

概念部分：

死锁定义：死锁是指两个或两个以上的进程在执行过程中，由于竞争资源或者由于彼此通信而造成的一种阻塞的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

举例：两家工厂都拥有智能手机的某些专利，但无法合作的情况下，都无法生产手机，最后工厂都倒闭了。

饥饿定义：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。

亦有定义为：由于别的并发的激活的过程持久占有所需资源,使某个异步过程载客预测的时间内不能被激活。

举例：在荒岛上给了一个人 a 鱼竿，一直在使用，另一个人 b 没有工具，持续一个月 ,最后 b 饿死了。

习题：

(一)

初始为 17 5 20 分配后剩余 2 3 3

序号 最大需求 已分配 需分配 分配后可用 次序

P1	5 5 9	2 1 2	3 4 7	17 5 20	5
P2	5 3 6	4 0 2	1 3 4	13 4 14	3
P3	4 0 11	4 0 5	0 0 6	9 4 12	2
P4	4 2 5	2 0 4	2 2 1	15 4 18	4
P5	4 2 4	3 1 4	1 1 0	5 4 7	1

由上表可知 安全 且次序为:P5-P3-P2-P4-P1

(二)

因为 P2Request (0 3 4) 其中 c 资源大于所剩余 (2 3 3)

所以不可以分配

(三)

P4:Request (2 0 1) < (2 3 3) 所以有下表 此时剩余 (0 3 2)

序号 最大需求 已分配 需分配 分配后可用 次序

P1	5 5 9	2 1 2	3 4 7	17 5 20	5
P2	5 3 6	4 0 2	1 3 4	15 4 18	4
P3	4 0 11	4 0 5	0 0 6	8 3 12	2
P4	4 2 5	4 0 5	0 2 0	4 3 7	1
P5	4 2 4	3 1 4	1 1 0	11 4 16	3

因为可以安全进行所以可以分配且

次序为：P4-P3-P5-P2-P1

(四)

在三的基础上 P1Request (0 2 0) < (0 3 2)

所以有下表：此时剩余 (0 1 2)

序号 最大需求 已分配 需分配 分配后可用 次序

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		

因为剩余 ABC 资源为 (0 1 2) P1-5 均无法分配成功 所以不可以将

资源 (0 2 2) 分配给 P1.

Over