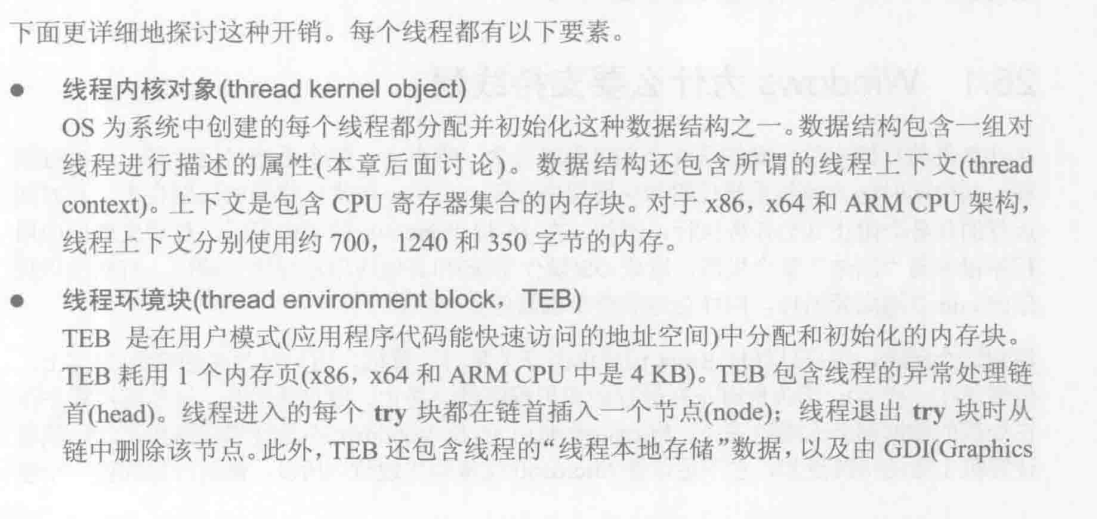
为什么需要线程？

如果没有线程，当只有一个CPU时，应用程序出现死循环就会导致死机。因此，线程的职责是对CPU虚拟化，windows为每个进程都提供专属的线程，如果该程序的代码死循环不会影响到其他进程。

