7.8 进程同步、死锁

2153401 赵一婷

- ch7 进程同步
- 临界
  - 临界资源：系统中某些资源一次只允许一个进程使用，这样的资源叫做临界资源（互斥资源
  - 临界区：进程中涉及到临界资源的程序段叫临界区
    - 进程A处于自己的临界区中时，仍能被进程B中断
- 临界区使用原则
  - 有空让进
  - 无空等待
  - 多中择一
  - 有限等待
  - 让权等待
- 软件方法
  - 单标志位法
  - 双标志位法：先检查后修改
  - 双标志位法：先修改后检查
  - 唯一正确：先修改后检查，后修改者等待
  - 缺点：忙等待，实现复杂需要较高的编程技巧
- 硬件方法
  - Test-and-set指令
  - Swap指令
  - 开关中断指令
  - 优点
    - 适用于任意数目进程
    - 解决方法简单
    - 可支持内存中有多个临界区
  - 缺点
    - 等待要消耗CPU时间，不能实现”让权等待“
    - 可能会出现饥饿现象
    - 可能会产生死锁
- 进程同步机制
  - 基本概念：OS从进程管理者角度来处理进程间的同步互斥问题的机制
  - 实现方法：信号量PV操作、管程、条件临界域

- 信号量定义：信号量表示资源的实体，是一个与队列有关的整型变量
  - 信号量只能通过初始化和PV原语访问
  - 访问信号量的进程不受进程调度的打断
  - OS用信号量对进程和资源进行管理和控制
  - 信号量的数据结构
  - 物理含义：
    - S>0：表示有S个资源可用
    - S=0：表示无资源可用
    - S<0：表示当前有S个进程在等待队列中等待
  - 公有信号量：实现进程互斥 初值为1 允许其联系的一组进程进行访问
  - 私有信号量：实现进程同步 初值为0 仅允许拥有它的进程访问
- PV机制
  - P原语：意味着申请分配一个单位的资源
    - S>=0：继续运行
    - S<0 :进程被堵塞，插入等待队列
  - V原语：意味着释放一个单位的资源
    - S>0:
    - S<=0:从等待队列的头部唤醒一个进程到就绪队列中（此时进程再等待系统调用进入临界区干活
  - PV机制优缺点
    - 优点：简单、表达能力强，可以解决同步和互斥问题
    - 缺点：
      - 不够安全，使用不当会出现死锁
      - 遇到复杂同步互斥问题时实现复杂
  - 注意点：PV操作必须成对出现
    - 互斥操作：处在同一进程中
    - 同步操作：处在不同进程中
  - 应用
    - 实现进程互斥
    - 实现进程同步
- 典型同步问题
  - 做题注意：1.说明谁是临界资源 2.说明信号量的设置以及初值 3.确定协作进程数与关系
  - 信号量的数量确定，一般与自己要协作的资源数量有关，要协作的关系有关。
  - 生产者消费者
  - 读者写者

        - 读者优先
    - 哲学家就餐
- ch8 死锁
    - 死锁问题
        - 产生原因
            - 一组阻塞进程分别占有一定资源并且等待获取另外一些已经被同组其他进程所占有的资源
            - 本质原因：资源竞争，竞争可能产生死锁，但不一定会死锁（资源的属性决定的
        - 系统模型
            - 进程使用资源的方式：申请、使用、释放
    - 死锁特征（四个同时满足时会死锁
        - 互斥：至少有一个资源处于互斥模式
        - 持有并等待：一个进程必须占有至少一个资源，并等待另一个资源，而该资源由其他进程所占用
        - 非抢占：资源不能被抢占
        - 循环等待：
    - 资源分配图
        - 存在环且每种资源类型只有一个实例，则表示出现死锁（具体情况具体分析
        - P--->R 资源请求线 R---->P资源分配线
    - 死锁处理方法
        - 死锁预防：确保至少一个产生死锁的必要条件不成立
            - 存在问题
                - 互斥：不能通过互斥条线来预防，因为资源本身不共享的性质无法改变
                - 持有并等待：资源利用率较低，可能发生饥饿
        - 死锁避免：通过增加申请资源分配的相关信息来避免
            - 例如：每个进程事先说明自己所需要的每种资源的最大数量
            - 通过该方法可以保证不存在循环等待的条件
        - 死锁检测：允许系统进入死锁状态，需要检测算法和恢复算法
            - 原因/背景：死锁是小概率事件并且影响范围小
        - 死锁恢复：人工处理死锁，让系统从死锁状态中自动恢复
            - 方法一：终止进程法：简单地终止一个多多个进程以打破循环等待
                - 终止进程的策略：
            - 方法二：资源抢占法：从一个或多个死锁进程处抢占一个或多个资源