

本节内容

# 死锁的概念

## 知识总览



### 死锁的概念

什么是死锁

进程死锁、饥饿、死循环的区别

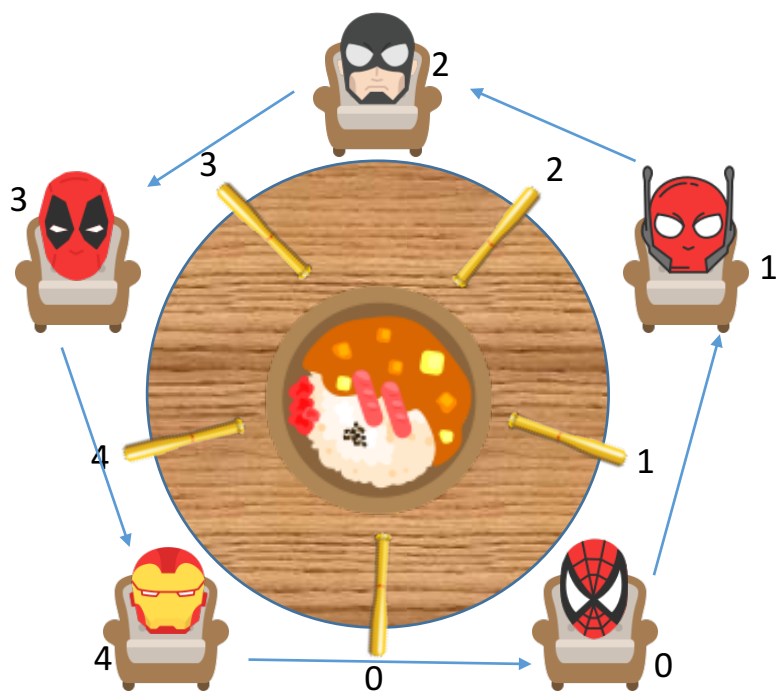
死锁产生的必要条件

什么时候会发生死锁

死锁的处理策略

## 什么是死锁

哲学家进餐问题中，如果5位哲学家进程并发执行，都拿起了左手边的筷子...



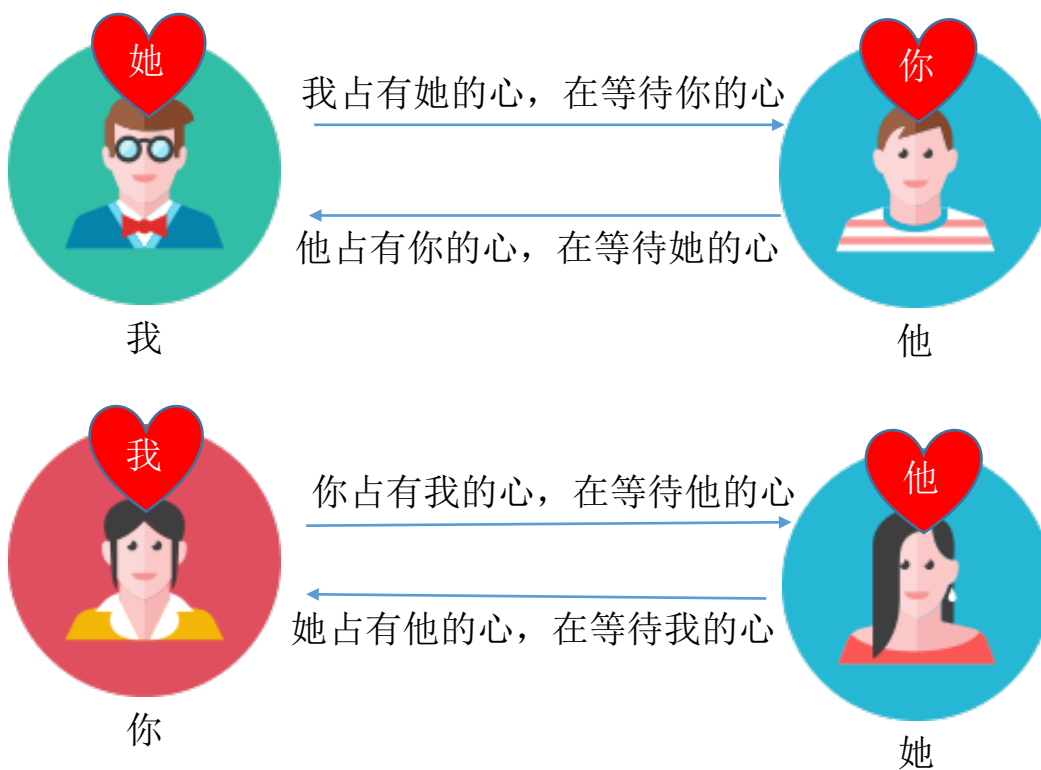
```
semaphore chopstick[5]={1,1,1,1,1};
Pi () {           //i号哲学家的进程
    while(1) {
        P(chopstick[i]);           //拿左
        P(chopstick[(i+1)%5]);     //拿右
        吃饭...
        V(chopstick[i]);           //放左
        V(chopstick[(i+1)%5]);     //放右
        思考...
    }
}
```

