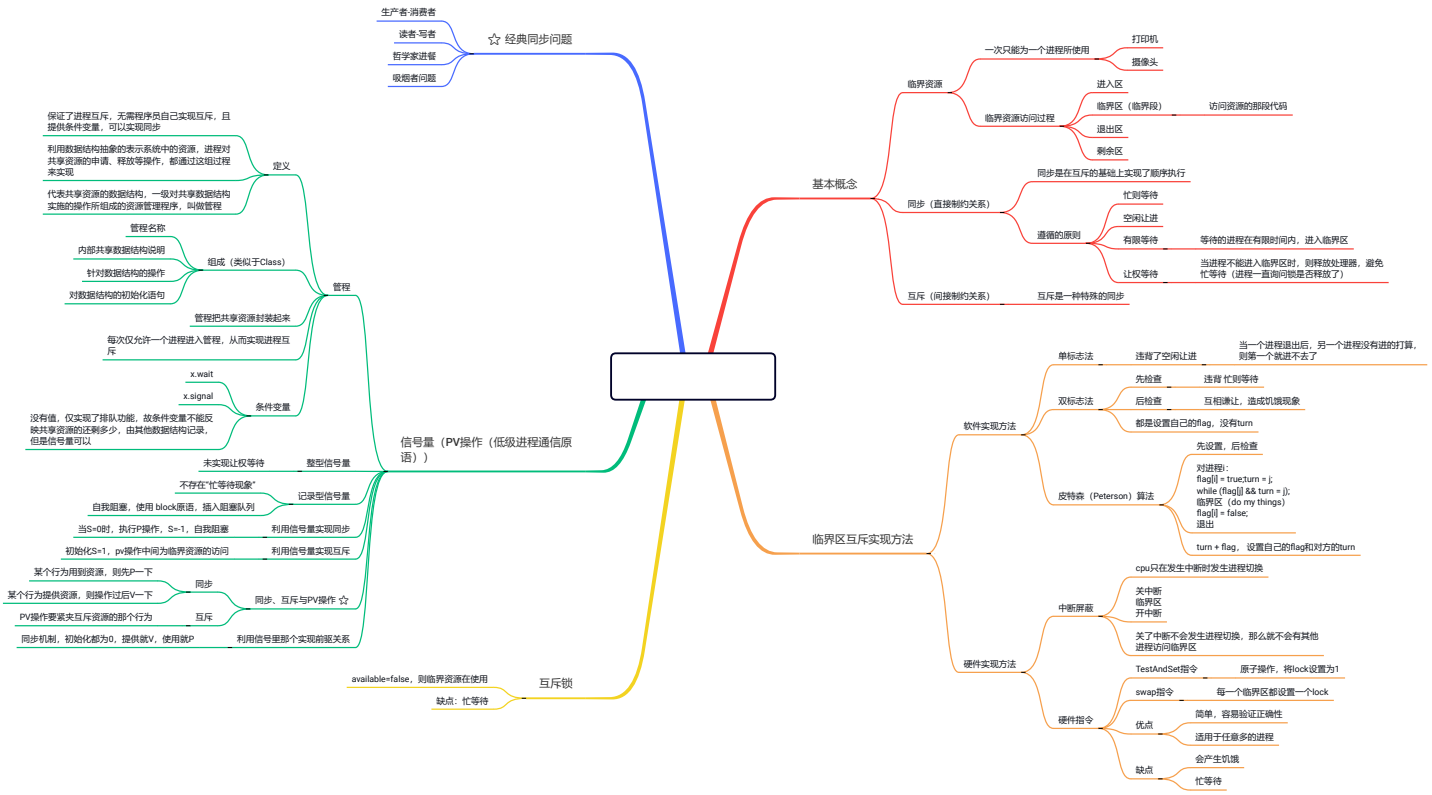
基本概念
临界资源
一次只能为一个进程所使用
打印机
摄像头
临界资源访问过程
进入区
临界区（临界段）
访问资源的那段代码
退出区
剩余区
同步（直接制约关系）
同步是在互斥的基础上实现了顺序执行
遵循的原则
忙则等待
空闲让进
有限等待
等待的进程在有限时间内，进入临界区
让权等待
当进程不能进入临界区时，则释放处理器，避免忙等待（进程一直询问锁是否释放了）
互斥（间接制约关系）
互斥是一种特殊的同步
临界区互斥实现方法
软件实现方法
单标志法
违背了空闲让进
当一个进程退出后，另一个进程没有进的打算，则第一个就进不去了
双标志法
先检查
违背 忙则等待
后检查
互相谦让，造成饥饿现象
都是设置自己的flag，没有turn
皮特森（Peterson）算法
先设置，后检查
对进程i：
flag[i] = true;turn = j;
while (flag[j] && turn = j);
临界区（do my things）
flag[i] = false;
退出
turn + flag，设置自己的flag和对方的turn
硬件实现方法
中断屏蔽
cpu只在发生中断时发生进程切换
关中断
临界区
开中断
关了中断不会发生进程切换，那么就不会有其他进程访问临界区
硬件指令
TestAndSet指令
原子操作，将lock设置为1
swap指令
每一个临界区都设置一个lock
优点
简单，容易验证正确性
适用于任意多的进程
缺点
会产生饥饿
忙等待
信号量（PV操作（低级进程通信原语））
☆ 经典同步问题
生产者-消费者
读者-写者
哲学家进餐
吸烟者问题
管程
定义
保证了进程互斥，无需程序员自己实现互斥，且提供条件变量，可以实现同步
利用数据结构抽象的表示系统中的资源，进程对共享资源的申请、释放等操作，都通过这组过程来实现
代表共享资源的数据结构，一级对共享数据结构实施的操作所组成的资源管理程序，叫做管程
组成（类似于Class）
管程名称
内部共享数据结构说明
针对数据结构的操作
对数据结构的初始化语句
管程把共享资源封装起来
每次仅允许一个进程进入管程，从而实现进程互斥
条件变量
x.wait
x.signal
没有值，仅实现了排队功能，故条件变量不能反映共享资源的还剩多少，由其他数据结构记录，但是信号量可以
整型信号量
未实现让权等待
记录型信号量
不存在"忙等待现象"
自我阻塞，使用 block原语，插入阻塞队列
利用信号量实现同步
当S=0时，执行P操作，S=-1，自我阻塞
利用信号量实现互斥
初始化S=1，pv操作中间为临界资源的访问
同步、互斥与PV操作 ☆
同步
某个行为用到资源，则先P一下
某个行为提供资源，则操作过后V一下
互斥
PV操作要紧夹互斥资源的那个行为
利用信号量实现前驱关系
同步机制，初始化都为0，提供就V，使用就P
互斥锁
available=false，则临界资源在使用
缺点：忙等待