**死锁：**是指两个或两个以上的进程（或线程）在执行过程中，因争夺资源而造成的一种互相等待的现象，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。

**饥饿**(starvation)  
　　指的是等待时间已经影响到进程运行，此时成为饥饿现象。如果等待时间过长，导致进程使命已经没有意义时，称之为“饿死”。

我觉得可以这么理解死锁和饥饿的区别，首先死锁是同步的，饥饿时异步的。也就是说，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。

　　介绍“死锁”的例子1：如果线程A锁住了记录R1并等待记录R2，而线程B锁住了记录R2并等待记录R1，这样两个线程A和B就发生了死锁现象。

　　介绍“死锁”的例子2：两个山羊过一个独木桥，两只羊同时走到桥中间，一个山羊等另一个山羊过去了然后再过桥，另一个山羊等这一个山羊过去，结果两只山羊都堵在中间动弹不得。

　　介绍饥饿的例子：资源在其中两个或以上线程或进程相互使用，第三方线程或进程始终得不到。想像一下三个人传球，其中两个人传来传去，第三个人始终得不到。

　　出现以下四种情况会产生死锁：

　　1，相互排斥。一个线程或进程永远占有共享资源，比如，独占该资源。

　　2，循环等待。例如，进程A在等待进程B，进程B在等待进程C，而进程C又在等待进程A。

　　3，部分分配。资源被部分分配，例如，进程A和B都需要访问一个文件，同时需要用到打印机，进程A得到了这个文件资源，进程B得到了打印机资源，但两个进程都不能获得全部的资源了。

　　4，缺少优先权。一个进程获得了该资源但是一直不释放该资源，即使该进程处于阻塞状态。

　　按照前面的分析，情况1和情况4是线程或进程无休止的等待另外的几个线程或进程所占据的资源。

针对上面可能出现死锁的几种情况，可以给出预防措施，比如控制线程或进程如果判断不能够获取到所需的全部的资源，那么就释放已经占有的资源等等。







