**死锁**：多个进程竞争共享资源或者由于彼此通信产生的一种阻塞的现象。

产生死锁的条件：

死锁产生的4个必要条件

    1、互斥：某种资源一次只允许一个进程访问，即该资源一旦分配给某个进程，其他进程就不能再访问，直到该进程访问结束。

    2、占有且等待：一个进程本身占有资源（一种或多种），同时还有资源未得到满足，正在等待其他进程释放该资源。

    3、不可抢占：别人已经占有了某项资源，你不能因为自己也需要该资源，就去把别人的资源抢过来。

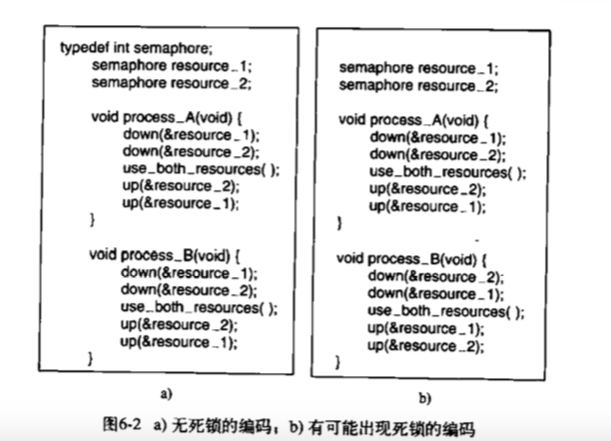
    4、循环等待：存在一个进程链，使得每个进程都占有下一个进程所需的至少一种资源。

       当以上四个条件均满足，必然会造成死锁，发生死锁的进程无法进行下去，它们所持有的资源也无法释放。这样会导致CPU的吞吐量下降。所以死锁情况是会浪费系统资源和影响计算机的使用性能的。那么，解决死锁问题就是相当有必要的了。

