MFC提供了多种同步对象，下面我们只介绍最常用的四种：

* 临界区（CCriticalSection）
* 事件（CEvent）
* 互斥量（CMutex）
* 信号量（CSemaphore）

***A、使用 CCriticalSection 类***

　　当多个线程访问一个独占性共享资源时,可以使用“临界区”对象。任一时刻只有一个线程可以拥有临界区对象，拥有临界区的线程可以访问被保护起来的资源或代码段，其他希望进入临界区的线程将被挂起等待，直到拥有临界区的线程放弃临界区时为止，这样就保证了不会在同一时刻出现多个线程访问共享资源

***B、使用 CEvent 类***

　　CEvent 类提供了对事件的支持。事件是一个允许一个线程在某种情况发生时，唤醒另外一个线程的同步对象。例如在某些网络应用程序中，一个线程（记为A）负责监听通讯端口，另外一个线程（记为B）负责更新用户数据。通过使用CEvent 类，线程A可以通知线程B何时更新用户数据。每一个CEvent 对象可以有两种状态：有信号状态和无信号状态。线程监视位于其中的CEvent 类对象的状态，并在相应的时候采取相应的操作。  
　　在MFC中，CEvent 类对象有两种类型：人工事件和自动事件。一个自动CEvent 对象在被至少一个线程释放后会自动返回到无信号状态；而人工事件对象获得信号后，释放可利用线程，但直到调用成员函数ReSetEvent()才将其设置为无信号状态。在创建CEvent 类的对象时，默认创建的是自动事件。 CEvent 类的各成员函数的原型和参数说明如下：

1、CEvent(BOOL bInitiallyOwn=FALSE,

BOOL bManualReset=FALSE,

LPCTSTR lpszName=NULL,

LPSECURITY\_ATTRIBUTES lpsaAttribute=NULL);

* bInitiallyOwn:指定事件对象初始化状态，TRUE为有信号，FALSE为无信号；
* bManualReset：指定要创建的事件是属于人工事件还是自动事件。TRUE为人工事件，FALSE为自动事件；
* 后两个参数一般设为NULL，在此不作过多说明。

2、BOOL CEvent：：SetEvent();

　　将 CEvent 类对象的状态设置为有信号状态。如果事件是人工事件，则 CEvent 类对象保持为有信号状态，直到调用成员函数ResetEvent()将 其重新设为无信号状态时为止。如果CEvent 类对象为自动事件，则在SetEvent()将事件设置为有信号状态后，CEvent 类对象由系统自动重置为无信号状态。  
  
如果该函数执行成功，则返回非零值，否则返回零。

3、BOOL CEvent：：ResetEvent();

　　该函数将事件的状态设置为无信号状态，并保持该状态直至SetEvent()被调用时为止。由于自动事件是由系统自动重置，故自动事件不需要调用该函数。如果该函数执行成功，返回非零值，否则返回零。我们一般通过调用WaitForSingleObject函数来监视事件状态。

***C、使用CMutex 类***

　　互斥对象与临界区对象很像.互斥对象与临界区对象的不同在于:互斥对象可以在进程间使用,而临界区对象只能在同一进程的各线程间使用。当然，互斥对象也可以用于同一进程的各个线程间，但是在这种情况下，使用临界区会更节省系统资源，更有效率。

***D、使用CSemaphore 类***

　　当需要一个计数器来限制可以使用某个线程的数目时，可以使用“信号量”对象。CSemaphore 类的对象保存了对当前访问某一指定资源的线程的计数值，该计数值是当前还可以使用该资源的线程的数目。如果这个计数达到了零，则所有对这个CSemaphore 类对象所控制的资源的访问尝试都被放入到一个队列中等待，直到超时或计数值不为零时为止。一个线程被释放已访问了被保护的资源时，计数值减1；一个线程完成了对被控共享资源的访问时，计数值增1。这个被CSemaphore 类对象所控制的资源可以同时接受访问的最大线程数在该对象的构建函数中指定。  
  
CSemaphore 类的构造函数原型及参数说明如下：

CSemaphore (LONG lInitialCount=1,

LONG lMaxCount=1,

LPCTSTR pstrName=NULL,

LPSECURITY\_ATTRIBUTES lpsaAttributes=NULL);

* lInitialCount:信号量对象的初始计数值，即可访问线程数目的初始值；
* lMaxCount：信号量对象计数值的最大值，该参数决定了同一时刻可访问由信号量保护的资源的线程最大数目；
* 后两个参数在同一进程中使用一般为NULL，不作过多讨论；

　　在用CSemaphore 类的构造函数创建信号量对象时要同时指出允许的最大资源计数和当前可用资源计数。一般是将当前可用资源计数设置为最大资源计数，每增加一个线程对共享资源的访问，当前可用资源计数就会减1，只要当前可用资源计数是大于0的，就可以发出信号量信号。但是当前可用计数减小到0时，则说明当前占用资源的线程数已经达到了所允许的最大数目，不能再允许其它线程的进入，此时的信号量信号将无法发出。线程在处理完共享资源后，应在离开的同时通过ReleaseSemaphore()函数将当前可用资源数加1。