每个线程都含有的开销：



报纸上的文字

描述已自动生成

虽然线程认为自己独享CPU的时间，但实际上每个线程只能运行一个时间片的时间，时间片到期Windows切换到下一个线程，切换上下文时Windows进行的操作：

文本

描述已自动生成

Ps：上下文切换是“净开销”不会产生任何程序执行上的收益，但是它保证了操作系统的健壮性和响应的灵敏性

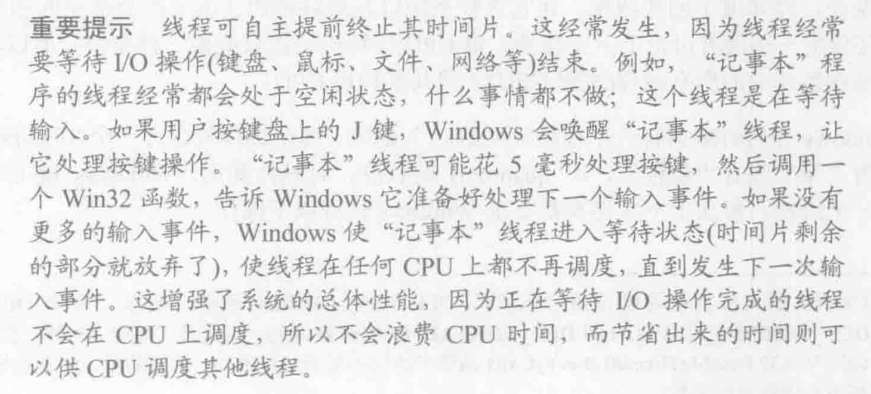
上下文切换导致更多的开销（除开上下文切换花费的时间以外的开销）：

一些文字和图片的手机截图

描述已自动生成

Ps：如果两个连续时间片都调度的同一个线程，那么不会执行上下文切换

时间片是可以提前终止的



线程需要消耗大量内存，且需要消耗很多时间来创建、管理、销毁；但还是得用，因为进程的时间和性能开销都比线程大很多。

CLR线程在使用上完全等价于Windows线程

使用专属线程执行异步的计算操作：

满足以下条件就可以显式创建线程：

文本

描述已自动生成

文本

描述已自动生成

创建线程的方法：

图片包含 文本

描述已自动生成

例子：

文本, 信件

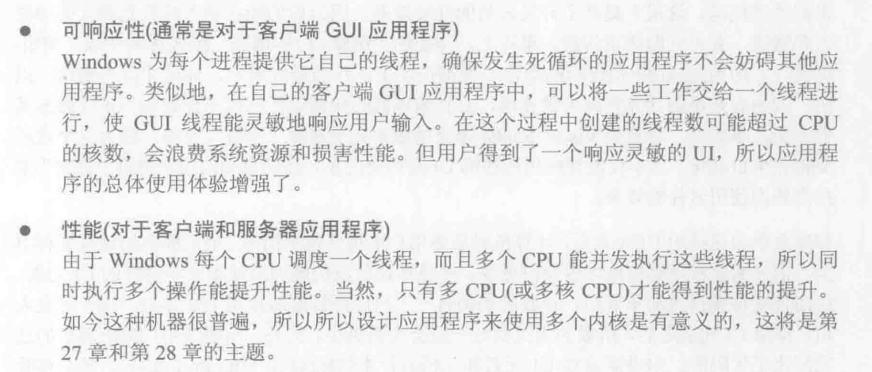
描述已自动生成

文本

描述已自动生成

Ps：线程执行先后是不可控的

使用线程的好处：



线程调度和优先级：

在Windows中，线程调度是基于优先级的抢占式调度算法。每个线程都有一个基本优先级（base priority）和一个动态优先级（priority）。优先级的值处于0~31之间，共分为三个类别

1. 0级表示系统优先级，为最低优先级，仅用于零页面线程。

2. 1-15级表示动态优先级，这是计划程序用于确定要执行的线程的优先级。最初线程的动态优先级与其基本优先级相同。系统可以提升和降低动态优先级，以确保其响应迅速，并且没有线程因处理器时间而耗尽。

3. 16-31级表示实时类别，这是优先级最高的。

Ps：当系统确定给哪个线程分配CPU时，它会首先查看优先级为31的线程，并以循环的方式进行调度。如果有优先级为31的线程可供调度，那么系统就会将CPU分配给该线程。在提高线程的动态优先级后，计划程序在每次线程完成一个时间片时，会将该优先级降低一级，直到线程回退到其基本优先级。线程的动态优先级始终不低于其基本优先级。

还有一些特定情况下系统会提升线程的动态优先级，例如将使用NORMAL\_PRIORITY\_CLASS的进程带到前台时，计划程序将提升与前台窗口关联的进程的优先级类，使其大于或等于任何后台进程的优先级类。当窗口接收计时器消息、鼠标消息或键盘输入等输入时，计划程序会提升拥有窗口的线程的优先级。满足阻塞线程的等待条件后，计划程序会提高线程的优先级。

相对线程优先级：

Time-critical：关键时间，这是最高的相对线程优先级。

Highest：最高，但并不是最高的相对线程优先级。

AboveNormal：高于正常。

Normal：正常，当一个线程被创建时，它的线程相对优先级默认为Normal。

BelowNormal：低于正常。

Lowest：最低。

Idle：空闲，这是最低的相对线程优先级。

进程优先级类：

IDLE\_PRIORITY\_CLASS：空闲，其线程仅在系统空闲时运行的进程。更高优先级类中运行的任何进程的线程都优先于此进程的线程。

BELOW\_NORMAL\_PRIORITY\_CLASS：低于正常，优先级高于IDLE\_PRIORITY\_CLASS但低于NORMAL\_PRIORITY\_CLASS的进程。

NORMAL\_PRIORITY\_CLASS：正常，没有特殊计划需求的流程。默认情况下，进程的优先级类为NORMAL\_PRIORITY\_CLASS。

ABOVE\_NORMAL\_PRIORITY\_CLASS：高于正常，优先级高于NORMAL\_PRIORITY\_CLASS但低于HIGH\_PRIORITY\_CLASS的进程。

HIGH\_PRIORITY\_CLASS：高，执行必须立即执行的时间关键型任务的进程。该进程的线程优先于普通或空闲优先级类进程的线程。

REALTIME\_PRIORITY\_CLASS：实时，具有最高优先级的进程。进程的线程会抢占所有其他进程的线程，包括执行重要任务的操作系统进程。

进程优先级类和相对线程优先级怎么映射到优先级值：



文本

描述已自动生成

Ps：高优先级线程大多数时间都应该处于等待状态

文本, 信件

描述已自动生成

使用代码只能设置5个相对线程优先级

报纸上的文字

描述已自动生成

前台线程：当进程中的所有后台线程停止运行时，CLR不会强制终止仍在运行的前台线程。因此，你应该用前台线程来执行你想完成的任务，比如把内存缓冲区的数据转移到磁盘。

后台线程：当进程中的所有前台线程停止运行时，CLR会强制终止仍在运行的后台线程。这些后台线程被直接终止：不会抛出异常。因此，应该为非关键的任务使用后台线程，比如重算电子表格的单元格，或者为记录建立索引。这些工作能在应用程序重启的时候继续，而且如果用户想终止应用程序，就没有必要强迫它保持活动状态。

例子：

图形用户界面, 文本, 应用程序

描述已自动生成

Ps：在线程的生存周期中，任何时候都可以从前台线程变成后台线程或者从后台线程变成前台线程。通过代码构造一个Thread对象显示的创建线程都是前台线程，通过线程池ThreadPool/Task创建的线程都是后台线程。

Ps：优先级反转，指高优先级进程被低优先级进程阻塞，因为低优先级进程锁住了两者的共享资源导致前者等待时间长。