每位哲学家都在等待自己右边的人放下筷子，这些哲学家进程都因等待筷子资源而被阻塞。即发生“死锁”

保持一个，请求另一个

## 什么是死锁

有一首歌的歌词：我爱你，你爱他，他爱她，她爱我.....这世界每个人都爱别人.....  
我们从资源占有的角度来分析，这段关系为什么看起来那么纠结...



每个人都占有一个资源，  
同时又在等待另一个人手  
里的资源。发生“死锁”

在并发环境下，各进程因竞争资源而造成的一种互相等待对方手里的资源，导致各进程都阻塞，都无法向前推进的现象，就是“死锁”。  
发生死锁后若无外力干涉，  
这些进程都将无法向前推进。

## 死锁、饥饿、死循环的区别

P144

死锁：各进程互相等待对方手里的资源，导致各进程都阻塞，无法向前推进的现象。

饥饿：由于长期得不到想要的资源，某进程无法向前推进的现象。比如：在短进程优先（SPF）算法中，若有源源不断的短进程到来，则长进程将一直得不到处理机，从而发生长进程“饥饿”。

死循环：某进程执行过程中一直跳不出某个循环的现象。有时是因为程序逻辑 bug 导致的，有时是程序员故意设计的。

	共同点	区别
死锁	都是进程无法顺利向前推进的现象（故意设计的死循环除外）	死锁一定是“循环等待对方手里的资源”导致的，因此如果有死锁现象，那至少有 <u>两个或两个以上的进程同时发生死锁</u> 。另外，发生死锁的进程一定处于 <u>阻塞态</u> 。
饥饿		可能只有 <u>一个进程发生饥饿</u> 。发生饥饿的进程既可能是阻塞态（如长期得不到的I/O设备），也可能是就绪态（长期得不到处理机） <small>→ 阻塞 如优先级调度中的进程：就绪态</small>
死循环		可能只有一个进程发生死循环。死循环的进程可以上处理机运行（可以是运行态），只不过无法像期待的那样顺利推进。死锁和饥饿问题是由于操作系统分配资源的策略不合理导致的，而死循环是由代码逻辑的错误导致的。 <u>死锁和饥饿是管理者（操作系统）的问题，死循环是被管理者的问题。</u>

## 死锁产生的必要条件

产生死锁必须同时满足一下四个条件，只要其中任一条件不成立，死锁就不会发生。

- 1 **互斥条件**：只有对必须互斥使用的资源的争抢才会导致死锁（如哲学家的筷子、打印机设备）。  
像内存、扬声器这样可以同时让多个进程使用的资源是不会导致死锁的（因为进程不用阻塞等待这种资源）。

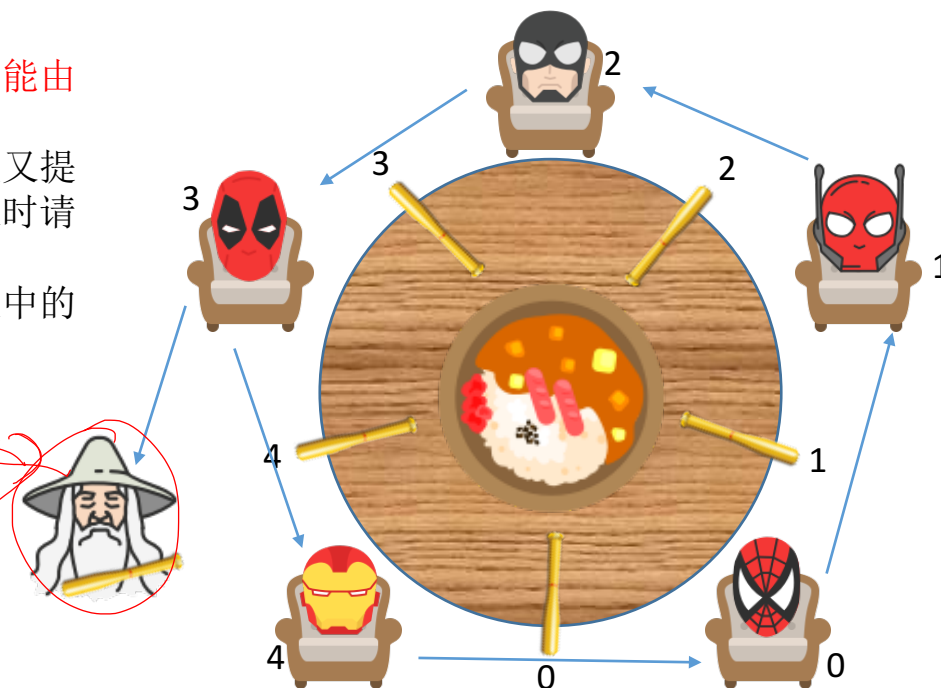
- 2 **不剥夺条件**：进程所获得的资源在未使用完之前，不能由其他进程强行夺走，只能主动释放。

