

线程间的通信、同步方式与进程间通信方式

1、线程间的通信方式

- 使用全局变量
主要由于多个线程可能更改全局变量，因此全局变量最好声明为[volatile](#)
- 使用消息实现通信
在Windows程序设计中，每一个线程都可以拥有自己的消息队列（UI线程默认自带消息队列和消息循环，工作线程需要手动实现消息循环），因此可以采用消息进行线程间通信SendMessage,postMessage。

```
1)定义消息#define WM_THREAD_SENDMSG=WM_USER+20;  
2)添加消息函数声明afx_msg int OnTSendmsg();  
3)添加消息映射ON_MESSAGE(WM_THREAD_SENDMSG,OnTSM)  
4)添加OnTSM()的实现函数;  
5)在线程函数中添加PostMessage消息Post函数
```

- 使用事件CEvent类实现线程间通信
Event对象有两种状态：有信号和无信号，线程可以监视处于有信号状态的事件，以便在适当的时候执行对事件的操作。

```
1)创建一个CEvent类的对象：CEvent threadStart;它默认处在未通信状态;  
2)threadStart.SetEvent();使其处于通信状态;  
3)调用WaitForSingleObject()来监视CEvent对象
```

2、线程间的同步方式

各个线程可以访问进程中的公共变量，资源，所以使用多线程的过程中需要注意的问题是如何防止两个或两个以上的线程同时访问同一个数据，以免破坏数据的完整性。数据之间的相互制约包括 1、直接制约关系，即一个线程的处理结果，为另一个线程的输入，因此线程之间直接制约着，这种关系可以称之为同步关系 2、间接制约关系，即两个线程需要访问同一资源，该资源在同一时刻只能被一个线程访问，这种关系称之为线程间对资源的互斥访问，某种意义上说互斥是一种制约关系更小的同步

线程间的同步方式有四种

- 临界区
临界区对应着一个CriticalSection对象，当线程需要访问保护数据时，调用EnterCriticalSection函数；当对保护数据的操作完成之后，调用LeaveCriticalSection函数释放对临界区对象的拥有权，以使另一个线程可以夺取临界区对象并访问受保护的数据。
PS:[关键段对象会记录拥有该对象的线程句柄即其具有“线程所有权”概念，即进入代码段的线程在leave之前，可以重复进入关键代码区域。所以关键段可以用于线程间的互斥，但不可以用于同步（同步需要在一个线程进入，在另一个线程leave）](#)
- 互斥量
互斥与临界区很相似，但是使用时相对复杂一些（互斥量为内核对象），不仅可以在同一应用程序的线程间实现同步，还可以在不同的进程间实现同步，从而实现资源的安全共享。
PS:[1、互斥量由于也有线程所有权的概念，故也只能进行线程间的资源互斥访问，不能由于线程同步；2、由](#)

[互斥量是内核对象，因此其可以进行进程间通信，同时还具有一个很好的特性，就是在进程间通信时完美的解决了"遗弃"问题](#)

- 信号量

信号量的用法和互斥的用法很相似，不同的是它可以同一时刻允许多个线程访问同一个资源，PV操作

PS:[事件可以完美解决线程间的同步问题，同时信号量也属于内核对象，可用于进程间的通信](#)

- 事件

事件分为手动置位事件和自动置位事件。事件Event内部它包含一个使用计数（所有内核对象都有），一个布尔值表示是手动置位事件还是自动置位事件，另一个布尔值用来表示事件有无触发。由SetEvent()来触发，由ResetEvent()来设成未触发。

PS:[事件是内核对象,可以解决线程间同步问题，因此也能解决互斥问题](#)

3、进程间通信方式

进程间通信又称IPC(Inter-Process Communication),指多个进程之间相互通信，交换信息的方法。根据进程通信时信息量大小的不同,可以将进程通信划分为两大类型: 1、低级通信,控制信息的通信(主要用于进程之间的同步,互斥,终止和挂起等等控制信息的传递) 2、高级通信,大批数据信息的通信(主要用于进程间数据块数据的交换和共享,常见的高级通信有管道,消息队列,共享内存等)。

- [管道\(pipe\)](#)：管道是一种[半双工的通信方式](#)，数据只能单向流动，而且[只能在具有亲缘关系的进程间使用](#)。进程的亲缘关系通常是指父子进程关系。
- [有名管道\(named pipe\)](#)：有名管道也是半双工的通信方式，但是它[允许无亲缘关系进程间的通信](#)。
- [信号量\(semaphore\)](#)：信号量是一个计数器，可以用来控制多个进程对共享资源的访问。[不是用于交换大批数据,而用于多线程之间的同步,常作为一种锁机制,防止某进程在访问资源时其它进程也访问该资源](#)。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。
- [消息队列\(message queue\)](#)：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列[克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点](#)。
- [信号\(signal\)](#)：信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。
- [共享内存(shared memory)]：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是[最快的IPC方式](#)，它是针对其他进程间通信方式运行效率低而专门设计的。[它往往与其他通信机制，如信号两，配合使用，来实现进程间的同步和通信](#)。
- [套接字\(socket\)](#)：套接口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同机器间的进程通信。

4、好文推荐 (reference)

- [秒杀多线程面试题系列](#)
- [VC中利用多线程技术实现线程之间的通信](#)
- [几种进程间的通信方式](#)
- [Linux下的进程间通信-详解](#)
- [Linux环境进程间通信](#)
- [进程的通信方式及特点](#)