为什么需要线程池：创建线程是开销很大的操作，因此为每个短暂的异步操作都创建线程会产生显著的开销。因此可以事先分配一定资源，将资源放入资源池（即在刚开始运行程序时一次性创建大量线程，在程序运行之初消耗大量资源），每次需要创建线程时从线程池中取出一个而避免创建从而减小开销，而当线程执行完毕时返回线程池中。

Ps：在需要创建的线程的数目多于线程池中线程数目时，会创建新的线程，这些线程在执行完毕后同样返回线程池挂起；相应的，长时间不被使用，线程池中休眠的线程也会短暂苏醒然后结束自己，从而减少线程池中线程的总数。

C#中如何使用线程池：通过System.Threading.ThreadPool类型可以使用.Net Framework提供的线程池，该线程池由CLR管理。

ThreadPool类提供的方法和属性：

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 文本

描述已自动生成

在线程池中异步执行委托：

Ps：异步编程模型(APM)(使用IAsyncResult和BeginInvoke)不再是异步调用的优选方法。从.NET Framework 4.5开始，基于任务的异步模式(TAP)(使用Task)是推荐的异步模型。

[System.PlatformNotSupportedException:“Operation is not supported on this platform.” - 阿水zev - 博客园 (cnblogs.com)](https://www.cnblogs.com/Zev_Fung/p/12823418.html)

书P42代码执行抛出异常：

电脑萤幕的截图

描述已自动生成

这是书上对这段代码的解释：

文本, 信件

描述已自动生成

文本, 信件

描述已自动生成

向线程池中放入异步操作：

QueueUserWorkItem：将方法排入队列以便执行。 此方法在有线程池线程变得可用时执行。

图形用户界面, 文本, 信件

描述已自动生成

例子：

文本

描述已自动生成

在线程池中取消异步操作：

在线程池中取消异步操作的三种方式：第一种，轮询检查CancellationToken结构体的IsCancellationRequest属性，如果为true说明操作需要被取消，放弃该操作。第二种，抛出OperationCancelledException异常，通过操作之外的代码取消操作。第三种，回调函数，当操作被取消时，线程池调用该回调函数，这种方式允许链式传递一个取消逻辑到另一个逻辑中。

Ps：CancellationToken结构体由CancellationTokenSource类创建，并且该类提供通知CancellationToken的方法

CancellationTokenSource类：

文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

CancellationToken结构体

图形用户界面, 文本, 应用程序

描述已自动生成

文档：[CancellationToken Struct (System.Threading) | Microsoft Learn](https://learn.microsoft.com/zh-cn/dotnet/api/system.threading.cancellationtoken?view=net-8.0)

[CancellationTokenSource 类 (System.Threading) | Microsoft Learn](https://learn.microsoft.com/zh-cn/dotnet/api/system.threading.cancellationtokensource?view=net-8.0)

线程池中使用等待事件处理器以及超时：

使用ThreadPool.RegisterWaitForSingleObject方法，该方法允许将回调函数放入线程池队列中，当提供的等待事件处理器收到信号或发生超时时，该回调方法将被调用。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

文档：[ThreadPool.RegisterWaitForSingleObject 方法 (System.Threading) | Microsoft Learn](https://learn.microsoft.com/zh-cn/dotnet/api/system.threading.threadpool.registerwaitforsingleobject?view=net-8.0#system-threading-threadpool-registerwaitforsingleobject(system-threading-waithandle-system-threading-waitortimercallback-system-object-system-int32-system-boolean))

例子：

图形用户界面, 文本

描述已自动生成

System.Threading.Timer类：

使用System.Threading.Timer类对象在设定的间隔之后生成事件，带有生成重复事件的选项。以此可以创建周期性调用的异步操作。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 表格

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

文档: [Timer Class (System.Threading) | Microsoft Learn](https://learn.microsoft.com/zh-cn/dotnet/api/system.threading.timer?view=net-8.0)

这是书上P52的例子:

文本

描述已自动生成

文本, 信件

描述已自动生成

BackGroundWorker组件:

借助于BackGroundWorker对象可以将异步代码组织为一系列事件及事件处理器，多使用于Windows窗体程序