https://mp.weixin.qq.com/s/s0X8gpnPgk-2Z\_fVUfWwUw

**一文详解，死锁与解决方案（附源码）**

#### ****什么是死锁？****

“死锁”就是两个或两个以上的线程在执行过程中，互相持有对方所需要的资源，导致这些线程处于等待状态，无法继续执行。若无外力作用，它们都将无法继续执行下去，就进入了“永久”阻塞的状态。

#### ****死锁产生的原因****

死锁的发生也必须具备一定的条件，必须具备以下四个条件：

* 互斥，共享资源 X 和 Y 只能被一个线程占用；
* 占有且等待，线程01 已经取得共享资源 X，在等待共享资源 Y 的时候，不释放共享资源 X；
* 不可抢占，其他线程不能强行抢占线程01 占有的资源；
* 循环等待，线程01 等待线程02 占有的资源，线程02 等待线程01 占有的资源，就是循环等待。

#### ****如何避免死锁？****

死锁一旦发生，并没有什么好的方法解决，通常我们只能避免死锁的发生。  
  
怎么避免呢？那就要看针对死锁发生的原因去解决。

1. 首先，“互斥”是没有办法避免的，你想从账户A转账到账户B，就必须加锁，就没法避免互斥的存在。
2. 对于“占用且等待”这个条件，我们可以一次性申请所有的资源，这样就不存在等待了。
3. 对于“不可抢占”这个条件，占用部分资源的线程进一步申请其他资源时，如果申请不到，可以在一定时间后，主动释放它占有的资源，这样就解决了不可抢占这个条件。
4. 对于“循环等待”，我们可以靠按“次序”申请资源来预防。所谓按序申请，就是给资源设定顺序，申请的时候可以先申请序号小的资源，再申请序号大的，这样资源线性化后，自然就不存在循环等待了。

**所以，总结来看，避免死锁的发生有三种方法：**

**破坏占用且等待的条件、**

**破坏不可抢占条件、**

**破坏循环等待条件。**