**一、死锁（定义出自维基百科）**

死锁（英语：Deadlock），又译为死结，计算机科学名词。当**两个以上**的运算单元，双方都在等待对方停止执行，以取得系统资源，但是没有一方提前退出时，就称为死结。

**二、饥饿（定义出自百度百科）**

进程饥饿，即为Starvation，指当等待时间给进程推进和响应带来明显影响称为进程饥饿。当饥饿到一定程度的进程在等待到即使完成也无实际意义的时候称为饥饿死亡。

**三、自己对于死锁和饥饿的通俗理解**

死锁：五个人围一桌吃饭，每两人中间一根筷子，只有手里有两根筷子时才能吃饭。这五个人同时拿起左手边的筷子，再去拿右手边的筷子，发现右手边的筷子都被占了，谁也吃不了饭。

饥饿：售票口有很多人在排队买票，我希望天黑之前坐车回到学校。但是由于买票人数太多，我到天黑都没有买到票。即使我先在买到了票，也无法在天黑之前坐车回到学校。

**四、练习题**

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

（1）

|  |  |
| --- | --- |
|  | A B C |
|  | 17 5 20 |
|  | 15 2 7 |
| T0 | 2 3 3 |
| P4 | 2 0 4 |
| T1 | 4 3 7 |
| P2 | 4 0 2 |
| T2 | 8 3 9 |
| P1 | 2 1 2 |
| T3 | 10 4 11 |
| P3 | 4 0 5 |
| T4 | 14 4 16 |
| P5 | 3 1 4 |
|  | 17 5 20 |

T0时刻为安全状态，存在一个安全序列：P4->P2->P1->P3->P5

（2）

Request2(0,3,4)≤Need2(1,3,4)

Request2(0,3,4)>Available2(2,3,3)

所以不能分配

（3）

①Request4(2,0,1)≤Need4(2,2,1)

②Request4(2,0,1)≤Available4(2,3,3)

③系统暂时先假定可为P4分配资源，并修改有关数据

|  |  |  |  |
| --- | --- | --- | --- |
|  | Allocation | Need | Available |
| P1 | 2 1 2 | 3 4 7 | 0 3 2 |
| P2 | 4 0 2 | 1 3 4 |  |
| P3 | 4 0 5 | 0 0 6 |  |
| P4 | 4 0 5 | 0 2 0 |  |
| P5 | 3 1 4 | 1 1 0 |  |

④进行安全检查：

|  |  |
| --- | --- |
|  | A B C |
|  | 17 5 20 |
|  | 17 2 18 |
| T0 | 0 3 2 |
| P4 | 4 0 5 |
| T1 | 4 3 7 |
| P2 | 4 0 2 |
| T2 | 8 3 9 |
| P1 | 2 1 2 |
| T3 | 10 4 11 |
| P3 | 4 0 5 |
| T4 | 14 4 16 |
| P5 | 3 1 4 |
|  | 17 5 20 |

存在一个安全序列：P4->P2->P1->P3->P5

此时系统为安全状态，可分配。

（4）

①Request1(0,2,0)≤Need1(3,4,7)

②Request1(0,2,0)≤Available1(2,3,3)

③系统暂时先假定可为P4分配资源，并修改有关数据

|  |  |  |  |
| --- | --- | --- | --- |
|  | Allocation | Need | Available |
| P1 | 2 3 2 | 3 2 7 | 0 1 2 |
| P2 | 4 0 2 | 1 3 4 |  |
| P3 | 4 0 5 | 0 0 6 |  |
| P4 | 4 0 5 | 0 2 0 |  |
| P5 | 3 1 4 | 1 1 0 |  |

④进行安全检查：

可用资源Available1(0,1,2)已不能满足任何进程的需要，系统进入不安全状态，此时系统不分配资源。