

# 第 5 章 死锁

月出皓兮 苏曙光老师的课堂笔记

2022 年 4 月 10 日

## 目录

|                                      |          |
|--------------------------------------|----------|
| <b>1 死锁</b>                          | <b>2</b> |
| 1.1 定义 . . . . .                     | 2        |
| 1.2 起因 . . . . .                     | 2        |
| 1.2.1 系统资源有限 . . . . .               | 2        |
| 1.2.2 并发进程的推进顺序不当 . . . . .          | 2        |
| 1.2.3 相关结论 . . . . .                 | 3        |
| 1.3 死锁产生的必要条件 . . . . .              | 3        |
| 1.3.1 互斥条件 . . . . .                 | 3        |
| 1.3.2 不剥夺条件 . . . . .                | 3        |
| 1.3.3 部分分配条件 . . . . .               | 3        |
| 1.3.4 环路条件 . . . . .                 | 4        |
| 1.4 解决死锁的策略 . . . . .                | 4        |
| 1.5 预防死锁 . . . . .                   | 4        |
| 1.5.1 破坏部分分配条件——预先静态分配 . . . . .     | 4        |
| 1.5.2 破坏环路条件——有序资源分配 . . . . .       | 4        |
| 1.6 避免死锁 . . . . .                   | 4        |
| 1.7 检测与恢复死锁 . . . . .                | 5        |
| 1.8 Windows 和 Linux 对死锁的策略 . . . . . | 5        |

# 1 死锁

## 1.1 定义

两个或多个进程无限期地等待永远不会发生的条件的一种系统状态。

在两个或多个进程中，每个进程都持有某种资源，但又继续申请其它进程已持有的某种资源。此时每个进程都拥有其运行所需的一部分资源，但是又都不够，从而每个进程都不能向前推进，陷于阻塞状态。这种状态称死锁。

其中资源的概念为：竞争使用，数量有限，共享

可抢占资源：可以被多个进程同时访问，或被一个进程占用后，在该进程使用完之前，可以被其他进程抢占，但是并不影响相关进程的运行结果。例如：CPU 和内存

不可抢占资源：被一个进程占用后，除非该进程已使用完并释放了它，其他进程不能强行抢占该资源。例如：大多数硬件资源和软件资源。

## 1.2 起因

### 1.2.1 系统资源有限

资源数目不足以满足所有进程的需要，引起进程对资源的竞争而产生死锁。

### 1.2.2 并发进程的推进顺序不当

进程在运行过程中，请求与释放资源的顺序不当，导致进程产生死锁。

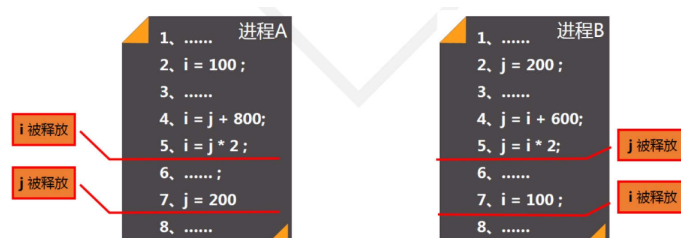


图 1: 并发进程的推进顺序不当造成死锁例 1

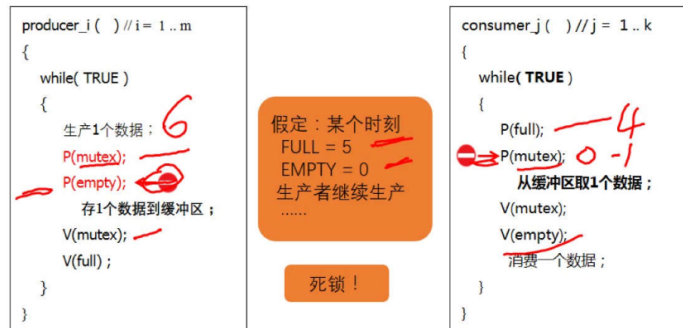


图 2: 并发进程的推进顺序不当造成死锁例 2

### 1.2.3 相关结论

参与死锁的进程至少是 2 个，即两个或以上进程才会出现死锁。

参与死锁的进程至少有 2 个已经占有资源。

参与死锁的所有进程都在等待资源。

参与死锁的进程是当前系统中所有进程的子集。

死锁会浪费大量系统资源，甚至导致系统崩溃。

## 1.3 死锁产生的必要条件

### 1.3.1 互斥条件

进程互斥使用资源，资源具有独占性。

### 1.3.2 不剥夺条件

进程在访问完资源前不能被其他进程强行剥夺。

### 1.3.3 部分分配条件

进程边运行边申请资源，临时需要临时分配

区别于 全部提前将资源分配好（规定好了运行顺序）。

### 1.3.4 环路条件

多个进程构成环路：环中每个进程已占用的资源被前一进程申请，而自己所申请资源又被环中后一进程占用着。

## 1.4 解决死锁的策略

预防死锁，避免死锁，检测死锁，恢复死锁。

### 1.5 预防死锁

通过设置某些限制条件，破坏死锁四个必要条件中的一个或多个，来防止死锁。

破坏不剥夺条件——代价大

较易实现，（早期）广泛使用。但缺点是由于限制太严格，导致资源利用率和吞吐量降低。

#### 1.5.1 破坏部分分配条件——预先静态分配

进程运行前将所需资源一次性全部分配给它。

特点：

1. 执行可能被延迟☐所需资源不能全部满足时。
2. 应用开销增大：运行前估算资源需求。
3. 资源利用率低☐资源被占而不里、改进。

改进方式为：资源分配的单位由进程改为程序步

#### 1.5.2 破坏环路条件——有序资源分配

目的☐破坏环路条件，使得环路无法构成。

策略：系统中的每个资源分配有一个唯一序号；**进程每次申请资源时只能申请序号更大的资源**。如果进程已占有资源的序号最大为 M，则下次只能申请序号大于 M 的资源，而不能再申请序号小于或等于 M 的资源。

### 1.6 避免死锁

在资源分配过程中，用某种方法去评估若分配资源是否会让系统进入死锁状态，若是，则拒绝此次分配资源，从而避免死锁的发生。

银行家算法(不做要求)[https://blog.csdn.net/qq\\_36260974/article/details/84404369](https://blog.csdn.net/qq_36260974/article/details/84404369)

### 1.7 检测与恢复死锁

允许死锁发生,但可通过检测机制及时检测出死锁状态,并精确确定与死锁有关的进程和资源,然后采取适当措施,将系统中已发生的死锁清除,将进程从死锁状态解脱出来。

但检测方法复杂,实现难度大。恢复方法靠人工撤消一些进程,回收资源再分配。

### 1.8 Windows 和 Linux 对死锁的策略

鸵鸟策略。

需要程序员和用户避免产生死锁,而死锁问题可通过**重启系统**的方法一次性解决。