各进程互相等待对方手里的资源,导致<mark>无限</mark>阻塞即为对互斥且不可剥夺的资源进行了不合理的分配导致的无限期阻塞

互斥使用

不剥夺条件(不放手)保持再请求(拿着再要)

产生的必要条件(缺一不可)

循环等待

预防死锁:破坏四个必要条件 (最强的办法)

避免死锁:银行家算法(安全序列的计算)(这是对死锁的预防阶段,并不能判断当前是否是死锁的)

死锁的检测

资源剥夺

死锁的检测和解除

撤销进程

死锁的解除

讲程回退

空闲让进

忙则等待

互斥的四个原则

有限等待 (不饿死)

让权等待(防止忙等)(非必要)

同步与互斥

临界资源:同时只能被一个进程所 访问的资源(包括数据和变量) 实现互斥的方法 (软件)

实现互斥的方法 (硬件)

单标志检查法:单变量改变来上锁检查,能保证单一访问,但是会违反空闲让进

双标志先/后检查:进程拥有各自的变

量, 先后表示先/后检查变量

先:会出现查完没来得及锁而导致都进入

后: 会出现锁完没来得及进导致都锁住

Peterson算法:结合了单变量与 双标志后检查法的特点

单变量的改变检查能保证单一访问(互 斥),双标志的后检查能保证不会空闲等 待,故结合这两种方法能实现前三种原则。 但仍然会出现忙等现象不满足让权等待

中断屏蔽方法:简单,只能用于 单处理机,用户不能拥有关中断 的权限

TSL (Test and set lock) 是一条硬件指令,<mark>资源有一个bool变量</mark>: ture表示上锁,flase表示空闲,该指令(原子操作)若为false则进入并且改为true(上锁),若为true则循环等待

Swap硬件指令方法,是一条<mark>原子</mark> 操作,用于交换两个字

简单,均不能实现让权等待

整型信号量/记录型信号量(会主动阻塞,解决忙等)

信号量实现互斥:初始为0,分别 用p v所夹

互斥锁(信号量) ↓ 用p v向

信号量实现同步:初始为1,先v 后p

生产消费者

经典的同步问题

读者写者 哲学家进餐

吸烟者

管程:通过共享数据机构并对其操作来抽象对临界资源的操作, 管程 并提供了具体操作函数(封装操作),从而实现进程同步

死锁