操作系统作业

死锁可以认为是两个线程或进程同时在请求对方占有的资源，饥饿可以认为是一个线程或是进程在无限的等待另外两个或多个线程或进程占有的但是不会往外释放的资源。

我的举例

饥饿：三人中两人一直通话，而第三个人一直等待接通。

死锁：两人争抢某物，僵持不下，并且互不让步。

初始系统状态是

|  |  |  |
| --- | --- | --- |
| 5 | 5 | 9 |
| 5 | 3 | 6 |
| 4 | 0 | 11 |
| 4 | 2 | 5 |
| 4 | 2 | 4 |

最大需求资源量

已分配的资源

|  |  |  |
| --- | --- | --- |
| 2 | 1 | 2 |
| 4 | 0 | 2 |
| 4 | 0 | 5 |
| 2 | 0 | 4 |
| 3 | 1 | 4 |

还需分配的资源量

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 7 |
| 1 | 3 | 4 |
| 0 | 0 | 6 |
| 2 | 2 | 1 |
| 1 | 1 | 0 |

剩余可分配的资源 （ 2 3 3 ）

1. P5运行后

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 7 |
| 1 | 3 | 4 |
| 0 | 0 | 6 |
| 2 | 2 | 1 |
| 0 | 0 | 0 |

P4运行后

|  |  |  |
| --- | --- | --- |
| 3 | 4 | 7 |
| 1 | 3 | 4 |
| 0 | 0 | 6 |
| 0 | 0 | 0 |
| 0 | 0 | 0 |

依此推算 得到序列 p5 p4 p3 p2 p1.

2. p2 申请的资源大于了可用的资源，因此不能分配。

3. 对p4 的申请进行分配，仍可以有安全序列 p4 p5 p3 p2 p1 。因此可以给p4分配。

4.若对p1 申请的资源进行分配

|  |  |  |
| --- | --- | --- |
| 3 | 2 | 7 |
| 1 | 3 | 4 |
| 0 | 0 | 6 |
| 0 | 2 | 0 |
| 1 | 1 | 0 |

此时剩余可分配的资源 （0 1 2）

不大于等于p1 到 p5 的任何一个还需要分配的资源的向量，会进入不安全状态，因此不给予p1 分配。