

死锁

概念

多个进程因竞争资源而造成的一种僵局（相互等待）
若无外力作用，这些进程都将无法向前推进

死锁产生的原因

- 系统资源的竞争
- 进程推进顺序非法

★死锁产生的必要条件

- 互斥条件：进程要求对所分配的资源进行排他性控制
- 不剥夺条件
- 请求保持条件：进程至少保持了一个资源，但又提出了新的资源请求
- 循环等待条件：存在一种进程资源循环等待链，链中每个进程已获得的资源同时被链中下一个进程所请求。

死锁

处理策略

死锁预防

设置限制条件，破坏死锁产生的4个必要条件之一

互斥条件一般无法破坏

摒弃请求保持条件：规定所有进程都必须一次性申请其在运行过程中所需的全部资源。简单易行，但可能会造成资源严重浪费

摒弃不剥夺条件：规定一个已经保持资源的进程在提出新的请求时若不能立即满足则释放其所有资源

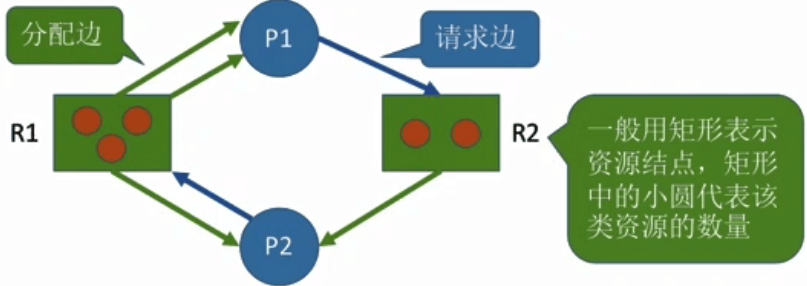
摒弃环路等待条件：将系统中的资源按类型赋予不同的序号并规定所有的进程必须严格按照序号递增的顺序申请资源

避免死锁

在资源的动态分配过程中，用某种方法防止系统进入不安全状态

系统安全状态

银行家算法



检测死锁的算法：

- 1) 在资源分配图中，找出既不阻塞又不是孤点的进程 P_i （即找出一条有向边与它相连，且该有向边对应资源的申请数量小于等于系统中已有空闲资源数量。如下图中，R1没有空闲资源，R2有一个空闲资源。若所有的连接该进程的边均满足上述条件，则这个进程能继续运行直至完成，然后释放它所占有的所有资源）。消去它所有的请求边和分配变，使之称为孤立的结点。在下图中，P1是满足这一条件的进程结点，于是将P1的所有边消去。
- 2) 进程 P_i 所释放的资源，可以唤醒某些因等待这些资源而阻塞的进程，原来的阻塞进程可能变为非阻塞进程。在下图中，P2就满足这样的条件。根据1)中的方法进行一系列简化后，若能消去途中所有的边，则称该图是**可完全简化的**。

资源分配图

死锁定理

S为死锁的条件是当且仅当S状态的资源分配图是不可完全简化的，该条件为死锁定理

死锁解除

- 资源剥夺法
- 撤销进程法
- 进程回退法