|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | MAX | ALLOCATION | NEED | AVAILABLE |
| 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. 是安全的状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | MAX | NEED | ALLOCATION | WORK+ALLOCATION | FINISH |
| P5 | 4 2 4 | 1 1 0 | 3 1 4 | 2 3 3+3 1 4=5 4 7 | TRUE |
| P4 | 4 2 5 | 2 2 1 | 2 0 4 | 5 4 7+2 0 4=7 4 11 | TRUE |
| P3 | 4 0 11 | 0 0 6 | 4 0 5 | 7 4 11+4 0 5=11 4 16 | TRUE |
| P2 | 5 3 6 | 1 3 4 | 4 0 2 | 11 4 16+4 0 2=15 4 18 | TRUE |
| P1 | 5 5 9 | 3 4 7 | 2 1 2 | 15 4 18+2 1 2=17 5 20 | TRUE |

安全序列为{P5,P4,P3,P2,P1}.

(2) P2申请的资源为（0,3,4），而由图可知P2实际需要的资源为（1,3,4），所以不满足，因此该申请不给予分配。

（3）1.因为REQUEST（2,0,1）<=NEED（2,2,1），转向第二步；

2.因为REQUEST（2,0,1）<=AVAILABLE（2,3,3），转向第三步；

3.试探分配：AVAILABLE（0,3,2）=AVAILABLE（2,3,3）-REQUEST（2,0,1）；

ALLOCATION（4,0,5）=ALLOCATION（2,0,4）+REQUEST（2,0,1）

NEED（0,2,0）=NEED（2,2,1）-REQUEST（2,0,1）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | MAX | NEED | ALLOCATION | WORK+ALLOCATION | FINISH |
| P4 | 4 2 5 | 0 2 0 | 4 0 5 | 4 3 7 | TRUE |
| P5 | 4 2 4 | 1 1 0 | 3 1 4 | 7 4 11 | TRUE |
| P3 | 4 0 11 | 0 0 6 | 4 0 5 | 11 4 16 | TRUE |
| P2 | 5 3 6 | 1 3 4 | 4 0 2 | 15 4 18 | TRUE |
| P1 | 5 5 9 | 3 4 7 | 2 1 2 | 17 5 20 | TRUE |

计算出安全序列为{P4,P5,P3,P2,P1}.

所以试探分配是安全的，正式将资源分配给P4.，所以可以实施资源分配。

（4）1.因为REQUEST（0,2,0）<=NEED（3,4,7），转向第二步；

2.因为REQUEST（0,2,0）<=AVAILABLE（0,3,2），转向第三步；

3.试探分配：AVAILABLE（0,1,2）=AVAILABLE（0,3,2）-REQUEST（0,2,0）；

ALLOCATION（2,3,2）=ALLOCATION（2,1,2）+REQUEST（0,2,0）

NEED（3,2,7）=NEED（3,4,7）-REQUEST（0,2,0）

因为AVAILABLE（0,1,2）不大于等于NEED（3,2,7），因此系统试探分配后处于不安全状态，所以不给予资源分配。