资源：分为硬件资源（打印机，磁盘等）和一组信息（数据库加锁的记录）

可抢占资源和不可抢占资源

资源所需要的事件顺序：请求资源， 使用资源， 释放资源

**资源死锁：**

进程集合中的进程等待死锁的进程所拥用的资源，由于进程不能运行因此不能释放资源

**资源死锁的条件**：

1）互斥条件，一个资源要么是分配给一个进程，要么是可用的

2）占有和等待条件，已经得到的某个资源可以申请新的资源

3）不可抢占条件，已经分配的资源不可以被强制性抢占，只能由占用它的进程显示释放

4）环路等待条件，死锁发生时，系统中一定有两个或者两个以上的进程组成的一条环路，该环路中的每个进程都在等待下一个进程所占用的资源

**四种处理死锁的策略：**

1）忽略该问题

2）检测死锁并且恢复

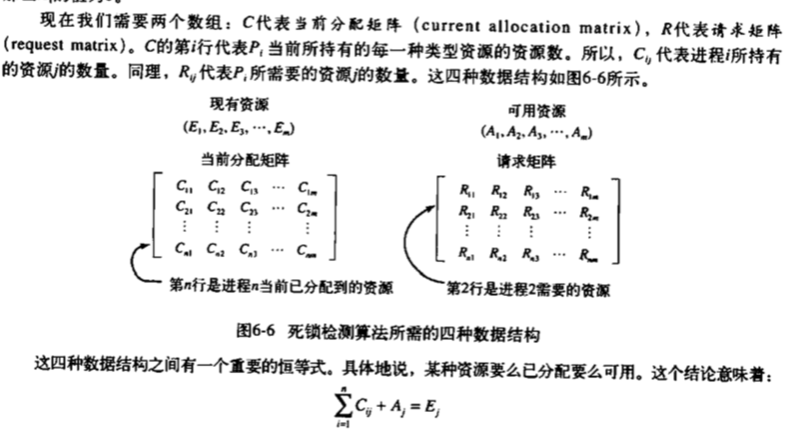
3）仔细对资源进行分配，动态的避免死锁

4）通过破坏死锁的四个必要条件之一，防止死锁的产生

**（1）鸵鸟算法**

将头埋入沙子不去理会外面的情况（死锁）

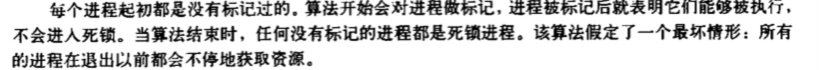
**（2）死锁检测和死锁恢复**



不去理会死锁的产生，但检测死锁发生时，采取恢复措施

1）每种类型一个资源的死锁检测

环路检测算法

2）每种类型多个资源的死锁检测

1）寻找一个没有标记的进程P， 对于它而言R矩阵的第i行 《= A

2）如果找到了这样的一个进程，那么C矩阵的第i行向量加到A中，标 记该进程，并转到第1步

3）如果没有这样的进程，那么算法终止



**3.从死锁中恢复**

1）利用抢占式恢复

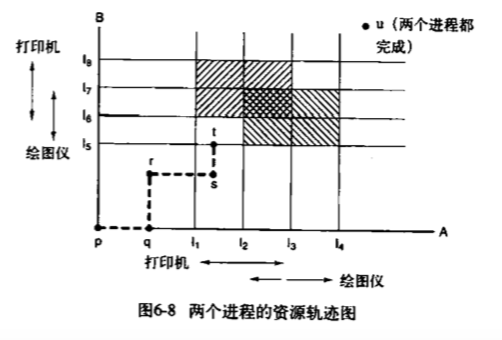
2）回滚机制

3）kill进程恢复

**死锁避免**

方案一：

资源轨迹图

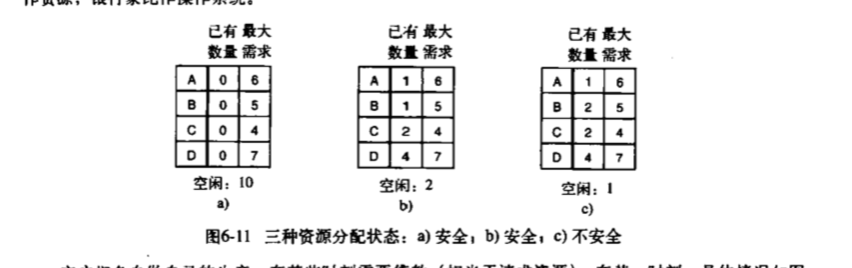


**安全状态和不安全状态：**

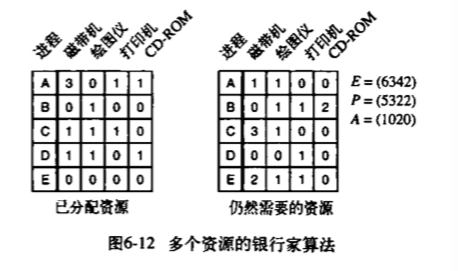
安全状态：即使每个进程突然请求最大资源的最大需求，也仍然存在某种调度次序能够使得每一个进程运行完毕

区别：安全状态保证系统能够完成所有进程，不安全则没有

**单个资源的银行家算法：**



**多个资源的银行家算法：**

****

**6.6死锁防御**

死锁避免是不可能的，因为它需要获取未来的请求，而这写请求是不可能的

**1.破坏互斥条件**

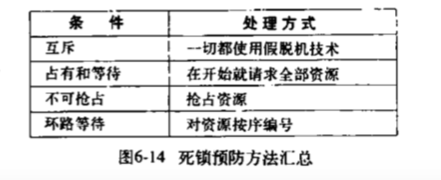
1）避免分配不是必须需要的资源。尽量做到尽可能少的进程可以真正的请求资源。

**2.破坏占有和等待条件**

规定所有进程在开始前请求所需的全部资源，如果所有资源都可以使用，那么就分配给这个进程 ，如果有一个或者多个资源正在使用，那么不分配，进程等待

**3.破坏不可抢占条件**

4.破坏环路等待条件



**通信死锁：**

一个进程等待另外一个进程引发的事件而产生的阻塞

网络资源由网络的各种协议栈解决