- 3 **请求和保持条件**：进程已经保持了至少一个资源，但又提出了新的资源请求，而该资源又被其他进程占有，此时请求进程被阻塞，但又对自己已有的资源保持不放。

- 4 **循环等待条件**：存在一种进程资源的循环等待链，链中的每一个进程已获得的资源同时被下一个进程所请求。

注意！发生死锁时一定有循环等待，但是发生循环等待时未必死锁（循环等待是死锁的必要不充分条件）

如果同类资源数大于1，则即使有循环等待，也未必发生死锁。但如果系统中每类资源都只有一个，那循环等待就是死锁的充分必要条件了。



资源数  $m$  进程最大需求  $n(k-1)+1$  不会发生死锁

## 什么时候会发生死锁

- 空间上 → 1. 对系统资源的竞争。各进程对不可剥夺的资源（如打印机）的竞争可能引起死锁，对可剥夺的资源（CPU）的竞争是不会引起死锁的。
- 时间上 → 2. 进程推进顺序非法。请求和释放资源的顺序不当，也同样会导致死锁。例如，并发执行的进程 P1、P2 分别申请并占有了资源 R1、R2，之后进程 P1 又紧接着申请资源 R2，而进程 P2 又申请资源 R1，两者会因为申请的资源被对方占有而阻塞，从而发生死锁。
3. 信号量的使用不当也会造成死锁。如生产者-消费者问题中，如果实现互斥的 P 操作在实现同步的 P 操作之前，就有可能导致死锁。（可以把互斥信号量、同步信号量也看做是一种抽象的系统资源）

死锁成因 1、2 主要

总之，对不可剥夺资源的不合理分配，可能导致死锁。

## 死锁的处理策略



1. 预防死锁。破坏死锁产生的四个必要条件中的一个或几个。
2. 避免死锁。用某种方法防止系统进入不安全状态，从而避免死锁（银行家算法）
3. 死锁的检测和解除。允许死锁的发生，不过操作系统会负责检测出死锁的发生，然后采取某种措施解除死锁。



## 知识回顾与重要考点

### 死锁的概念

什么是死锁 ⊖ 各进程互相等待对方手里的资源，导致各进程都阻塞，无法向前推进

死锁、饥饿、死循环的区别 ⊖

死锁：至少是两个进程一起死锁，死锁进程处于阻塞态

饥饿：可以只有一个进程饥饿，饥饿进程可能阻塞也可能就绪

死循环：可能只有一个进程发生死循环，死循环的进程可上处理机

死锁和饥饿是操作系统要解决的问题，死循环是应用程序员要解决的

死锁产生的必要条件 ⊖

互斥条件 ⊖ 对必须互斥使用的资源的争抢才会导致死锁

不剥夺条件 ⊖ 进程保持的资源只能主动释放，不可强行剥夺

请求和保持条件 ⊖ 保持着某些资源不放的同时，请求别的资源

循环等待条件 ⊖  
存在一种进程资源的循环等待链  
循环等待未必死锁，死锁一定有循环等待

什么时候会发生死锁 ⊖ 对不可剥夺资源的不合理分配，可能导致死锁

死锁的处理策略 ⊖

预防死锁 ⊖ 破坏死锁产生的四个必要条件

避免死锁 ⊖ 避免系统进入不安全状态（银行家算法）

死锁的检测和解除 ⊖ 允许死锁发生，系统负责检测出死锁并解除

的  
并发性  
高