0 1 2 |
| P1 | 5 5 9 | 2 1 2 | 3 4 7 |  |
|  |  | （2 3 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 | 2 2 1 |  |
|  |  | （4 0 5） | （0 2 0） |  |
| P5 | 4 2 4 | 3 1 4 | 1 1 0 |  |

因为资源向量Available（0,1,2）无法满足任何进程的资源分配，所以没法找到安全序列，因此系统进入不安全状态，不可分配。