1. 死锁
2. 概述

死锁是一种发生在一组相互竞争或同步的进程之间的现象，主要是由于多个进程对资源需求的冲突引起的，当某个进程提出资源的使用请求后，使得系统中一些进程处于无休止的阻塞状态。

1. 资源
2. 独占资源

在任一时刻只能允许一个进程占有的资源。又分为可剥夺式资源和不可剥夺式资源，CPU就是可剥夺式资源，当优先级高的进程发出请求，能够抢到CPU；而像打印机这种设备一旦占用，就不可以停止。死锁与不可剥夺资源有关。

1. 永久性资源

永远存在，可以重复使用，即独占资源。

1. 临时性资源

可消耗的资源。

1. 产生死锁的四个条件
2. 互斥使用

资源是互斥的，其他申请资源的进程必须等待。

1. 非剥夺控制

占用资源的进程不会被其他进程强迫释放资源。

1. 零散请求

进程可以按照需要逐次申请资源，而不是集中性地一次请求所有资源。这样，进程在已占有资源的情况下，又申请其他资源而得不到满足的情况下，并不会释放已占有的资源。

1. 循环等待

等待资源的进程形成了一个闭环，环中的进程都在等待下一个进程占有的资源，无止境地等待。

以上四个条件是产生死锁的必要条件，非充分条件，因此只要破坏其中之一，就可以预防死锁的产生。

1. 死锁的预防
2. 破坏互斥条件

如果资源允许共享，那么死锁一定不会发生，通过SPOOLING技术允许若干个进程同时使用资源，不适合大部分资源。

1. 破坏不可剥夺条件

一种方法是如果一个已占有资源的进程再次申请资源得不到满足时，必须先放弃已占有的资源；

另一种方法是如果申请资源的进程的优先级比占有该资源的进程优先级更高，则它可以强迫占有资源的进程放弃使用。

这些方法实现困难，需要保存进程放弃资源的现场以便再次恢复，需要耗费很多时间和空间，只适用于CPU和存储器。

1. 破坏零散请求条件

采用静态分配策略，当一个进程得到了它所需要的所有资源后才能够执行，资源利用率很低。

1. 破坏循环等待条件

按序分配资源，系统依据一定的策略给资源编号，如资源特性，使用频率等，进程必须按照从小到大的顺序申请资源，并规定进程占有的资源号小于申请到的资源号时才能够得到申请资源。

1. 死锁的避免
2. 概述

死锁的预防策略对资源的申请加以限制，降低了系统的效率和资源的利用率。

死锁的避免策略是用动态的方法判断资源的使用情况和系统的状态，在分配资源之前，系统将判断假若满足进程的要求是否会发生死锁，如果会，资源就不分配，从而避免死锁的发生。

1. 系统状态

系统的状态分为安全状态和不安全状态，所谓安全状态，指的是多个进程动态地申请资源时，系统将按照某种顺序逐次地为每个进程分配资源，使每个进程都可以在最终得到最大需求后，依次顺利地完成。如果不存在一种分配顺序使得进程都能够顺序完成，则系统处于不安全状态。

不安全状态不一定发生死锁，但是死锁一定属于不安全状态，因此避免死锁的关键就是让系统在动态分配资源的过程中，不要进入不安全状态。

1. 银行家算法

资源分配时避免死锁的算法。即检查每一种资源分配方案，能够满足所有的请求并且不会进入不安全状态，只要有一种安全状态方案即可。

1. 死锁的解除
2. 重新启动，粗暴方法；
3. 撤销进程，可以撤销优先级低，占用资源少的进程，尽量减少系统的损失；
4. 剥夺资源，可以只剥夺死锁进程的资源，直到解除死锁；
5. 进程回退，回退到非死锁状态；’