



第二章 Windows CE体系结 构与功能

教学目的

- 熟悉进程的同步与互斥的概念；理解 Windows CE5.0 中进程间通信的手段，并能够理解并编写简单的多线程应用程序。

项目4：多线程-PDA下多任务实现

本节课程主要知识点

- Windows CE5.0进程的同步与互斥（重点）
- Windows CE5.0进程间通信（难点）

■ 教学重点

- 同步与互斥的概念

■ 教学难点

- 进程的通信

进程的互斥

- 系统中的很多资源是共享资源。有些资源每时刻只允许由一个进程使用，同时请求该资源的进程就需要竞争使用这些资源，进程的这种关系为进程的互斥。

生活中的实例

- 共享资源每时每刻只允许一个人使用，这样的例子在生活中多的很吧？
- 图书馆的座位，公用电话...
- 同学们自己找找看？

临界资源critical resource

■ 临界资源

一次仅允许一个进程使用的资源。

- 很多物理设备都属于临界资源。例如：纸带或卡片输入机、打印机、磁带机等。
- 除了物理设备外，许多变量、数据、表格、队列等也可以由若干个进程共享使用，但每时刻仅允许一个进程使用，这些资源也属于临界资源。

临界区critical section

■ 临界区

针对同一临界资源进行操作的程序段，也称为互斥段。

- 任一共享进程一旦对临界资源开始操作，则在该操作结束之前，其它进程就不能对该临界资源进行处理。也就是说，执行这些操作的程序段必须是互斥执行的。

2.2.4 同步

- 产生的原因：多线程、分时共享技术的使用；
- 作用：对多线程共享资源的访问加以保护；
- 必要性：造成数据损坏或者死锁。

□ Windows CE 的同步机制

线程A读写数据时，另一个线程B也对这个数据进行读写，导致数据冲突，引起数据混乱

内核

线程A需要线程B中的数据才能够运行，而线程B也需要线程A中的数据才能运行

- Critical Section
- Interlocked Functions

用户态

□ 同步对象的状态

- 通知(Signaled): 阻塞线程转为就绪态，等待调度执行
- 未通知(Non-Signaled): 表示该同步对象被一个或多个线程占有，不能被其它等待的线程占有

□ 同步的基本原理：当某个同步对象的状态变为通知状态时，等待在它上面的阻塞线程会得到通知，并且转为就绪态，等待调度执行。

□ 同步对象的使用

- 由句柄(Handle)表示，一个32位的指针
- 句柄的指向表示该对象的一个数据结构
- 可以有名字也可以没有名字
- 若跨进程使用同步对象，一般要命名。

□ 等待函数（见P42）

■ WaitForSingleObject

□ 等待单个同步对象

■ WaitForMultipleObjects

□ 等待多个同步对象

} 两个函数都会阻塞当前的调用线程，直到等待的同步对象变为通知状态

□ 2.2.4.1 Mutex

- Mutual Exclusion, 互斥
- 同时只有一个线程可以占有Mutex对象
- 当共享资源同时只能被一个线程访问,使用Mutex进行同步
- 相关函数
 - CreateMutex
 - 创建一个有名或无名的Mutex对象
 - bInitialOwner参数指定该Mutex是否为创建的线程所占有
 - ReleaseMutex
 - 释放对Mutex对象的占有

□ 2.2.4.2 Semaphore

- 信标或信号灯
- 带有引用计数的**Mutex**，即带有计数**n**，表示同时最多可以有**n**个线程占有该信号灯
- 若共享资源同时被固定数量的线程使用，则选用**Semaphore**进行同步
- 相关函数
 - **CreateSemaphore**
 - 创建信号灯
 - 参数**IMaximumCount**指定最多可有多少个线程占有该信号灯对象
 - 参数**IInitialCount**为信号灯指定初始计数

□ ReleaseSemaphore

- 使用此函数释放信号灯对象，则InitialCount指定的初始值增加ReleaseCount
- 当有线程使用ReleaseSemaphore函数释放Semaphore对象时，该初始值增加1；当有线程使用等待函数占有信号灯时，该初始值减1
- 若信号灯的计数大于0，则信号灯处于通知状态；若计数小于0，处于未通知状态

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/417054036110006106>