

在C Builer中多线程的实现 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/133/2021_2022__E5_9C_A8C__Buil_c97_133836.htm

还在Dos时代，人们就在寻求一种多任务的实现。于是出现了TSR类型的后台驻留程序，比较有代表性的有Side Kick、Vsafe等优秀的TSR程序，这类程序的出现和应用确实给用户使用计算机带来了极大的方便，比如Side Kick，我们编程可以在不用进编辑程序的状态下，一边编辑源程序，一边编译运行，非常方便。但是，Dos单任务操作系统的致命缺陷注定了在Dos下不可能开发出真正的多任务程序。进入Windows3.1时代，这种情况依然没有根本的改变，一次应用只能做一件事。比如数据库查询，除非应用编得很好，在查询期间整个系统将不响应用户的输入。进入了Windows NT和Windows 9x时代，情况就有了彻底的改观，操作系统从真正意义上实现了多任务（严格地说，Win9x还算不上）。一个应用程序，在需要的时候可以有许多个执行线程，每个线程就是一个小的执行程序，操作系统自动使各个线程共享CPU资源，确保任一线程都不能使系统死锁。这样，在编程的时候，可以把费时间的任务移到后台，在前台用另一个线程接受用户的输入。对那些对实时性要求比较高的编程任务，如网络客户服务、串行通信等应用时，多线程的实现无疑大大地增强了程序的可用性和稳固性。

在Windows NT和Windows 9x中，多线程的编程实现需要调用一系列的API函数，如CreateThread、ResumeThread等，比较麻烦而且容易出错。我们使用Inprise公司的新一代RAD开发工具C++ Builder，可以方便地实现多线程的编程。与老

牌RAD工具Visual Basic和Delphi比，C++ Builder不仅功能非常强大，而且它的编程语言是C++，对于系统开发语言是C的Windows系列操作系统，它具有其它编程语言无可比拟的优势。利用C++ Builder提供的TThread对象，多线程的编程变得非常简便易用。那么，如何实现呢？且待我慢慢道来，让你体会一下多线程的强大功能。

1. 创建多线程程序：首先，先介绍一下实现多线程的具体步骤。在C++ Builder中虽然用TThread对象说明了线程的概念，但是TThread对象本身并不完整，需要在TThread下新建其子类，并重载Execute方法来使用线程对象。在C++ Builder下可以很方便地实现这一点。

在C++ Builder IDE环境下选择菜单File New，在New栏中选中Thread Object，按OK，接下来弹出输入框，输入TThread对象子类的名字MyThread，这样C++ Builder自动为你创建了一个名为TMyThread的TThread子类。同时编辑器中多了一个名为Unit2.cpp的单元，这就是我们创建的TMyThread子类的原码，如下：

```
#include #pragma hdrstop #include "Unit2.h"
#pragma package(smart_init) // - - - - -
- - - - - // Important: Methods and properties of
objects in VCL can only be // used in a method called using
Synchronize, for example: // // Synchronize(UpdateCaption). // //
where UpdateCaption could look like: // // void __fastcall
MyThread::UpdateCaption() // { // Form1 - >Caption =
" Updated in a thread " . // } // - - - - -
- - - - - __fastcall MyThread::MyThread(bool
CreateSuspended) : TThread(CreateSuspended) { } // - - - - -
- - - - - void __fastcall
```

```
MyThread::Execute() { // - - - - Place thread code here - - -  
- } // - - - - - - - - - - - - - - - - - - - - - - - - 其中  
的Execute ( ) 函数就是我们要在线程中实现的任务的代码所  
在处。在原代码中包含Unit2.cpp，这个由我们创建  
的TMyThread对象就可以使用了。使用时，动态创建一个  
TMyThread 对象，在构造函数中使用Resume ( ) 方法，那  
么程序中就增加了一个新的我们自己定义的线程TMyThread  
，具体执行的代码就是Execute ( ) 方法重载的代码。要加载  
更多的线程，没关系，只要继续创建需要数量的TMyThread  
对象就成。以上我们初步地实现了在程序中创建一个自定义  
的线程，并使程序实现了多线程应用。但是，多线程应用的  
实现，并不是一件简单的工作，还需要考虑很多使多个线程  
能在系统中共存、互不影响的因素。比如，程序中公共变量  
的访问、资源的分配，如果处理不当，不仅线程会死锁陷入  
混乱，甚至可能会造成系统崩溃。总的来讲，在多线程编程  
中要注意共享对象和数据的处理，不能忽视。100Test 下载频  
道开通，各类考试题目直接下载。详细请访问
```

www.100test.com