# 锁机制

在多线程编程中，操作系统引入了锁机制。通过锁机制，能够**保证在多核多线程环境中，在某一个时间点上，只能有一个线程进入临界区代码，从而保证临界区中操作数据的一致性**。

## 死锁

是指**两个或两个以上的进程在执行过程中,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。

## 产生死锁的原因

1. 因为**系统资源不足**。
2. 进程运行**推进的顺序不当**。
3. **资源分配不当**等。

如果系统资源充足，进程的资源请求都能够得到满足，死锁出现的可能性就很低，否则

就会因**争夺有限的资源而陷入死锁**。其次，**进程运行推进顺序与速度不同**，也可能产生死锁。

## 死锁的四个必要条件

1. **互斥条件**：一个资源每次只能被一个进程使用。

2. **请求与保持条件**：一个进程因请求资源而阻塞时，对已获得的资源保持不放。

3. **不剥夺条件**:进程已获得的资源，在末使用完之前，不能强行剥夺。

4. **循环等待条件**:若干进程之间形成一种头尾相接的循环等待资源关系。

这四个条件是死锁的必要条件，只要系统发生死锁，这些条件必然成立，而只要上述条件之

一不满足，就不会发生死锁。

## 死锁的解除与预防

<http://blog.csdn.net/abigale1011/article/details/6450845/>

理解了死锁的原因，尤其是产生死锁的四个必要条件，就可以最大可能地避免、预防和解除死锁。所以，在系统设计、进程调度等方面**注意如何不让这四个必要条件成立**，如何**确定资源的合理分配算法**，避免进程永久占据系统资源。此外，也要防止进程在处于等待状态

的情况下占用资源。因此，对资源的分配要给予合理的规划。

# 线程调度