

## Ch 05 进程同步互斥机制

### 进程并发执行

顺序环境：在计算机系统中只有一个程序在运行，这个程序独占系统中的所有资源，其执行不受外界影响

并发环境：

- 程序执行结果的不可再现性
- 在并发环境下程序的执行是间断性的
- 资源共享
- 独立性与制约性
- 程序和计算不再一一对应

### 竞争条件

两个或多个进程读写某些共享数据，而最后的结果取决于进程运行的精确时序

进程互斥：各进程使用共享资源，而这些资源具有排他性，各进程之间竞争使用这些资源

临界资源：系统中某些资源一次只允许一个进程使用，称这样的资源为临界资源或互斥资源。

临界区：各个进程中对某个临界资源实施操作的程序片段

### 进程的同步

系统中多个进程中发生的事件存在某种时序关系，需要相互合作，共同完成一项任务

## 实现进程互斥的方案

软件解法：

- Dekker 算法
- Peterson 算法

硬件解法：

- 中断屏蔽
- “测试并加锁”

## 信号量及 $P$ 、 $V$ 操作

$P$ 、 $V$  操作为原语

- 生产者 - 消费者问题
- 读者 - 写者问题

## 管程

为什么引入管程：

1. 信号量机制不足：编写困难、效率低
2. Hansen、Hoare 管程
3. 一种高级同步机制

## 管程的定义

1. 是一个特殊模块
2. 有一个名字
3. 由关于共享资源的数据结构及其在其上的一组过程组成

进程只能通过调用管程中的过程间接访问管程中的数据结构

管程要保证什么

互斥：管程是互斥进入的，由编译器保证

同步：设置条件变量、等待/唤醒操作

Hoare 管程

管程的入口处设置一个进程等待队列，入口等待队列

管程内设置进程等待队列，称为紧急等待队列

条件变量的实现

条件变量——管程内部说明使用的变量，执行 wait 和 signal 操作

管程的应用

管程的实现：

1. 直接构造：效率高
2. 间接构造：用已存在的机制构建

MESA 管程

Hoare 有两次额外的进程切换

Hoare 与 MESA 管程的比较

MESA 管程出错比较少

## 进程间通信机制

为什么需要通信机制：

1. 信号量及管程的不足
2. 多处理器情况下原语失效