一次只允许一个进程使用的资源(打印机,特殊变量,数据) 进入区:检查进程是否可以进入临界区 临界资源 临界区:可以访问临界资源的代码 临界资源的访问过程 退出区:将正在访问临界区的标志清除 剩余区:代码中的其余部分 空闲让进:临界区空闲时,可以允许一个请求进入临界区的进程立即进入临界区 基本概念 忙则等待:已有进程进入临界区后,其他试图进入临界区的进程必须等待 同步:直接制约关系,为了完成某种过任务而建立的多个进程,相互合作,所以要相互进行通信同步 遵循的原则 有限等待:对于请求访问临界区的进程,在有限时间内进入临界 让权等待:进程不能进入临界区的时候,应当立即释放处理机 互斥:间接制约关系,当一个进程访问临界资源的时候,其他进程不能访问 两个程序进程交替进入临界区 单标志法 优点:实现简单 缺点:可能会违背空闲让进,造成资源无法充分利用 每个进程访问临界区资源前,先检查临界资源是否被访问,如果空闲才能进入 双标志法先检查 优点:不用交替进入可以连续使用 缺点:两个进程可能同时进入临界区,违背忙则等待 软件实现方法 先设置自己标志,表明自己想要进入,检查对方标志,如果对方也要进入,那么就等待否则就进入 双标志法后检查 优点:不会导致两个进程都进入临界区 缺点:双方可能会互相谦让,导致饥饿现象 2.3进程同步 防止两个进程无限期等待,在算法三的基础上增加一个标志位,从而防止饥饿 皮特森算法 优点:解决了饥饿现象 实现临界区互斥的基本方法 适用于任意数目的进程 缺点:算法复杂 优点 简单且容易验证正确性 对中断进行屏蔽、关中断 支持进程内有多个临界区 中断屏蔽法 优点:关中断非常方便 优缺点 不能实现让权等待 硬件实现方法 缺点:限制了处理机交替执行程序的能力 缺点 可能会导致饥饿现象 硬件指令法 读出指定标志后,将该标志置为真 wait: 资源-1 signal=资源+1 整形信号量 没有遵循让权等待机制,会导致进程处于"忙等"状态 记录型信号量 记录型信号量不存在"忙等"现象,除了需要一个用于代表资源树木的整型变量value外,在增加一个进程链表L,用于链接所有等待该资源的进程 利用信号量实现同步 设S为进程P1和P2同步的公共信号量,初值为0,通过设置S的值可以使得P1与P2按照一定顺序执行 信号量 利用信号量实现互斥通过设置S的值,可以实现进程对临界资源的互斥访问 利用信号量实现前驱关系 通过设置不同的进程运行结束后,产生不同的信号量,从而可以使得目标进程运行,从而实现前驱关系 一组数据以及定义在这组数据之上的对这组数据的操作组成的软件模块,这组操作能初始化并改变管程中的数据和同步 定义 进程 局部于管程的共享结构数据说明 组成 对该数据结构进行操作的一组过程 管程 对局部于管程的共享数据设置初始值的语句 局部于管程的数据只能被局部于管程内的过程所访问 基本特性 一个进程只有通过调用管程内的过程才能进入管程访问共享数据 每次仅允许一个进程在管程内执行某个内部过程