

System.Threading	
InterLocked	为被多个线程共享访问的类型提供原子操作
Monitor	使用锁定和等待信号来同步线程对象.C#的lock关键字在后台使用的就是Monitor对象
Mutex	互斥体,可用于应用程序域边界之间的同步
ParameterizedThreadStart	委托,它允许线程调用包含任意多个参数的方法
Semaphore	用于限制对一个资源或一类资源的并发访问的线程数量
Thread	代表CLR中执行的线程,使用这个类型,能够在处室的应用程序域中常见额外的线程
ThreadPool	用于的一个进程中的线程池交互
ThreadPriority	代表了线程调度的优先级别(Highest,Normal等)
ThreadStart	该委托用于id你故意一个线程所调用的方法.和ParameterizedThreadStart委托不同,这个方法的目标必须符合一种固定的原型
TreadState	代表线程处于的状态(Running,Aborted等)
Timer	提供以指定的时间间隔执行方法的机制
TimerCallback	该委托类型应与Timer类型一起使用

System.Threading.Thread 类	
CurrentContext	只读属性,返回当前线程的上下文
CurrentThread	只读属性,返回当前线程的引用
GetDomain()和GetDomainId()	返回当前应用程序域的引用或当前线程正在运行的域的ID
Sleep()	将当前线程挂起指定的时间
IsAlive	返回布尔值,指示线程是否开始了
IsBackground	获取或者设置一个值,指示线程是否为后台线程
Name	给线程指定友好的名字
Priority	获取或设置线程的调度优先级.它是ThreadPriority枚举中的值之一
ThreadState	获取当前线程的状态.它是ThreadState枚举中的

ThreadState	值之一
Abort()	通知CLR尽快终止本线程
Interrupt()	中断当前线程,唤醒处于等待中的线程
Join()	阻塞调用线程,直到某个(调用Join()的)线程终止为止
Resume()	使已挂起的线程继续执行
Start()	通知CLR尽快执行本线程
Suspend()	挂起当前线程,如果线程已挂起,调用Suspend()则不起作用

System.Threading.Interlocked	
CompareExchange()	安全的比较两个值是否相等.如果相等,将第3个值与其中1个值交换
Decrement()	安全递减1
Exchange()	安全的交换数据
Increment()	安全递增1