

(1)

由题意可列出所有进程属性如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Request | Allocation | Need | Finish |
| P1 | 5 5 9 | 2 1 2 | 3 4 7 | T |
| P2 | 5 3 6 | 4 0 2 | 1 3 4 | T |
| P3 | 4 0 11 | 4 0 5 | 0 0 6 | T |
| P4 | 4 2 5 | 2 0 4 | 2 2 1 | T |
| P5 | 4 2 4 | 3 1 4 | 1 1 0 | T |

可得安全序列为 P4 P2 P3 P5 P1，因此T0时为安全状态

(2)

由题意可得初始状态的可利用向量Available的值为（2 3 3），而request的值为（0 3 4）大于need的值，因此无法分配

(3)

由题意，可得更新后的资源表为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Request | Allocation | Need | Finish |
| P1 | 5 5 9 | 2 1 2 | 3 4 7 | T |
| P2 | 5 3 6 | 4 0 2 | 1 3 4 | T |
| P3 | 4 0 11 | 4 0 5 | 0 0 6 | T |
| P4 | 4 2 5 | 4 0 5 | 0 2 0 | T |
| P5 | 4 2 4 | 3 1 4 | 1 1 0 | T |

可得安全序列为 P4 P2 P3 P5 P1，因此T0时为安全状态，可以分配

(4)

P1：Request（0，2，0） < Need（3，4，7）

P1：Request（0，2，0） < Available（0，3，2）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Request | Allocation | Need | Finish |
| P1 | 5 5 9 | 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 | 4 0 5 | 0 2 0 |  |
| P5 | 4 2 4 | 3 1 4 | 1 1 0 |  |

此时 Available = (0 1 2)

P1-P5所有的需求均无法被满足，因此系统进入不安全状态，不能分配资源。