

2.4 死锁

死锁的概念

- 死锁的定义
 - 多个进程因为竞争资源造成的一种僵局，没有外力作用，这些进程都无法向前继续推进
- 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：进程对分配的资源进行排他性控制
 - 不可剥夺条件：进程获得资源在未使用完之前，不能被其他进程强行夺走
 - 请求并保持条件：进程已经保持了至少一个资源，提出新的资源请求，而该资源已经被其他进程占有，此时该进程被阻塞，但是自己已经获得的资源保持不放
 - 循环等待条件：你等我释放 我等你释放

死锁的处理策略

- 死锁预防
 - 破坏四个必要条件中的一个或几个，防止死锁
 - 资源分配保守，宁可资源闲置
 - 一次性请求所有资源，资源剥夺，资源按序分配
 - 优点：适用于突发式处理的进程，不必进行剥夺
 - 缺点：效率低，进程初始化时间长，剥夺次数过多，不变灵活申请新资源
- 避免死锁
 - 在资源的动态分配中，用某种方法防止系统进入不安全状态，避免死锁
 - 运行过程中预测分配资源是否会死锁
 - 寻找可能的安全序列
 - 优点：不必进行剥夺
 - 缺点：必须知道将来的资源需求，进程不能被长时间阻塞
- 死锁的检测及解除
 - 允许进程死锁，通过检测及时的判断死锁，然后对其进行解除
 - 宽松，只要允许就分配资源
 - 定期检查是否死锁
 - 优点：不延长初始化时间，允许对死锁进行现场处理
 - 缺点：通过剥夺解除死锁，造成损失

死锁预防

- 破坏互斥条件：某些资源只能被互斥访问，并且某些情况下必须保护互斥性
- 破坏不剥夺条件
 - 释放已经占有的资源
 - 特点：增加系统开销 实现复杂 降低吞吐量
 - 用于状态易于保存和恢复的数据（CPU的寄存器及内存资源）
- 破坏请求并保持条件
 - 一次性申请完所需要的全部资源
 - 特点：实现简单，但是资源被严重浪费，甚至可能导致进程饥饿
- 破坏循环等待条件
 - 采用顺序资源法，对进程进行顺序推荐
 - 特点：进程编号必须稳定，可能会导致资源浪费，并且不利于用户编程

死锁避免

- 系统安全状态
 - 按照某种方式分配资源后，是否会导致死锁，如果会导致死锁，那么就是不安全状态，反之就是安全状态
- 银行家算法
 - 思想：通过计算当前资源的不同分配方式，从而预测系统是否会进入不安全状态
 - 就像是银行贷款，是否会导致银行没有足够的资金对外出借

死锁的检测和解除

- 资源分配图
 - 圆圈表示进程，框表示一类资源，进程到资源的有向边称为请求边，资源到进程的边称为分配边
 - 在资源分配图中找到分配满足的进程，然后消去其请求边与分配边
- 死锁定理
 - 如果最后所有边都可以被消去，那么就是可以简化的，不存在死锁，反之存在死锁
- 死锁解除
 - 资源剥夺法：挂起某些死锁进程，抢占资源，将这些资源分配给其他死锁进程，但是要防止挂起时间过长
 - 撤销进程法：强制撤销部分甚至全部死锁进程，并且剥夺他们的资源，撤销原则可以根据优先级和撤销进程的代价进行
 - 进程回退法：让一个或者多个进程回退到足以回避死锁的地步，进程回退时自愿释放资源而非被剥夺。要求系统保持进程历史信息，设置还原点

死锁、饥饿、死循环的区别

- 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
- 死循环：某进程执行过程中一直跳不出某个循环的现象